MARX® software security GERMANY

securing the digital world™

Application Notes

AutoCrypt

AN-1

Purpose of Application: Automatic Protection of applications without programming efforts
Version: Smarx OS PPK 8.18 and higher, AutoCrypt Package V1.5 and higher
Last Update: 9 Aug 2024
Target Operating Systems: Windows (CRYPTO-BOX formatting also under Linux and macOS)
Target Processor Platforms: x86/amd64
Access to source code needed (of protection application): □ Yes ☑ No
Applicable for Product: CRYPTO-BOX® SC / XS / Versa

Automatic software protection - AutoCrypt

AutoCrypt provides protection of applications without any programming efforts.

During the automatic protection process, your original application will code will be encrypted, compressed and wrapped in a protective layer that prevents it from working unless the license was successfully validated. AutoCrypt has many licensing features which can be used creatively in distribution strategies. These features include: expiration dates, usage limits, periodic hardware checks, passwords and much more.

- Quick and easy protection in just minutes with minimal efforts
- No programming knowledge or source code required
- Licensing options such as Expiration Date, number of allowed program starts or Network Licenses can be defined easily and updated later using Remote Update Management System (RUMS)
- Protects 32/64 Bit Windows applications (EXE and DLL files and .NET executables, including .NET Core)

CRYPTO-BOX®

- Quick and easy protection of Windows applications with AutoCrypt
- Individual implementations with API for all common programming languages and platforms (Windows, Linux, macOS)
- Unique and stable metal case, optional with customer-specific color and labeling
- Network and remote update capability
- EAL 4+ certified Smart Card chip with AES/Rijndael encryption implemented on-chip.
- RSA 2048 bit support on-chip (CRYPTO-BOX SC) or on driver level (CRYPTO-BOX XS).
- The CRYPTO-BOX® is designed in Germany and manufactured in the European Union (TAA compliant).
- Customer specific solutions, such as implementation of specific functions or algorithms in firmware on request.
- Also available with USB-C plug and in different colors and customized product labeling.

Order your CRYPTO-BOX Evaluation Kit now – or contact us for any questions:

MARX Software Security GmbH
Vohburger Strasse 68
85104 Wackerstein, Germany
Phone: +49 (0) 8403 / 9295-0
contact-de@marx.com

MARX CryptoTech LP
489 South Hill Street
Buford, GA 30518 U.S.A.
Phone: (+1) 770 904 0369
contact@marx.com

www.marx.com

Table of Contents

1. AutoCrypt: Overview....3
1.1. Considering the suitable AutoCrypt Package for your needs....3
1.1.1. AutoCrypt Wizard....3
1.1.2. AutoCrypt SxAF....3
1.1.3. Command Line Tools (AC_Tool, SmrxProg)....3
1.2. AutoCrypt Download and Installation....4
1.2.1. Smarx® OS Professional Protection Kit (PPK)....4
1.2.2. AutoCrypt Wizard Package....4

2. AutoCrypt Wizard....5
2.1. Starting AutoCrypt Wizard....5
2.1.1. Smarx® OS Professional Protection Kit (PPK)....5
2.1.2. AutoCrypt Wizard Package....5
2.2. Protecting Applications....6
2.2.1. Steps for Protecting Applications with AutoCrypt Wizard....6
2.2.2. Creating New Projects or Selecting Existing Projects....6
2.2.3. Project Settings....7
2.2.4. Hardware Settings....8
2.2.5. Licensing Options....9
2.2.6. Advanced Options....10
2.2.7. Dialog Messages....11
2.2.8. Application Settings....12
2.2.9. Protecting the Application....14
2.3. Format CRYPTO-BOX®....15
2.4. Exporting XML Script for Use with Command Line Tools....16
2.5. Process Remote Updates....16
2.5.1. Creating Remote Update Tool....16
2.5.2. Initiating Remote Updates (End-User)....17
2.5.3. Generate Activation Code....18
2.5.4. Executing the Activation Code (End User)....18

3. AutoCrypt SxAF (Smarx® Application Framework)....19
3.1. Starting AutoCrypt SxAF....19
3.2. Smarx Application Framework (SxAF)....19
3.3. Protecting Applications....20
3.3.1. Steps for Protecting Applications with AutoCrypt....20
3.3.2. Creating new projects or selecting existing projects....20
3.3.3. Importing Settings from AutoCrypt Wizard....21
3.3.4. General Project Settings....22
3.3.5. Adding Applications to the Project....24
3.3.6. Application protection settings....24
3.3.7. Licensing Options (Data Objects)....25
3.3.8. Advanced Protection options....26
3.3.9. Defining Dialog Boxes....27
3.3.10. .Net Options....27
3.3.11. Product Editions....29
3.3.12. Protecting the Application....29
3.3.13. Generating XML Script for Usage with Command Line Tools....30
3.4. CRYPTO-BOX® Format: Configuring and Programming....30
3.4.1. Selecting projects to format....30
3.4.2. Formatting CRYPTO-BOX units....31
3.5. Creating Remote Update Tool....32
3.6. End User Management....32

4. Command Line Tools....33

4.1. AutoCrypt - Command Line Version....33
4.2. SmrxProg - Command Line based CRYPTO-BOX® Formatting....33
5. Distributing Protected Applications to your End Users....34
6. FAQ – Frequently Asked Questions....35

1. AutoCrypt: Overview

1.1. Considering the suitable AutoCrypt Package for your needs

AutoCrypt protects applications without any programming efforts. It wraps any existing executable file with a secure layer of protection, including compression and encryption of the original code. AutoCrypt has many features which enhance creative distribution strategies, including expiration dates, usage limits, periodic hardware checks, passwords and much more.

MARX offers 3 AutoCrypt variants:

1.1.1. AutoCrypt Wizard

This is the easiest way to protect your .EXE or DLL files. The Wizard guides you through each step of protection, licensing and CRYPTO-BOX configuration for quick results. You may also export the project file for usage with with our command line tools to automate the protection process.

If you are looking for a quick protection solution providing common licensing features, then AutoCrypt Wizard is the right choice.

See chapter *2* for step-by-step instructions on using AutoCrypt Wizard.

1.1.2. AutoCrypt SxAF

This solution is less intuitive compared to AutoCrypt Wizard, but provides you with advanced licensing options contained in SxAF, such as support for Product Editions (see chapter *3.3.11*) as well as project and user management (see chapter *3.6*).

This solution is the right choice if the you need these advanced options, plus project management.

See chapter *3* for detailed description of all AutoCrypt SxAF features.

1.1.3. Command Line Tools (AC_Tool, SmrxProg)

The biggest advantage of the command line tools: The protection process can be controlled within other applications or batch-files. This allows a high grade of automation. Projects can be exported from AutoCrypt Wizard or AutoCrypt SxAF for usage with the AC_Tool and SmrxProg.

AC_Tool manages the application protection. SmrxProg manages the CRYPTO-BOX configuration (formatting): It writes the desired licensing information to the CRYPTO-BOX.

If you want to automate the protection and/or deeply integrate it into your specific distribution strategy, then AC_Tool/SmrxProg is the right choice.

See chapter *4* for further details.

SmrxProg is available for Linux and macOS, too. Please refer to the readme files in the corresponding “Smarx OS 4 Linux”/”Smarx OS 4 Mac” package.MyMARX registration and a valid Support Contract are required to download the Linux/macOS packages.

Download the latest Application Notes: www.marx.com/en/support/documents

1.2. AutoCrypt Download and Installation

MARX offers 2 different packages which include AutoCrypt. Which package you prefer depends from the AutoCrypt variant you are going to use (see chapter *1.1* for a comparison):

- The **Smarx OS Professional Protection Kit (PPK)** contains ALL available AutoCrypt variants, including AutoCrypt SxAF. See chapter 1.2.1 for installing the PPK.
- If the functionality of the AutoCrypt Wizard is sufficient for you, or you only need the latest release of AutoCrypt Wizard and the command line tools (AC_Tool, SmrxProg, RU_Tool), then the **AutoCrypt Wizard Package** is the right choice. It is small-sized and portable – no installation is required. Jump to chapter 1.2.2 for continuing with the AutoCrypt Wizard package.

Visit the Download section at marx.com to download the latest Smarx OS Professional Protection Kit (PPK) setup or the AutoCrypt Wizard package (MyMARX registration and valid Support Contract required). For new customers, Economy Support is included for the first 45 days.

1.2.1. Smarx® OS Professional Protection Kit (PPK)

Visit marx.com and download the “Smarx OS PPK”. Then double-click the file to start installation. Once the installation has finished, attach the CRYPTO-BOX to your computer. Windows will automatically locate the CRYPTO-BOX drivers and install them. Once the PPK installation has finished, attach the CRYPTO-BOX to your PC. Windows will locate the drivers and install the CRYPTO-BOX automatically.

The Smarx Professional Protection Kit (PPK) consists of the following components:

- Smarx PPK Control Center – the start menu which provides quick access to all available PPK components;
- AutoCrypt Wizard (see chapter *2*);
- Smarx Application Framework (SxAF) – a project oriented environment which automates protection of software, data and digital media and offers different licensing scenarios. AutoCrypt is a part of SxAF (see chapter *3.2*);
- Command line tools as alternative to AutoCrypt Wizard and AutoCrypt SxAF, especially for automation and script control within other applications (see chapters *4.1* and *4.2*);
- Tools for CRYPTO-BOX driver installation and diagnostic (see chapter *5*);
- Libraries and sample code for manual implementation into the application source code (see separate White Paper on API implementation at www.marx.com → Support → Documents → White Papers).

The “Smarx Compendium” contains a description of all PPK components and options. You can download the Compendium from www.marx.com → Support → Documents.

1.2.2. AutoCrypt Wizard Package

Visit marx.com and download the “AutoCrypt Wizard Package”. Then Unzip the archive into any folder on your computer.

Before you can use the AutoCrypt Wizard package, you have to make sure that the CRYPTO-BOPX drivers are available on your system. If you have an Internet connection, Windows will install the required driver automatically when plugging the CRYPTO-BOX into the USB port. Alternatively you can use our **CBUSetup** tool to install the drivers – download at www.marx.com → Support → Downloads → Driver and Diagnostic Tools.

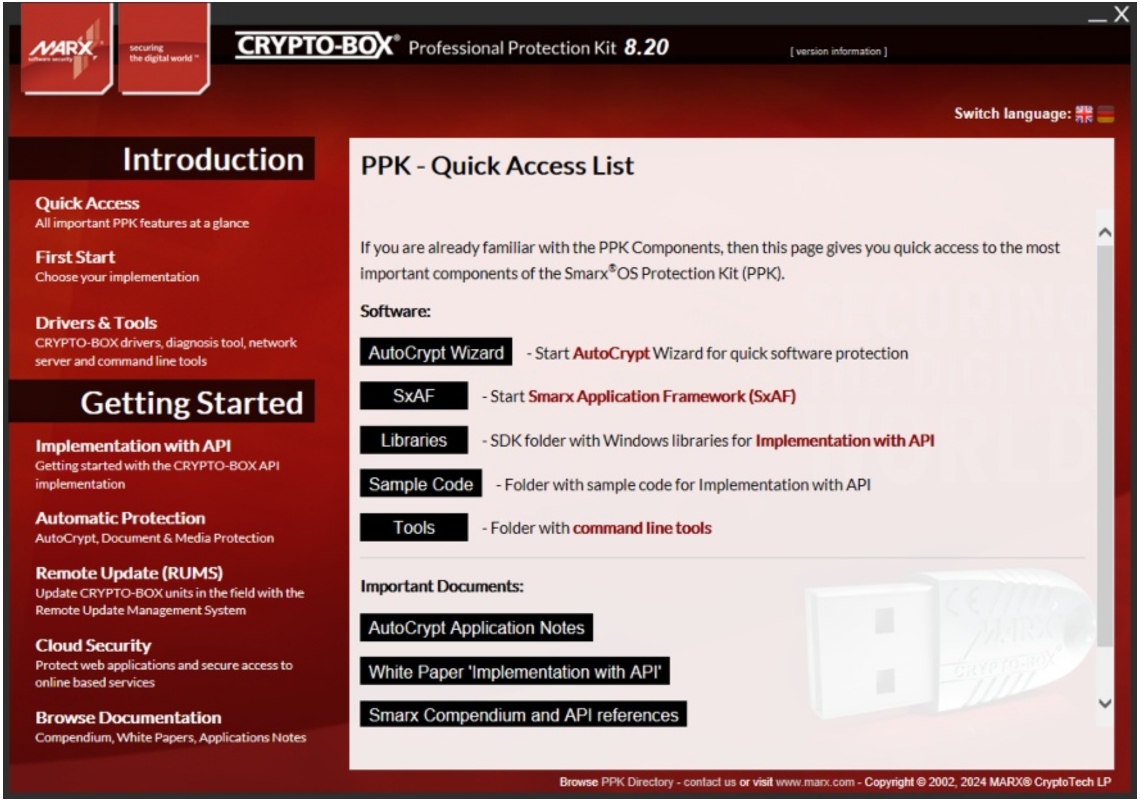
2. AutoCrypt Wizard

2.1. Starting AutoCrypt Wizard

Depending on the package you have chosen (see chapter *1.2*), follow the instructions in chapter *1.2.1* for the **PPK**, or in chapter *1.2.2* for the **AutoCrypt Wizard package**.

2.1.1. Smarx® OS Professional Protection Kit (PPK)

After the PPK installation (see *1.2.1*), click on the “MARX PPK Control Center” desktop shortcut. The Control Center provides an overview of the components installed, including a brief introduction and links to these components. From the menu select "Quick Access", then "AutoCrypt Wizard".



MARX software security
securing the digital world ™
CRYPTO-BOX® Professional Protection Kit 8.20
[version information]
Switch language:

Introduction

Quick Access
All important PPK features at a glance

First Start
Choose your implementation

Drivers & Tools
CRYPTO-BOX drivers, diagnosis tool, network server and command line tools

Getting Started

Implementation with API
Getting started with the CRYPTO-BOX API implementation

Automatic Protection
AutoCrypt, Document & Media Protection

Remote Update (RUMS)
Update CRYPTO-BOX units in the field with the Remote Update Management System

Cloud Security
Protect web applications and secure access to online based services

Browse Documentation
Compendium, White Papers, Applications Notes

PPK - Quick Access List

If you are already familiar with the PPK Components, then this page gives you quick access to the most important components of the Smarx®OS Protection Kit (PPK).

Software:

AutoCrypt Wizard - Start AutoCrypt Wizard for quick software protection

SxAF - Start Smarx Application Framework (SxAF)

Libraries - SDK folder with Windows libraries for Implementation with API

Sample Code - Folder with sample code for Implementation with API

Tools - Folder with command line tools

Important Documents:

AutoCrypt Application Notes

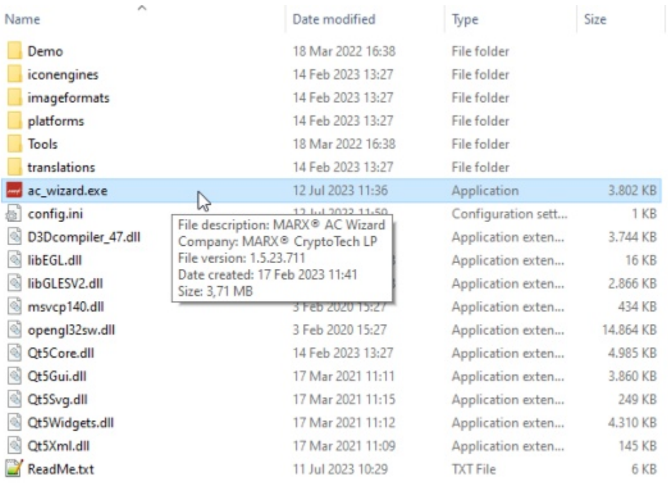
White Paper 'Implementation with API'

Smarx Compendium and API references

Browse PPK Directory - contact us or visit www.marx.com - Copyright © 2002, 2024 MARX® CryptoTech LP

2.1.2. AutoCrypt Wizard Package

If you have chosen the AutoCrypt Wizard package (see chapter *1.2.2*), start the file ac_wizard.exe in the unzipped archive.



Name	Date modified	Type	Size
Demo	18 Mar 2022 16:38	File folder	
iconengines	14 Feb 2023 13:27	File folder	
imageformats	14 Feb 2023 13:27	File folder	
platforms	14 Feb 2023 13:27	File folder	
Tools	18 Mar 2022 16:38	File folder	
translations	14 Feb 2023 13:27	File folder	
ac_wizard.exe	12 Jul 2023 11:36	Application	3.802 KB
config.ini	12 Jul 2023 11:50	Configuration sett...	1 KB
D3Dcompiler_47.dll		Application exten...	3.744 KB
libEGL.dll		Application exten...	16 KB
libGLESV2.dll		Application exten...	2.866 KB
msvcp140.dll	3 Feb 2020 15:27	Application exten...	434 KB
opengl32sw.dll	3 Feb 2020 15:27	Application exten...	14.864 KB
Qt5Core.dll	14 Feb 2023 13:27	Application exten...	4.985 KB
Qt5Gui.dll	17 Mar 2021 11:11	Application exten...	3.860 KB
Qt5Svg.dll	17 Mar 2021 11:15	Application exten...	249 KB
Qt5Widgets.dll	17 Mar 2021 11:12	Application exten...	4.310 KB
Qt5Xml.dll	17 Mar 2021 11:09	Application exten...	145 KB
ReadMe.txt	11 Jul 2023 10:29	TXT File	6 KB

File description: MARX® AC Wizard
Company: MARX® CryptoTech LP
File version: 1.5.23.711
Date created: 17 Feb 2023 11:41
Size: 3,71 MB

2.2. Protecting Applications

2.2.1. Steps for Protecting Applications with AutoCrypt Wizard

It is recommended you follow these steps when protecting your application with AutoCrypt:

1. Create a new AutoCrypt Wizard project. A project includes all information that is used for programming the CRYPTO-BOX.
2. Add the application(s) you want to protect to the project and choose your desired protection settings and licensing options. See chapter *2.2.5* for an overview about available Licensing Options (data object types).
3. Protect the application(s).
4. Use the Format option (see chapter *2.3*) to format your CRYPTO-BOX units with the project settings.
5. Additionally, you can export your AutoCrypt Wizard project for usage with the command line tools AC_Tool and SmrxProg to automate application protection and CRYPTO-BOX formatting (see chapter *4*).
6. If you plan to update licensing options (e.g., expiration date or usage counter) in the CRYPTO-BOX at your end-user's site, you can create the Remote Update Utility for this project and ship it along with the CRYPTO-BOX to your end-users (see chapter *2.5* for more information).
7. Test the protection and your selected licensing options carefully.
8. Ship your protected application together with the CRYPTO-BOX and the necessary supplemental files (drivers, network server in case of network licensing). MARX provides an easy-to-use redistributable setup for this. See chapter *5* for more details.

2.2.2. Creating New Projects or Selecting Existing Projects

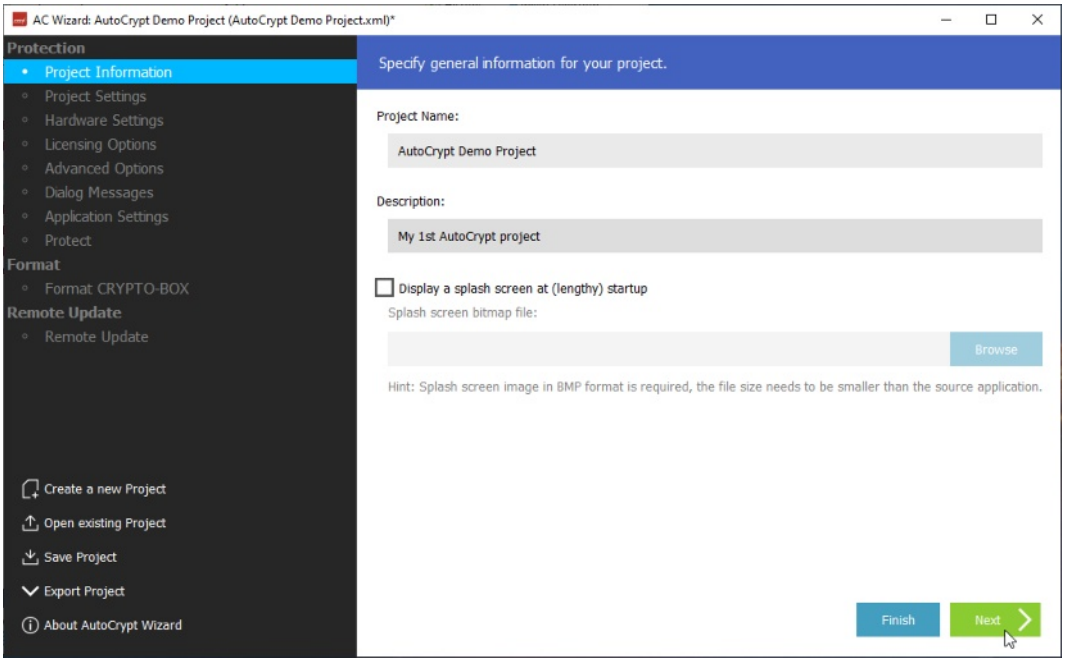
On the Wizard main screen, you can either *create a new project* or *work with an existing project*. We strongly recommend to start with the AutoCrypt demo project: Click on “Open existing project” and choose the “AutoCrypt Demo Project.xml” file.

Now, enter a project name and description (optional).

If desired, you may also add a splash screen which is displayed before the protected application starts. Select a graphics file – it must be in bitmap (.bmp) format, and the size of the file must be smaller than the source application you are going to protect.

If you do not have an image in bitmap (.bmp) format, you can use the Windows application “Paint” to save any image file in .bmp format. If required, you can reduce the resolution or the color depth to reduce the file size.

Important: Splash screens are not supported for .Net 6.0 (or higher) apps, splash screen settings will be ignored in that case!



Specify general information for your project.

Project Name:

AutoCrypt Demo Project

Description:

My 1st AutoCrypt project

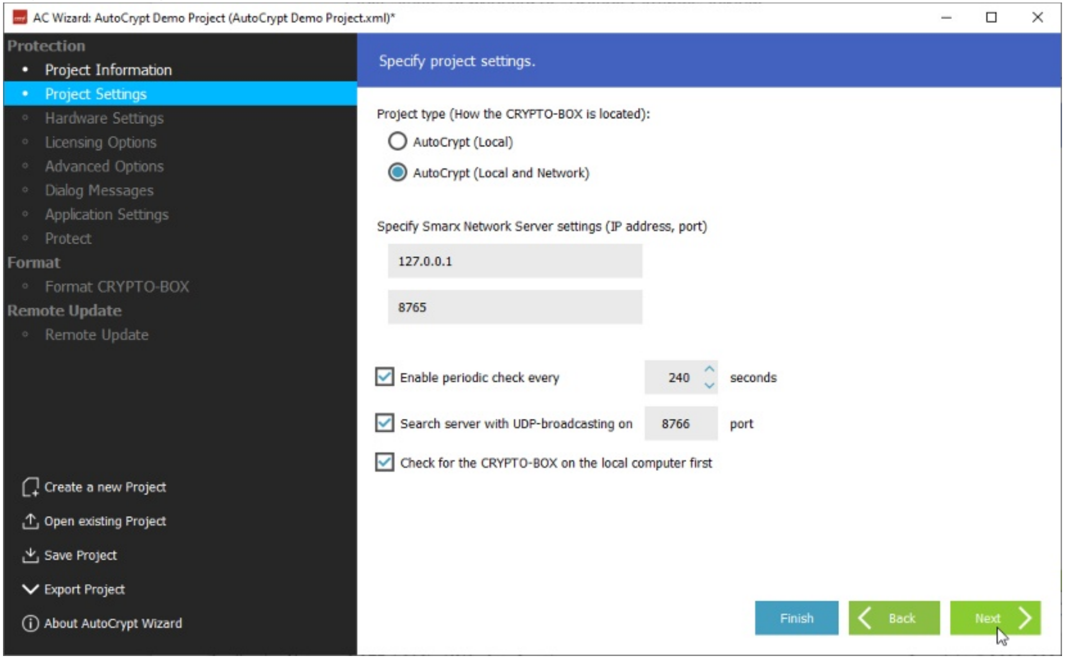
□ Display a splash screen at (lengthy) startup

Splash screen bitmap file:

Hint: Splash screen image in BMP format is required, the file size needs to be smaller than the source application.

Click “Next” to proceed to “Project Settings” screen.

2.2.3. Project Settings



Specify project settings.

Project type (How the CRYPTO-BOX is located):

○ AutoCrypt (Local)

● AutoCrypt (Local and Network)

Specify Smarx Network Server settings (IP address, port)

127.0.0.1

8765

☑ Enable periodic check every 240 seconds

☑ Search server with UDP-broadcasting on 8766 port

☑ Check for the CRYPTO-BOX on the local computer first

First, select the project type. AutoCrypt projects can be either local (applications are protected with

CRYPTO-BOX on local PC) or network-based (applications are protected with CRYPTO-BOX located on the Server). Select “AutoCrypt (Local)”, if you want to use local protection and licensing only. Select “AutoCrypt (Local and Network)” for network protection and licensing or combined Local/Network approach.

When selecting “AutoCrypt (Local and Network)”, you can specify Network Server settings (IP address, port); by default, the IP address is “127.0.0.1” (this is the local IP address of your computer), the port is 8765. You can also submit a computer name (e.g. “PC-517”) instead of an IP address. If the server can not be found at the specified IP address, the protected application will open a dialog upon startup, asking for server settings.

When the periodic check option is enabled, CRYPTO-BOX presence is checked for within the defined time-out while the application is running. If the CRYPTO-BOX is not found, the “Protection error” dialog message (see chapter *2.2.7 Dialog Messages*) will be shown. If the CRYPTO-BOX is still missing the application will be terminated. The value should not be less than 60 seconds (more is recommended) for one application and increased when several applications are protected with one CRYPTO-BOX, or when network mode is used.

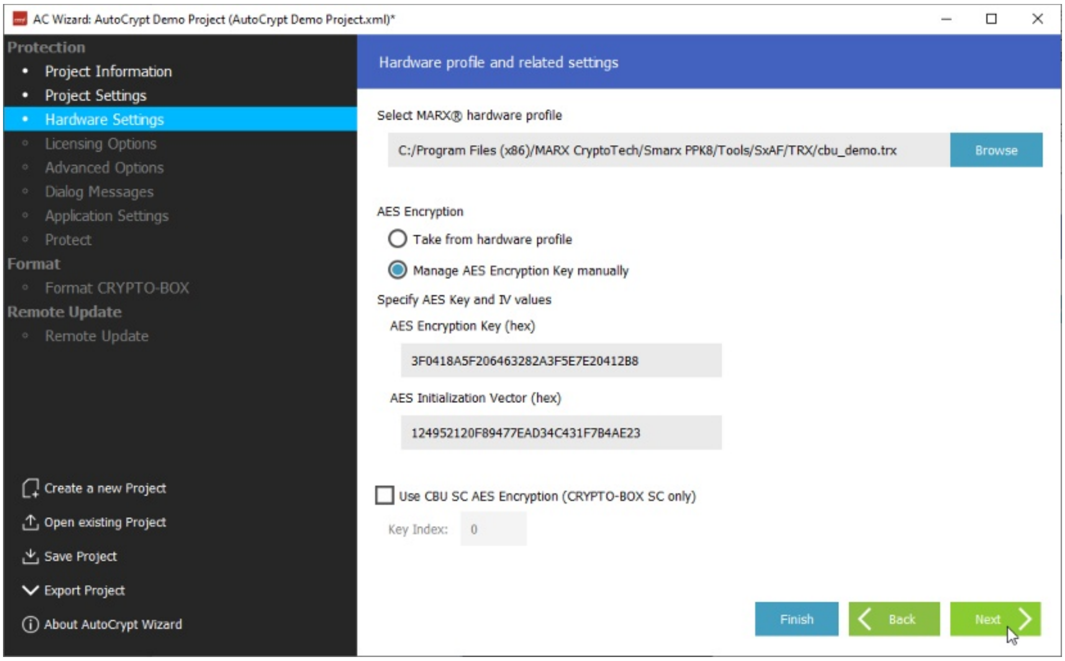
The last two options are available in network mode only: search server with UDP broadcasting means that the protected application will automatically search for available CRYPTO-BOX Servers (the server must be on the same sub-net). Default port is 8766, UDP port of the server can be changed in server configuration if required (see White Paper “Network Licensing” for more information on network configuration). When the option “Check for the CRYPTO-BOX on the local computer first” is activated, the protected application will look for locally attached CRYPTO-BOX first before it starts a network search.

Click “Next” to proceed to “Hardware Settings” screen.

2.2.4. Hardware Settings

You need to select a hardware profile for the project. This profile contains the access codes to be used by the protected application to access the CRYPTO-BOX. Every MARX customer will get his customer specific TRX file. Select the "cbu_demo.trx" profile for the CRYPTO-BOX that was shipped with the Evaluation Kit. If you already use customer specific CRYPTO-BOX units, select the profile which you received along with your first CRYPTO-BOX delivery from MARX.

See the “TRX File” White Paper at marx.com → Support → Documents for further details on hardware profile handling.



AC Wizard: AutoCrypt Demo Project (AutoCrypt Demo Project.xml)*

Hardware profile and related settings

Select MARX® hardware profile: C:/Program Files (x86)/MARX CryptoTech/Smarx PPK8/Tools/SxAF/TRX/cbu_demo.trx

AES Encryption
○ Take from hardware profile
● Manage AES Encryption Key manually

Specify AES Key and IV values
AES Encryption Key (hex): 3F0418A5F206463282A3F5E7E20412B8
AES Initialization Vector (hex): 124952120F89477EAD34C431F7B4AE23

□ Use CBU SC AES Encryption (CRYPTO-BOX SC only)
Key Index: 0

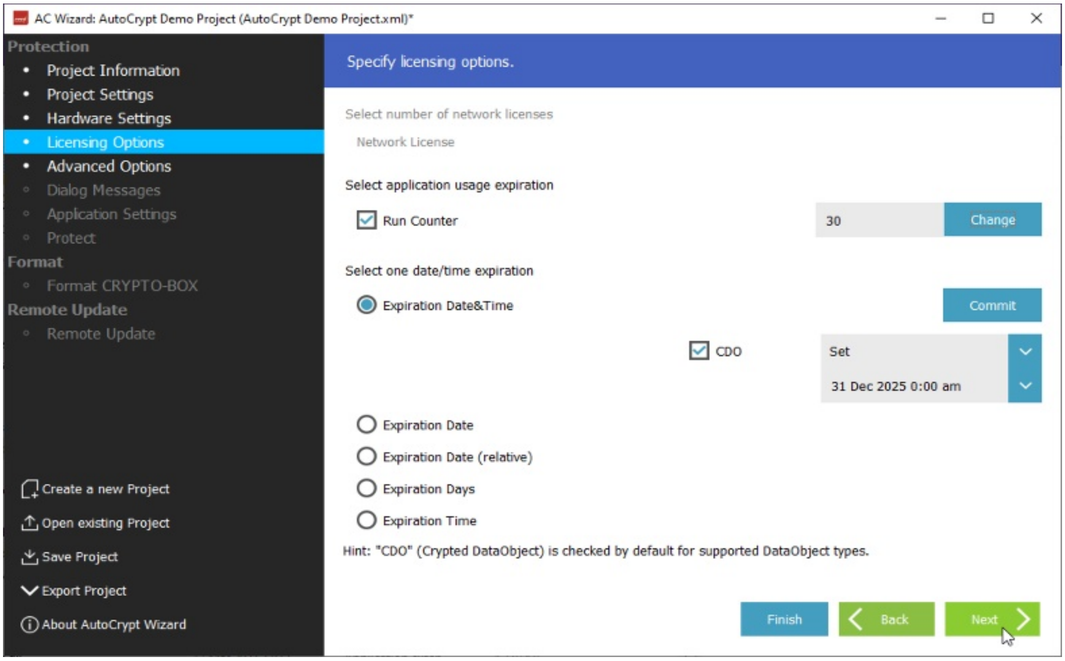
The “Manage AES Encryption Key” option allows you to define values for the AES/Rijndael Private Key and the Initialization Vector used to encrypt the application. You can either take the predefined values (which were assigned to all your CRYPTO-BOX units at MARX production stage) by choosing the “Take from hardware profile” option, or you can manage the encryption key settings manually.

All protected applications will be compressed and encrypted by default. For application encryption, the AES Private Key of the CRYPTO-BOX is used. For more details on CRYPTO-BOX AES encryption, please refer to the Smarx OS Compendium, chapter 10.
With the CRYPTO-BOX SC it is possible to have separate AES keys for every application/ set of applications. Check the "Use CBU SC AES encryption" field to do so and specify a key index value between 0 and 5. For more information on CRYPTO-BOX SC features, please refer to marx.com.

Important Note:
If the encryption key value inside the CRYPTO-BOX does not match to the key value with was specified in the AutoCrypt project, the protected application will not start!
If you want to protect multiple applications with one CRYPTO-BOX, you have to make sure that all projects use the same encryption key. In that case, either choose “Take from hardware profile” option, or select the same key/IV value for all these projects. Plus you have to make sure to choose different Partitions for storing your license information in the CRYPTO-BOX (see chapter *2.2.8*)
Alternatively, you can use the AutoCrypt SxAF, which allows to add multiple applications to the same project (see chapter *3.3.5*)
An exception is the CRYPTO-BOX SC which allows to define an individual AES key for every project: The key index can be between 0 and 5. More keys are possible on request, please contact us.

Click “Next” to proceed to “Licensing Options” screen.

2.2.5. Licensing Options



Specify licensing options.

Select number of network licenses

Network License

Select application usage expiration

☑ Run Counter 30

Select one date/time expiration

● Expiration Date&Time

☑ CDO Set 31 Dec 2025 0:00 am

○ Expiration Date

○ Expiration Date (relative)

○ Expiration Days

○ Expiration Time

Hint: "CDO" (Crypted DataObject) is checked by default for supported DataObject types.

Here you can define required licensing logic for your application by choosing licensing data objects of predefined types, such as Network License, Expiration Date, Run Counter, etc.

If you have chosen the network mode for your project (see chapter *2.2.3*) you will have the “Network License” option enabled which allows you to set up the number of network licenses for the application. It defines how many instances of your application can run in the network at the same time. See White Paper “Network Licensing” more information on Network License Management.

Below the network licensing option you can select between several license expiration options (data objects) for your application:

- To add a licensing option, check corresponding check box or radio button.
- To edit the value of a data object, press related "Change"/"Commit" button.
- To remove the data object, click the check box or radio button near data object.

> For the most DataObjects you have the choice between "CDO" (Crypted DataObject) and standard DataObjects ("CDO" deactivated). CDO offers additional protection against manipulations, so we recommend to keep this option activated. If you plan to combine AutoCrypt and API implementation and additionally want to query these Crypted DataObjects via API commands, please check the sample code in our PPK and corresponding readme files for CDO compatibility. Not all API libraries offer CDO support, eg. libraries for older or exotic compilers. Please contact our Technical Support for any questions.

The following types of licensing data objects are supported:

"Network License"	Number of network licenses for the application: Defines how many instances of your application can run in the network at the same time. **Important:** License counters are supported by the CRYPTO-BOX SC and CRYPTO-BOX XS models only. The CRYPTO-BOX Versa also offers network support, but the number of network licenses is always unlimited.
"Run Counter"	Number of application executions (runs);
"Expiration Date & Time"	Exact date and time when the protected application is going to expire, for example "31 JAN 2025 0:00";
"Expiration Date"	Fixed expiration date, submitted in the format "31 JAN2025". This data object type is obsolete and only preserved for compatibility purposes, we recommend to use "Expiration Date & Time" instead.
"Expiration Date (relative)"	Specifies the number of days the application is allowed to be used **from the first run**. i.e. counter is activated on the first application launch.
"Expiration Days"	"Expiration Days" is a flexible expiration date, submitted as number of days the application is allowed to be used, starting **from the day the CRYPTO-BOX was formatted**.
"Expiration Time"	Real-time expiration, submitted as a period of allowed application usage (in seconds).

> You can update the licensing data in the CRYPTO-BOX later with Remote Update. See chapter *2.5* for details.

Click "Next" to proceed to "Advanced Options" screen.

2.2.6. Advanced Options

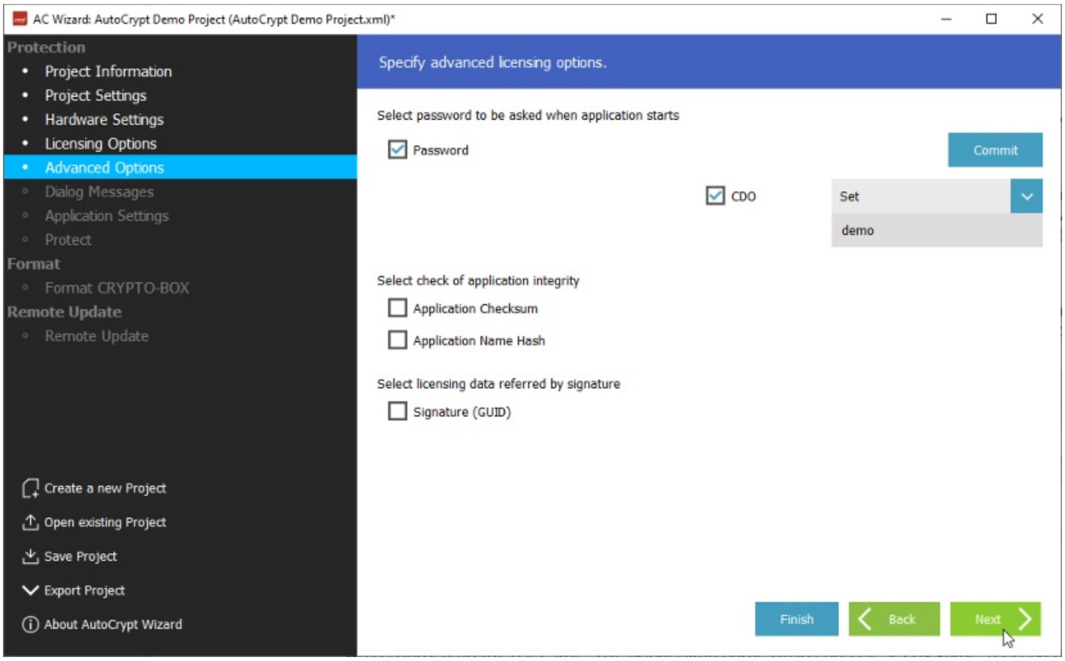
Password
This data object defines an application password, which will be required every time the application is launched. This feature can be useful for additional security, or as a "nag screen" when making demo versions.

Application Checksum
This data object defines a checksum, calculated for the protected application file. Its value will be stored in the CRYPTO-BOX and used to check if the application was manipulated or corrupted, as well as prevent unauthorized modification. The checksum cannot be set by value; it is calculated automatically after successful application protection.

Application Name Hash
This data object is specific to AutoCrypt projects: it contains a hash value calculated for the name of protected application. This hash value will be stored in the CRYPTO-BOX and used to check if the application

was renamed on the user side. The application name hash can't be set by value; it is calculated after successful application protection.

> Application Checksum and Application Name Hash should not be used if you plan to update your application regularly, because the new application will not be compatible with existing Checksum/Hash settings stored in the CRYPTO-BOX!


AC Wizard: AutoCrypt Demo Project (AutoCrypt Demo Project.xml)*

Specify advanced licensing options.

Select password to be asked when application starts

☑ Password

☑ CDO   Set
demo

Select check of application integrity

☐ Application Checksum

☐ Application Name Hash

Select licensing data referred by signature

☐ Signature (GUID)

Signature (GUID)
This option is mainly for software developers: It allows you to store an individual data block (up to 16 bytes length) in the CRYPTO-BOX memory (e.g. customer specific data) and provide it directly to your protected application – with minimal efforts! The advantage of this approach: there is no knowledge about the CRYPTO-BOX API necessary – just copy the code snippet from our sample source code to your application source code.

After license validation was successful, the protected application will read the data from the CRYPTO-BOX, decrypts it and writes it to a buffer signed with a unique signature which can be identified and read by the code you implemented into your application (see above).
A readme file with instructions and sample code can be found in the Protection Kit:

[Smarx OS PPK root folder]\SmarxOS\API\Win\Samples\ReadMemoryBySignature

Click "Next" to proceed to the "Dialog Messages" screen.

2.2.7. Dialog Messages

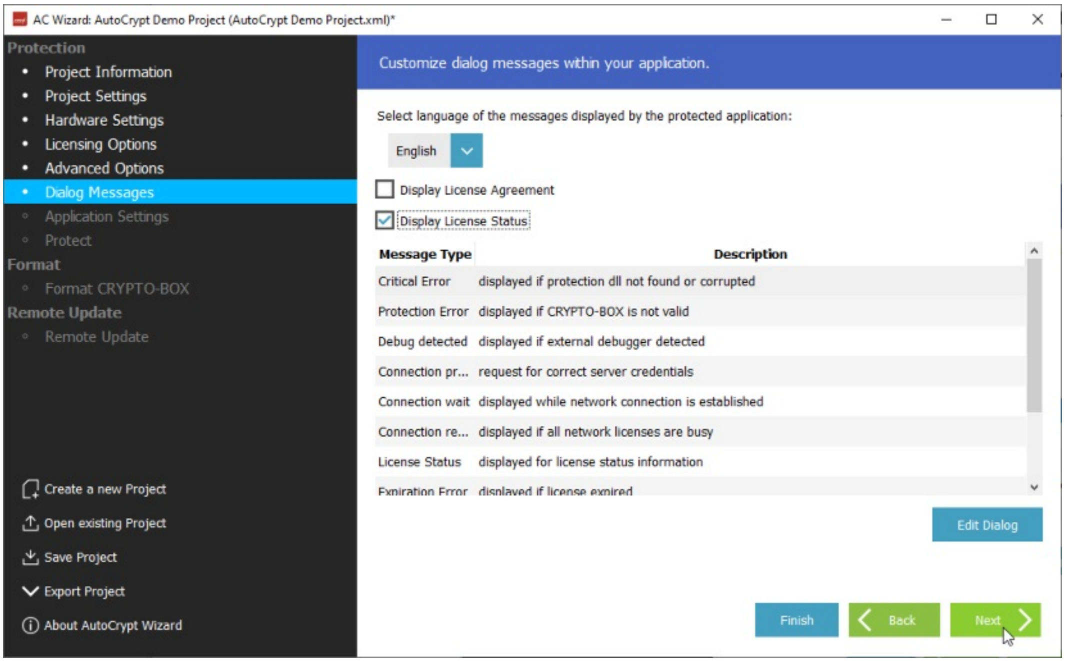
During protected application execution, some message dialogs can be displayed, for example: license status, protection errors/warnings, etc.

You can adjust these messages according to your needs: First, select the predefined language (English or German), then choose the dialog message you want to modify and click the "Edit Dialog" button. Next, you can set the message title (caption) and the body text. If you do not wish to have messages displayed, leave the title and body empty.

> If you need the dialog messages in another language, just overwrite one of the predefined settings (e.g. German) with messages in your desired language. Alternatively you can edit the message texts directly in the project XML file.

If the “Display License Agreement” check box is enabled, a License Agreement will be displayed before the application starts. You may add a customer specific text here.

If the “Display License Status” check box is enabled, the protected application will display the license status before it starts and indicate how often or how many more days the application may be executed. If no data objects with licensing options were defined (see chapter *2.2.5*), the “Display License Status” checkbox will be unavailable.



Customize dialog messages within your application.

Select language of the messages displayed by the protected application:

English

☐ Display License Agreement

☑ Display License Status

Message Type	Description
Critical Error	displayed if protection dll not found or corrupted
Protection Error	displayed if CRYPTO-BOX is not valid
Debug detected	displayed if external debugger detected
Connection pr...	request for correct server credentials
Connection wait	displayed while network connection is established
Connection re...	displayed if all network licenses are busy
License Status	displayed for license status information
Expiration Error	displayed if license expired

2.2.8. Application Settings

Here you can add your source application (.exe or .dll file) to the project.

First, specify the number of the CRYPTO-BOX partition in which the application protection settings (data objects) will be stored. The number of the partition must be in range between 101 and 65535. We recommend to leave the predefined settings.

Now select the original application that is to be protected – it may be a Windows EXE or a DLL file.

For testing purposes, you can use one of the predefined test applications: Click “Browse” to open the samples application folder.

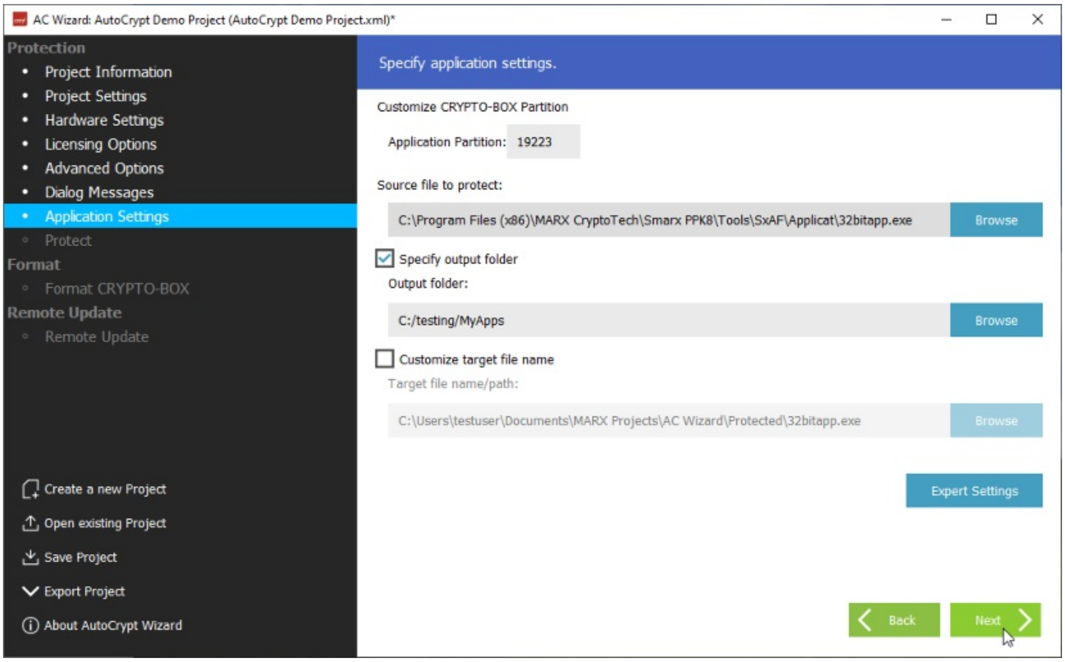
If you use the **PPK**, the sample applications are at:
[Smarx OS PPK root folder]\Tools\SxAF\Applicat

If you use the **AutoCrypt Wizard Package**, the sample applications are at:
[AutoCrypt Package root folder]\Demo

Select the desired Win32/Win64 sample application or Win32Dll.dll/Win64Dll.dll for sample DLL. You can test the protected DLL with the DllLauncher.exe/ DllLauncher64.exe program.

Now choose where the protected application will be stored:

- When checking the “Specify output folder”, the application will be stored in the selected target folder under the same name as the source application.
- Select “Customize target file name” if you also want to modify the name of the protected application.


AC Wizard: AutoCrypt Demo Project (AutoCrypt Demo Project.xml)*

Specify application settings.

Customize CRYPTO-BOX Partition

Application Partition: 19223

Source file to protect:

C:\Program Files (x86)\MARX CryptoTech\Smarx PPK8\Tools\SxAF\Applicat\32bitapp.exe

☑ Specify output folder

Output folder:

C:/testing/MyApps

□ Customize target file name

Target file name/path:

C:\Users\testuser\Documents\MARX Projects\AC Wizard\Protected\32bitapp.exe

If your source application is a .NET based application, you will have further options under "**Expert Settings**":

- The **"Obfuscate"** option offers automatic .NET application obfuscation by using the open source software "Obfuscar" in background. Refer to the Obfuscar webpage for further details.
- The **"Anti-Dump Protection"** option hinders dumping of .NET applications. We recommend to leave it activated. It should be activated only in case the protected application does not start.
- The **"Fix Assembly Location"** option resolves the issue that .NET applications might not work properly after protection with AutoCrypt if they use Location property to get path of executing assembly, e.g. Assembly.GetExecutingAssembly().Location will return empty string. This happens because the protected assembly does not exist in the file system, since our AutoCrypt loader loads it from the memory.
- The **"Loader version"** option determines which type of loader is used to protect a .NET application. This can be useful when the protected .NET application does not start or if there is a false-positive detection of the protected application by your Antivirus solutions. The following options are available:
  - DEFAULT – AutoCrypt will automatically detect the type of .NET application using .NET_CORE for newer .Net apps (if there is an .exe, .dll and .runtimeconfig.json with the same name in the source file folder), and .NET_20 for all other applications.
  - .NET_20 – Force using older loader for .NET 2.x and 3.x applications (some .NET 4.5+ applications might not work with these settings)
  - .NET_45 – Force using loader for .NET 4.x applications
  - .NET_48 – Force using loader for .NET 4.8 applications requiring TLS 1.2 support
  - .NET_Core – Force using loader for .NET Core applications (.NET 6.0 and higher)
  - .NET_SPLIT_LOAD – Force using experimental loader to avoid false-positive detections by Antivirus solutions. Supports .Net 4.x applications only (for .Net 6 and higher this functionality is already integrated in the .NET_Core loader).
- The **"Console Application"** option is only available when .NET_Core or .NET_SPLIT_LOAD loader is selected and has to be used when protecting a console application service.

Important notes for .Net 6.0+ (.Net Core) applications:

1. For .Net 6.0+ applications, please always specify the corresponding .dll file as the original application (see 2.2.8), not the .exe! The .exe in .Net 6.0+ is just a loader that loads the actual application which is in the .dll file. AutoCrypt protects the .dll and replaces the .exe with its own loader. If the .exe is specified, AutoCrypt will attempt to protect the associated .dll if a .NET 6.0+ application was detected.
2. For .Net 6.0+ applications, the target file must have the same name as the original file (see 2.2.8), otherwise the protected application will not start!
3. In case of .Net 6.0+ applications you can specify the same folder for both the original application and the protected application. In that case AutoCrypt replaces the original application with the protected files and moves the original files to the _backup folder. If you choose a different destination path, always remember to copy the associated runtime configuration files (.json files) to the destination directory, otherwise the protected application will not start.
4. .Net 6.0+ applications can be protected only with STANDARD or DOTNET_Core option, with all other settings the protected application will not start!
5. AutoCrypt cannot protect .Net 5.0 applications out of the box! Either upgrade to a newer .Net version (6.0 or higher), or use the workaround described in the AC_Tool Readme file to adjust the .Net version in your runtime configuration file. See chapter 4.1 for more information.

Important notes on the .NET_SPLIT_LOAD option:

.NET applications protected with the .NET_SPLIT_LOAD option require the installation of our AC_Loader component on the target system! AC_Loader can be found in the PPK:

[Smarx PPK root folder]\Redistributable\AC_Tool\AC_loader.msi

Hint: AC_Loader includes the installation of the CRYPTO-BOX driver, so you do not need to install them separately with CBUSetup. Please refer to the AC_Loader readme file for further details. AC_Loader supports silent installation option of .msi packages with the /Quiet option.

If your source application is not a .NET based application, then the options in “Expert Settings” are ignored.

Click “Next” to proceed with application protection.

2.2.9. Protecting the Application

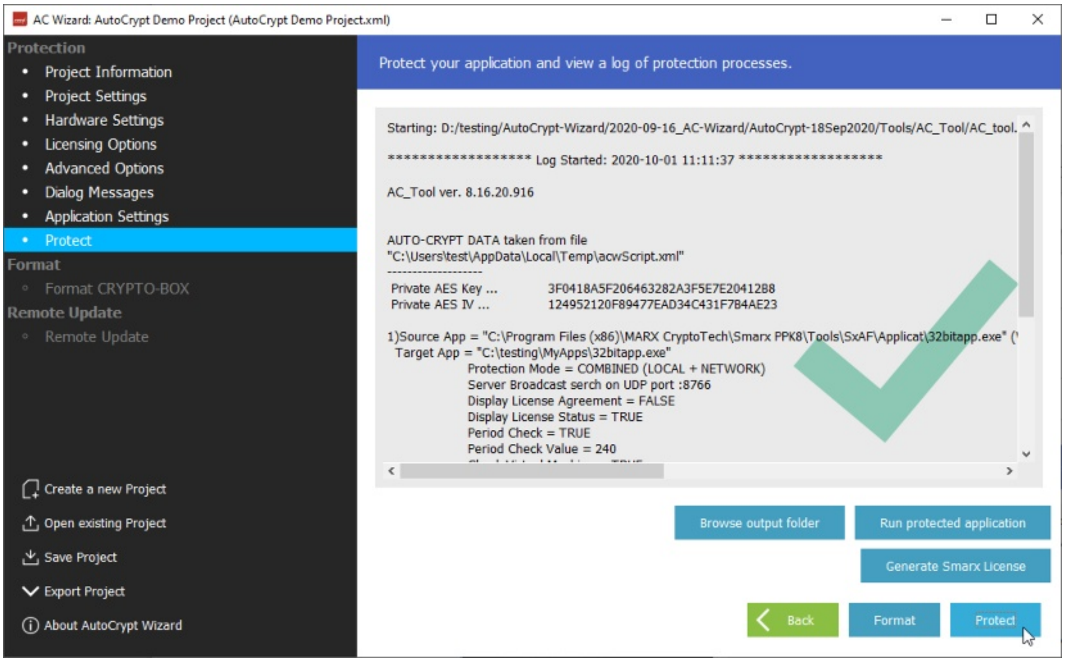
Make sure that the CRYPTO-BOX which fits to the hardware profile you specified under “Hardware Settings” (see *2.2.4*) is plugged to the USB port and click the “Protect” button.

Now the original application file will be protected. The protected application and a functional module (.dll file) will be stored in the output folder specified in the previous step. The wizard will also display a log output. If the protection was successful, a check mark will be displayed.

AutoCrypt will automatically compress and encrypt the protected application (AES Rijndael Private Key of the CRYPTO-BOX is used for that, see chapter *2.2.4*). Additionally, it will be protected against debugging.

If the protected application will not work, please contact us – in almost all cases we can modify AutoCrypt to work with your executable.

If your virus scanner treats your protected application as infected, please refer to chapter *6* (see FAQ entry number 5) for possible solutions.



```
Starting: D:/testing/AutoCrypt-Wizard/2020-09-16_AC-Wizard/AutoCrypt-18Sep2020/Tools/AC_Tool/AC_tool.

****************** Log Started: 2020-10-01 11:11:37 ******************

AC_Tool ver. 8.16.20.916


AUTO-CRYPT DATA taken from file
"C:\Users\test\AppData\Local\Temp\acwScript.xml"
--------------------
 Private AES Key ...          3F0418A5F206463282A3F5E7E2041288
 Private AES IV ...           124952120F89477EAD34C431F7B4AE23

1)Source App = "C:\Program Files (x86)\MARX CryptoTech\Smarx PPK8\Tools\SxAF\Applicat\32bitapp.exe" ('
  Target App = "C:\testing\MyApps\32bitapp.exe"
                Protection Mode = COMBINED (LOCAL + NETWORK)
                Server Broadcast serch on UDP port :8766
                Display License Agreement = FALSE
                Display License Status = TRUE
                Period Check = TRUE
                Period Check Value = 240
```

Now click the “Format” button to proceed with CRYPTO-BOX formatting.

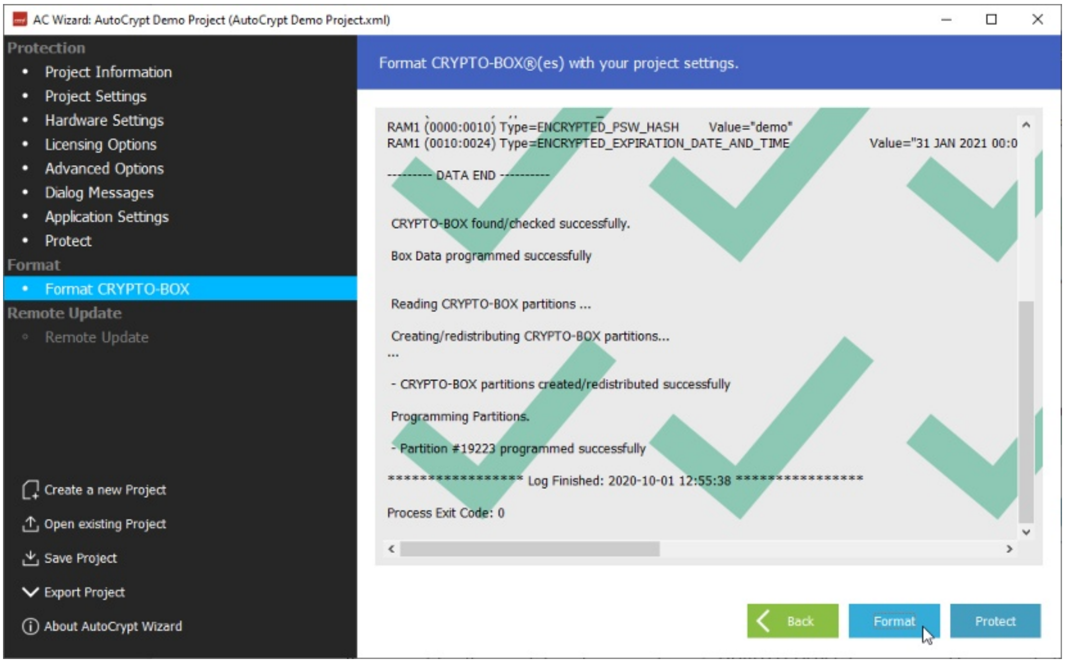
> You can automatize the protection of your applications with AutoCrypt by using AC_Tool.exe. This utility is a command line version of AutoCrypt which can be controlled within other applications using command line switches. See chapter 4.1 *AutoCrypt - Command Line Version* for more details.

2.3. Format CRYPTO-BOX®

After your application was protected successfully, it is time to format (program) the CRYPTO-BOX with the settings you specified in your project.

Click the “Format” button. You can check the process in the log output. If the protection was successful, a check mark will be displayed.

> The command line tool SmrxProg allows you to automate CRYPTO-BOX formatting. See chapter *4.2 SmrxProg - Command Line based CRYPTO-BOX® Formatting* for details.



Format CRYPTO-BOX®(es) with your project settings.

```
RAM1 (0000:0010) Type=ENCRYPTED_PSW_HASH      Value="demo"
RAM1 (0010:0024) Type=ENCRYPTED_EXPIRATION_DATE_AND_TIME          Value="31 JAN 2021 00:0

--------- DATA END ----------


 CRYPTO-BOX found/checked successfully.

 Box Data programmed successfully


 Reading CRYPTO-BOX partitions ...

 Creating/redistributing CRYPTO-BOX partitions...
...

 - CRYPTO-BOX partitions created/redistributed successfully

 Programming Partitions.

 - Partition #19223 programmed successfully

***************** Log Finished: 2020-10-01 12:55:38 ****************

Process Exit Code: 0
```

2.4. Exporting XML Script for Use with Command Line Tools

If you want to integrate application protection and CRYPTO-BOX formatting with your own administration/ distribution strategy, choose the "Export Project" in the menu bar on the lower left side. This exports the current project to a XML script file for further usage with the AC_Tool.exe and SmrxProg.exe Command line tools.

AC_Tool (for protecting applications, see chapter 4.1) and SmrxProg (for configuring CRYPTO-BOX modules, see chapter *4.2*) are console applications. They are controlled via command line switches and can be called up by applications or scripts.

2.5. Process Remote Updates

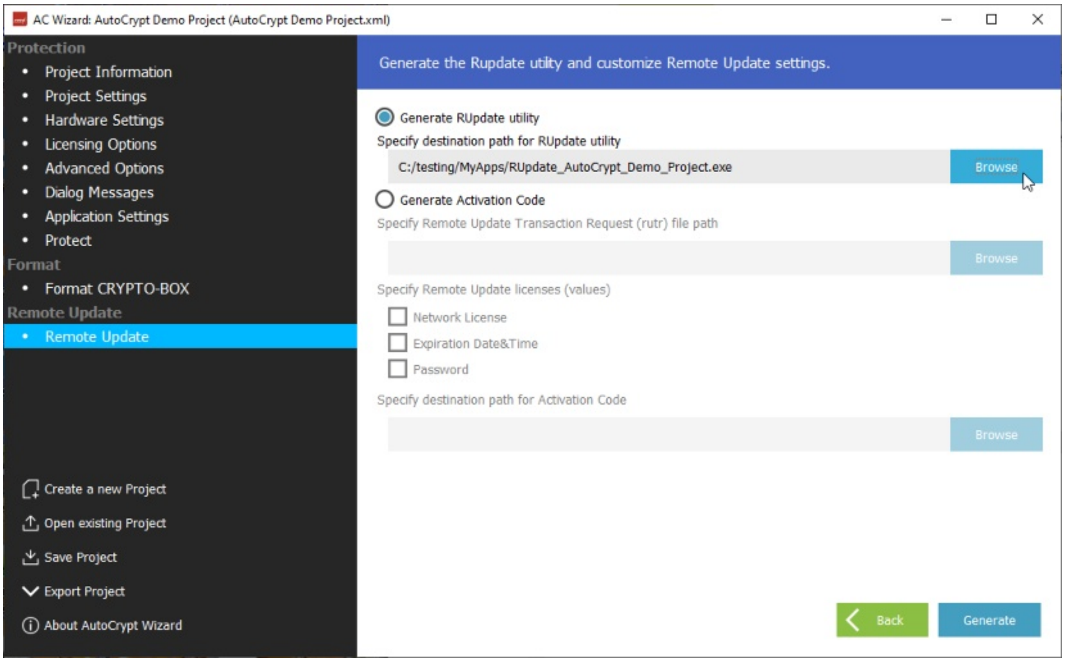
2.5.1. Creating Remote Update Tool

If you want to allow updates of your end user's CRYPTO-BOX later, you need to generate the Remote Update Tool. This program encapsulates the project and CRYPTO-BOX specific data and can be distributed to the end user together with CRYPTO-BOX.

To proceed with Remote Updates, choose the "Remote Update" option in the left navigation bar of the AutoCrypt Wizard.

Select the "Generate RUpdate utility" radio button. Now click "Browse" to select the path where the Remote Update Utility should be extracted to, and the file name. Then click the "Generate" button.

⚠ Generating the Remote Update Utility assumes a valid RUMS license which is available as an option. This license can be obtained from your MARX distributor in the form of an updated hardware profile (.TRX file). Please visit marx.com → Shop → Solutions → RUMS for details and pricing, or contact MARX for more information.
If no RUMS license is available, the message "Error: RUMS not licensed" will be displayed.


AC Wizard: AutoCrypt Demo Project (AutoCrypt Demo Project.xml)

Generate the Rupdate utility and customize Remote Update settings.

● Generate RUpdate utility
Specify destination path for RUpdate utility
C:/testing/MyApps/RUpdate_AutoCrypt_Demo_Project.exe

○ Generate Activation Code
Specify Remote Update Transaction Request (rutr) file path

Specify Remote Update licenses (values)
□ Network License
□ Expiration Date&Time
□ Password

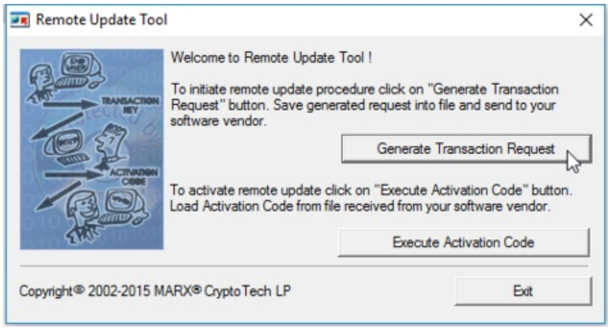
Specify destination path for Activation Code

Click “Browse output folder” - you will see 3 files (.exe file and language resource files). Provide these files to your end-user to process remote updates for his CRYPTO-BOX.

The Remote Update process can be controlled within other applications or script files. This allows a high grade of automation. Detailed information on this can be found in the “RUMS Application Notes” which are available on our web page: marx.com → Support → Documents.

2.5.2. Initiating Remote Updates (End-User)

To initiate a Remote Update, the **end user** connects the CRYPTO-BOX to his computer and starts the RUpdate Utility. Now he needs to click the “Generate Transaction Request” button:


Remote Update Tool

Welcome to Remote Update Tool !

To initiate remote update procedure click on "Generate Transaction Request" button. Save generated request into file and send to your software vendor.

To activate remote update click on "Execute Activation Code" button. Load Activation Code from file received from your software vendor.

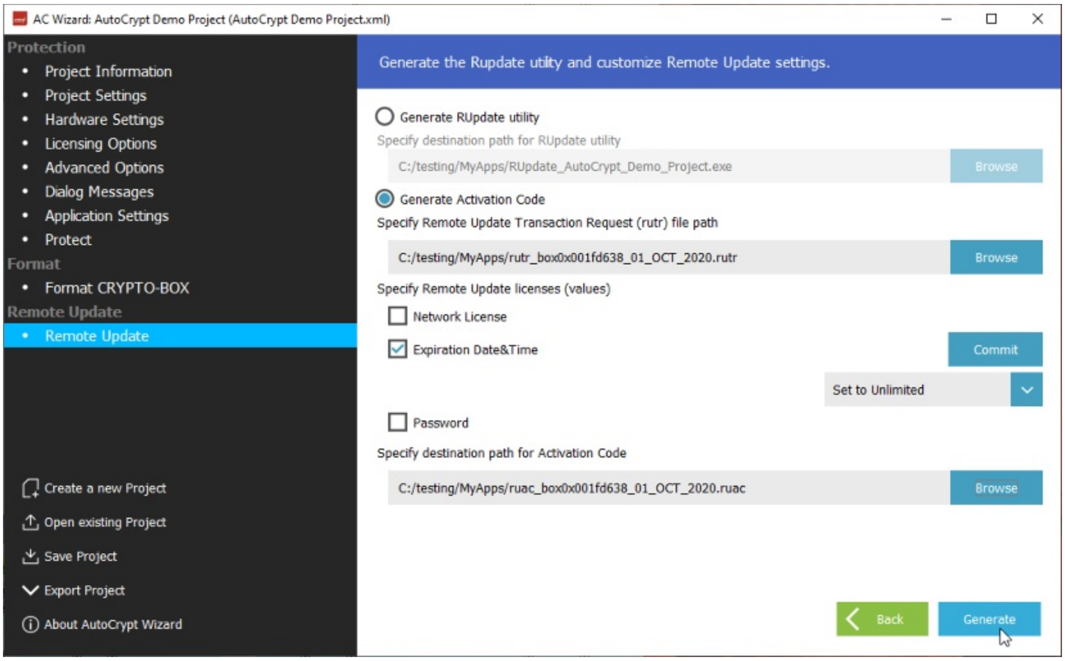
Copyright© 2002-2015 MARX® CryptoTech LP

A file with the extension .rutr will be created. This file needs to be sent back to the **software distributor** (For instance via Email).

During this process, a Transaction ID will be stored in the Transaction Request file as well as in the CRYPTO-BOX itself. This ensures that the update (which is generated by the software distributor later) will be valid only for the CRYPTO-BOX that has the proper Transaction ID.

2.5.3. Generate Activation Code

After retrieving the Transaction Request file from the end-user, start AutoCrypt Wizard again and choose the "Remote Update" option in the left navigation bar. Select the "Generate Activation Code" radio button and specify path to the Transaction Request (.rutr) file.
Below you can select the licensing options you want to change: check the desired option, and click "Change" to modify the data object. For instance, you can extend the run counter or expiration date, or even set it to unlimited.
Do not forget to set the out put folder where the Activation Code should be stored, or leave predefined settings.


AC Wizard: AutoCrypt Demo Project (AutoCrypt Demo Project.xml)

Generate the Rupdate utity and customize Remote Update settings.

○ Generate RUpdate utility
Specify destination path for RUpdate utility
C:/testing/MyApps/RUpdate_AutoCrypt_Demo_Project.exe

● Generate Activation Code
Specify Remote Update Transaction Request (rutr) file path
C:/testing/MyApps/rutr_box0x001fd638_01_OCT_2020.rutr

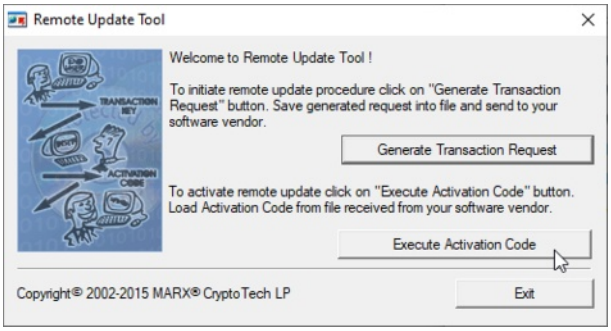
Specify Remote Update licenses (values)
□ Network License
☑ Expiration Date&Time
Set to Unlimited
□ Password

Specify destination path for Activation Code
C:/testing/MyApps/ruac_box0x001fd638_01_OCT_2020.ruac

After finishing all settings, click "Generate" to generate the Activation Code. Then send the Activation Code (.ruac) file to your end-user.

2.5.4. Executing the Activation Code (End User)

As soon as the-end user receives the Activation Code, he can perform an update of the CRYPTO-BOX. To do so, the end user launches the Remote Update Tool, attaches the CRYPTO-BOX and clicks the "Execute Activation Code" button. The unique transaction code (generated for each request) ensures that only the CRYPTO-BOX which was attached during transaction code generation can be updated.


Remote Update Tool

Welcome to Remote Update Tool !

To initiate remote update procedure click on "Generate Transaction Request" button. Save generated request into file and send to your software vendor.

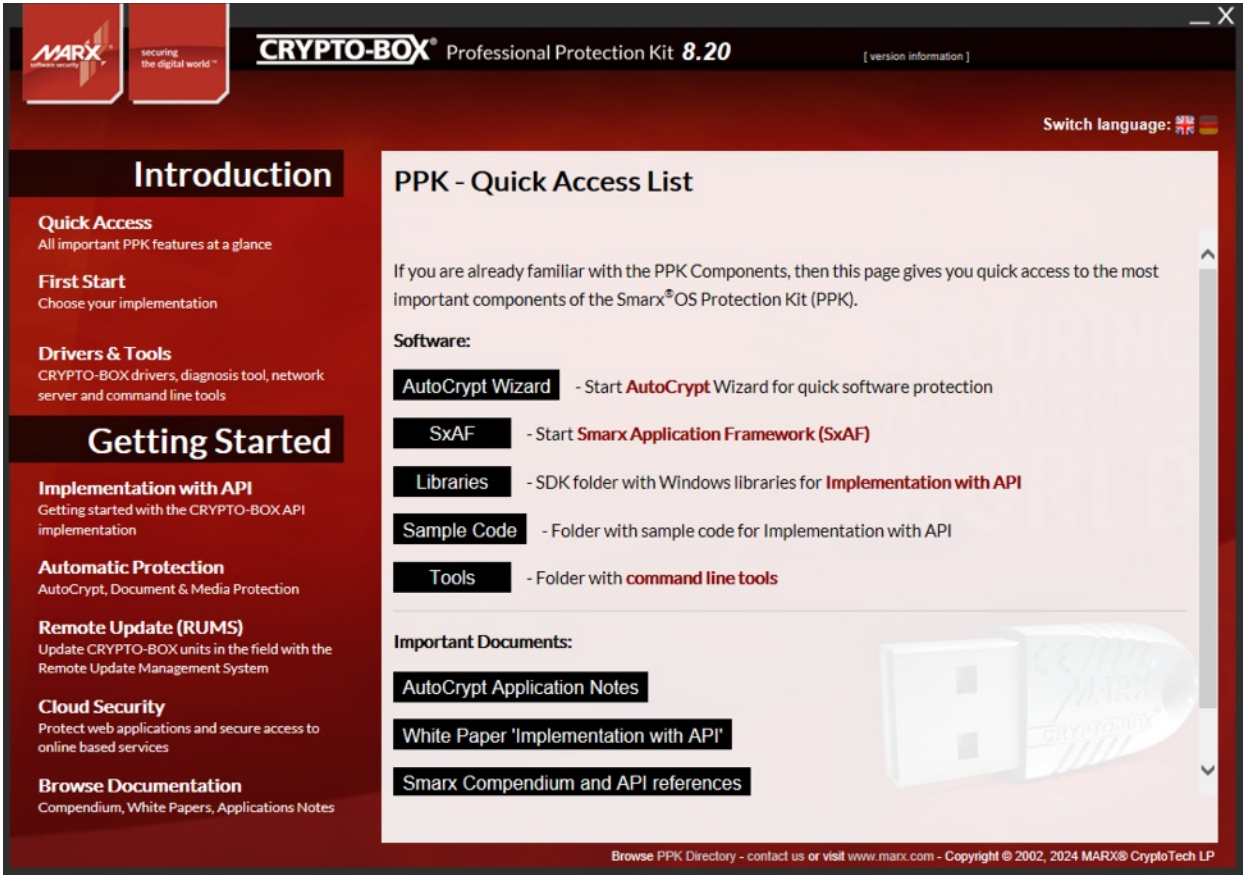
To activate remote update click on "Execute Activation Code" button. Load Activation Code from file received from your software vendor.

Copyright© 2002-2015 MARX® CryptoTech LP

3. AutoCrypt SxAF (Smarx® Application Framework)

3.1. Starting AutoCrypt SxAF

Click on the “MARX PPK Control Center” desktop shortcut. The Control Center provides an overview of the components installed, including a brief introduction and links to these components. From the menu select "Quick Access", then "SxAF".


MARX software security
securing the digital world™
CRYPTO-BOX® Professional Protection Kit 8.20
[version information]
Switch language:

Introduction

Quick Access
All important PPK features at a glance

First Start
Choose your implementation

Drivers & Tools
CRYPTO-BOX drivers, diagnosis tool, network server and command line tools

Getting Started

Implementation with API
Getting started with the CRYPTO-BOX API implementation

Automatic Protection
AutoCrypt, Document & Media Protection

Remote Update (RUMS)
Update CRYPTO-BOX units in the field with the Remote Update Management System

Cloud Security
Protect web applications and secure access to online based services

Browse Documentation
Compendium, White Papers, Applications Notes

PPK - Quick Access List

If you are already familiar with the PPK Components, then this page gives you quick access to the most important components of the Smarx®OS Protection Kit (PPK).

Software:

AutoCrypt Wizard - Start AutoCrypt Wizard for quick software protection

SxAF - Start Smarx Application Framework (SxAF)

Libraries - SDK folder with Windows libraries for Implementation with API

Sample Code - Folder with sample code for Implementation with API

Tools - Folder with command line tools

Important Documents:

AutoCrypt Application Notes

White Paper 'Implementation with API'

Smarx Compendium and API references

Browse PPK Directory - contact us or visit www.marx.com - Copyright © 2002, 2024 MARX® CryptoTech LP

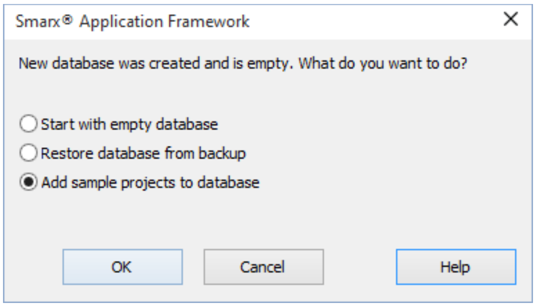
A detailed description of all components of the Smarx Application Framework can be found in the “Smarx Compendium”. Check marx.com to download the latest version of the Compendium as PDF file.

3.2. Smarx Application Framework (SxAF)

The Smarx Application Framework is a complete project oriented environment for software vendors and distributors to perform software and data protection and licensing scenarios. SxAF provides the following main tasks:

- Protect and license your software with a graphical user interface, including definition of protection and licensing options and dialog messages;
- Configure (format) CRYPTO-BOX according to chosen protection and licensing scenario;
- Remotely update licensing data contained in the CRYPTO-BOX distributed to the end-user (see separate RUMS Application Notes for further details);
- Manage end-user profiles.

When Smarx Application Framework (SxAF) is started for the very first time, a new database is created and you will see the following dialog:



New database was created and is empty. What do you want to do?

○ Start with empty database
○ Restore database from backup
● Add sample projects to database

If you choose “Leave database empty”, the database contains only default “cbu_demo” Hardware Profile and no projects. If you choose “Create demo projects”, two demo projects for AutoCrypt (Local and Network) are created which are intended for Evaluation.
After that, the Smarx Application Framework main screen will appear.

It is strongly recommended to make regular backups of the SxAF database, especially before updating to a new version of the Protection Kit. To backup the database, select the “Backup" option in the "Database" menu.

3.3. Protecting Applications

3.3.1. Steps for Protecting Applications with AutoCrypt

It is recommended you follow these steps when protecting your application with AutoCrypt:

1. Create a new Smarx Application Framework (SxAF) project type AutoCrypt. A project includes all information that is used for programming the CRYPTO-BOX. The projects are stored in the internal SxAF database.
2. Add the application(s) you want to protect to the project and choose your desired protection settings and licensing options. See chapter *3.3.7* for an overview about available Licensing Options (data object types).
3. Protect the application(s).
4. Use CB Format (see chapter *3.4*) to format your CRYPTO-BOX units with the project settings.
5. Additionally, you can export your project settings into an XML file for usage with command line based tools to automate application protection and CRYPTO-BOX formatting (see chapter *3.3.13*).
6. If you plan to update licensing options (e.g., expiration date or usage counter) in the CRYPTO-BOX at your end-user's site, you can create the Remote Update Utility for this project and ship it along with the CRYPTO-BOX to your end-users (see chapter *3.5* for more information).
7. Test the protection and your selected licensing options carefully.
8. Ship your protected application together with the CRYPTO-BOX and the necessary supplemental files (drivers, network server in case of network licensing). MARX provides an easy-to-use redistributable setup for this case. See chapter *5* for more details.

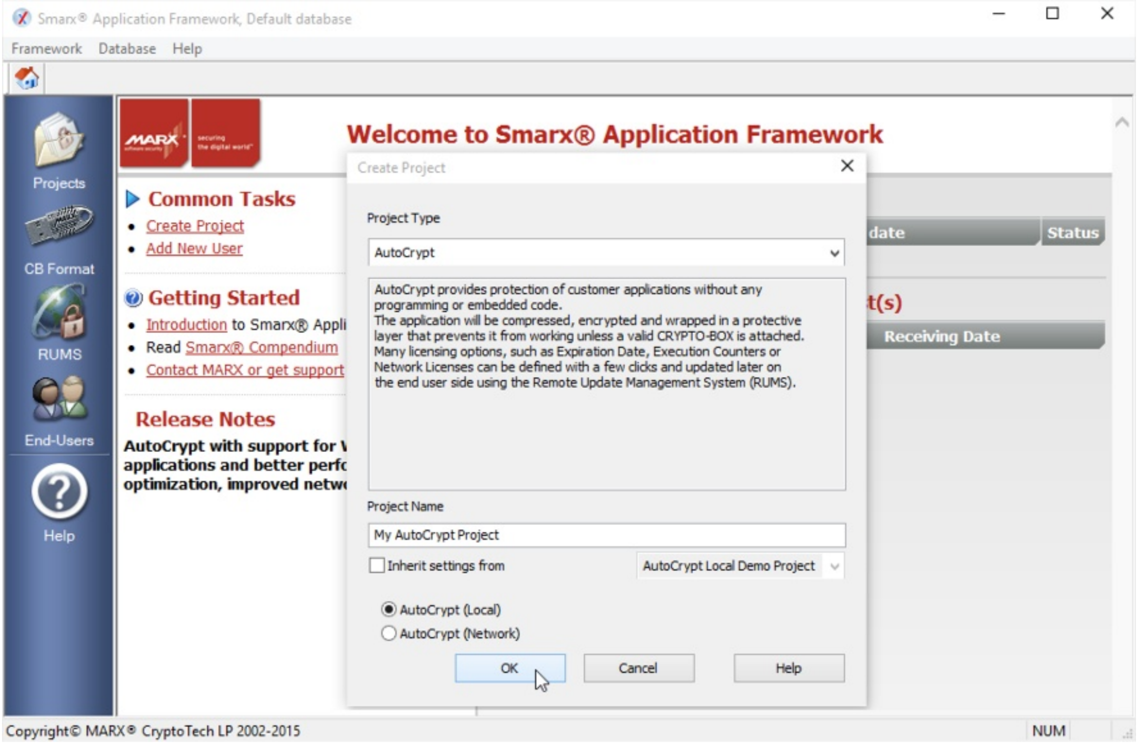
3.3.2. Creating new projects or selecting existing projects

On the main screen of Smarx Application Framework (SxAF), you can either create a new project or work with an existing project. If you are working with an existing project (or you want to evaluate the existing AutoCrypt demo projects), click on the “Projects” tab in the left navigation bar and choose one of the existing projects to change its settings.

Click the “Create Project” button to create a new project.

Now, enter a project name. AutoCrypt projects can be either local (applications are protected with CRYPTO-BOX on local PC) or network-based (applications are protected with CRYPTO-BOX located on the Server). Select “AutoCrypt (Local)”, if you want to use local protection and licensing only. Select “AutoCrypt (Network)” for network protection and licensing or combined Local/Network approach.

See chapter 5 for details on network server installation. Additional information on using the CRYPTO-BOX in networks can be found in the Smarx Protection Kit Compendium, chapter 5. Check marx.com to download the latest version of the Compendium as PDF file.



AutoCrypt with support for \
applications and better perfo
optimization, improved netw

Project Type: AutoCrypt

AutoCrypt provides protection of customer applications without any programming or embedded code.
The application will be compressed, encrypted and wrapped in a protective layer that prevents it from working unless a valid CRYPTO-BOX is attached.
Many licensing options, such as Expiration Date, Execution Counters or Network Licenses can be defined with a few clicks and updated later on the end user side using the Remote Update Management System (RUMS).

Project Name: My AutoCrypt Project

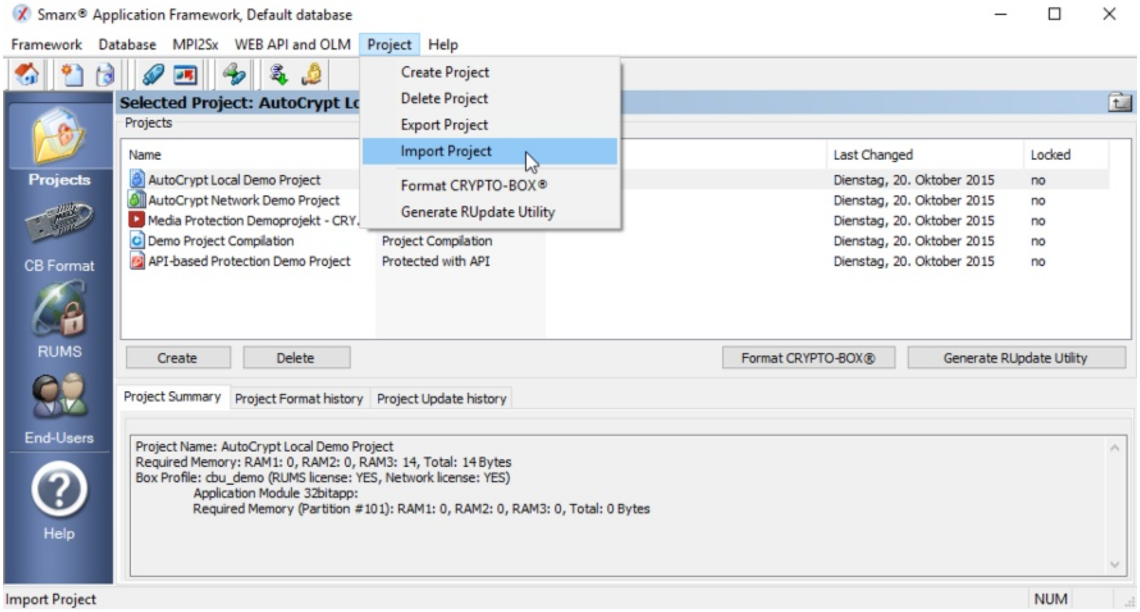
☐ Inherit settings from: AutoCrypt Local Demo Project

● AutoCrypt (Local)
○ AutoCrypt (Network)

The “Inherit settings from” option allows you to create an independent copy of an existing project. All applications, data objects, project and application settings will be copied. Note that the inherited project will be the same type as the source project (local or network)

3.3.3. Importing Settings from AutoCrypt Wizard

If you worked with the AutoCrypt Wizard before (see chapter 2), you can import your settings to AutoCrypt SxAF to take advantage of SxAF features, such as extended project and end-user management. To import the project, click on the “Projects” tab in the left navigation bar and choose “Import Project” from the “Project” menu.



Name		Last Changed	Locked
AutoCrypt Local Demo Project		Dienstag, 20. Oktober 2015	no
AutoCrypt Network Demo Project		Dienstag, 20. Oktober 2015	no
Media Protection Demoprojekt - CRY.		Dienstag, 20. Oktober 2015	no
Demo Project Compilation	Project Compilation	Dienstag, 20. Oktober 2015	no
API-based Protection Demo Project	Protected with API	Dienstag, 20. Oktober 2015	no

```
Project Name: AutoCrypt Local Demo Project
Required Memory: RAM1: 0, RAM2: 0, RAM3: 14, Total: 14 Bytes
Box Profile: cbu_demo (RUMS license: YES, Network license: YES)
        Application Module 32bitapp:
        Required Memory (Partition #101): RAM1: 0, RAM2: 0, RAM3: 0, Total: 0 Bytes
```

After the import was successful, the project will appear in the list.

3.3.4. General Project Settings

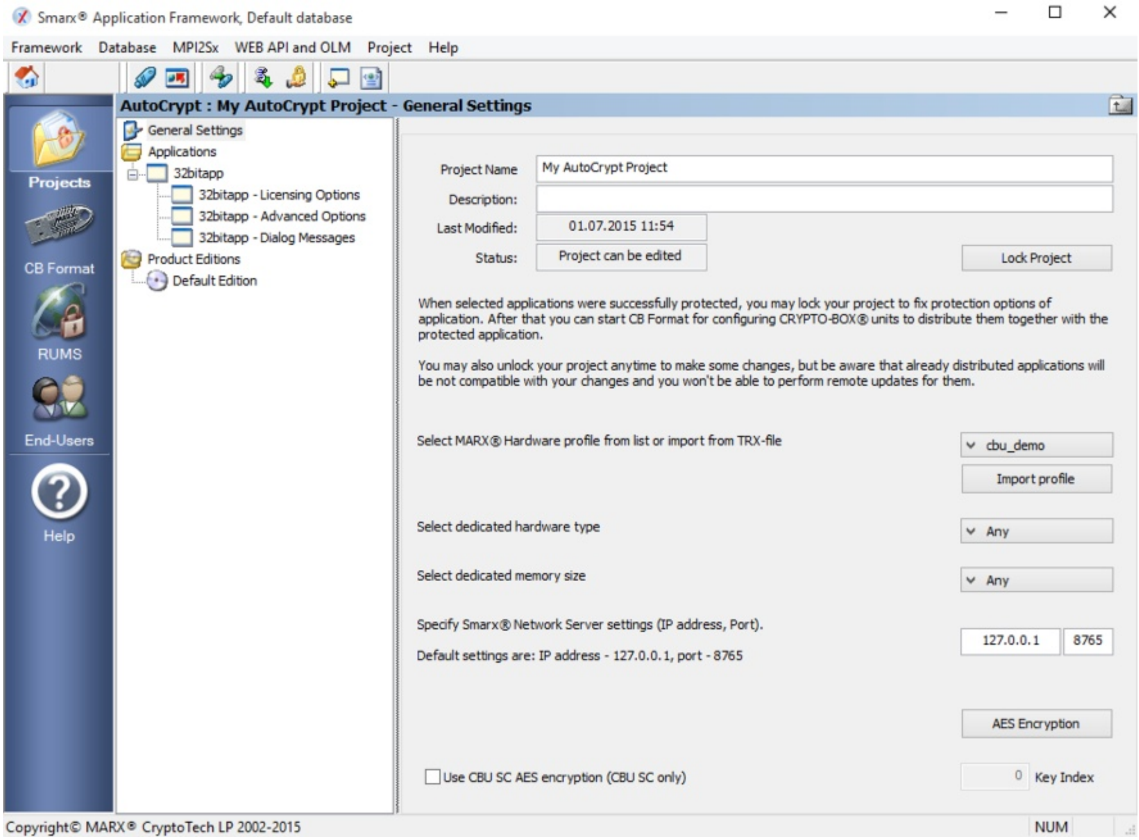
The “General Settings” tab in the navigation tree on the left side allows you to edit settings for the selected project.

You can change and edit the project name and description, as well as lock the project (use it in read-only mode). This prevents changes to the project. You can always unlock the project for editing, but keep in mind that your previously formatted CRYPTO-BOX units (which may have been already delivered to end-users) might not work properly with the application(s) protected with the modified project.

In the lower part of the Window, you need to select the CRYPTO-BOX hardware profile for the project. This hardware profile contains the access codes to be used by the protected application to access the CRYPTO-BOX. Every MARX customer will get his customer specific TRX file. Select the "cbu_demo" profile for the CRYPTO-BOX that was shipped with the Evaluation Kit, or click on "Import profile" to import the profile from the CDROM you received along with your customer specific CRYPTO-BOX units from MARX.

See the “TRX File” White Paper at marx.com for details on hardware profile handling.

You may also select the exact CRYPTO-BOX type and/or its memory size, but this is optional: if you leave default settings, SxAF will detect the attached CRYPTO-BOX automatically.



AutoCrypt : My AutoCrypt Project - General Settings

Project Name: My AutoCrypt Project
Description:
Last Modified: 01.07.2015 11:54
Status: Project can be edited

When selected applications were successfully protected, you may lock your project to fix protection options of application. After that you can start CB Format for configuring CRYPTO-BOX® units to distribute them together with the protected application.

You may also unlock your project anytime to make some changes, but be aware that already distributed applications will be not compatible with your changes and you won't be able to perform remote updates for them.

Select MARX® Hardware profile from list or import from TRX-file: cbu_demo

Select dedicated hardware type: Any

Select dedicated memory size: Any

Specify Smarx® Network Server settings (IP address, Port). 127.0.0.1 8765
Default settings are: IP address - 127.0.0.1, port - 8765

□ Use CBU SC AES encryption (CBU SC only) 0 Key Index

Copyright© MARX® CryptoTech LP 2002-2015

When creating or editing a network project, you can specify Network Server settings (IP address, port); by default, the IP address is “127.0.0.1” (this is local IP address of your computer), the port is 8765. You can also submit a computer name (e.g. “PC-517”) instead of an IP address. If the server can not be found at the specified IP address, the protected application will open a dialog asking for server settings.

The button “AES Encryption” allows you to define values for the AES/Rijndael Private Key and the Initialization Vector used to encrypt the application.

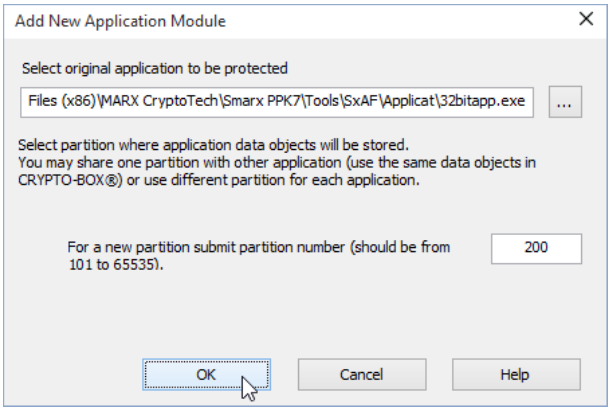
All protected applications will be compressed and encrypted by default. For application encryption, the AES Private Key of the CRYPTO-BOX is used. All protected applications in one project will share the same AES key value for their encryption.
With the CRYPTO-BOX SC it is possible to have separate AES keys for every application/ set of applications. Check the "Use CBU SC AES encryption" field to do so. For more information on CRYPTO-BOX SC features, please visit marx.com.

Important Note:
Every AutoCrypt project may have its individual AES key. However, this means: if a CRYPTO-BOX was formatted for Project #1, it cannot be used with an application which was protected with project #2 if the projects have different AES key values (which is default setting). So if you need different (AutoCrypt) projects on one CRYPTO-BOX, you have to set the AES encryption key to the same values for all these projects.
Alternatively, multiple applications can be added to the same project - then they will share the same key value (see chapter *3.3.5* for details*)*.
An exception is the CRYPTO-BOX SC which allows to define an individual AES key for every project.

3.3.5. Adding Applications to the Project

With the "Applications" tab you can add, edit, and remove applications to/from the project. To add a new application to the project, click the “Add Application” button. Select the original application file and path/name for the protected application. Browse for the original application that is to be protected (the path must exist) – it may be a Windows EXE or a DLL file. For testing purposes, you can use the included test applications: Browse to “[SmarxOS PPK root]\Tools\SxAF\Applicat” and select desired Win32/Win64 sample application or Win32Dll.dll/Win64Dll.dll for sample DLL, which can be tested with DllLauncher.exe/ DllLauncher64.exe program.

Next, specify the number of the CRYPTO-BOX partition in which the application protection settings (data objects) will be stored. The number of the partition must be in range between 101 and 65535. The “Share existing partition” check box is only available if you already have other applications in this project. It allows you to share one partition (set of licensing data) between two or more applications (use the same data objects in the CRYPTO-BOX memory). Leave this option unchecked if you want to have individual licensing and protection settings for your applications.


Add New Application Module

Select original application to be protected

Files (x86)\MARX CryptoTech\Smarx PPK7\Tools\SxAF\Applicat\32bitapp.exe

Select partition where application data objects will be stored.
You may share one partition with other application (use the same data objects in CRYPTO-BOX®) or use different partition for each application.

For a new partition submit partition number (should be from 101 to 65535). 200

Press the “OK” button to continue with the protection settings.

One project can contain several applications (.exe/.dll files), protected with one CRYPTO-BOX. All licensing options for each application are stored in a dedicated partition which will be created for every application. The maximum number of partitions in a CRYPTO-BOX is 32. If this is not enough, you may share one partition with multiple applications.

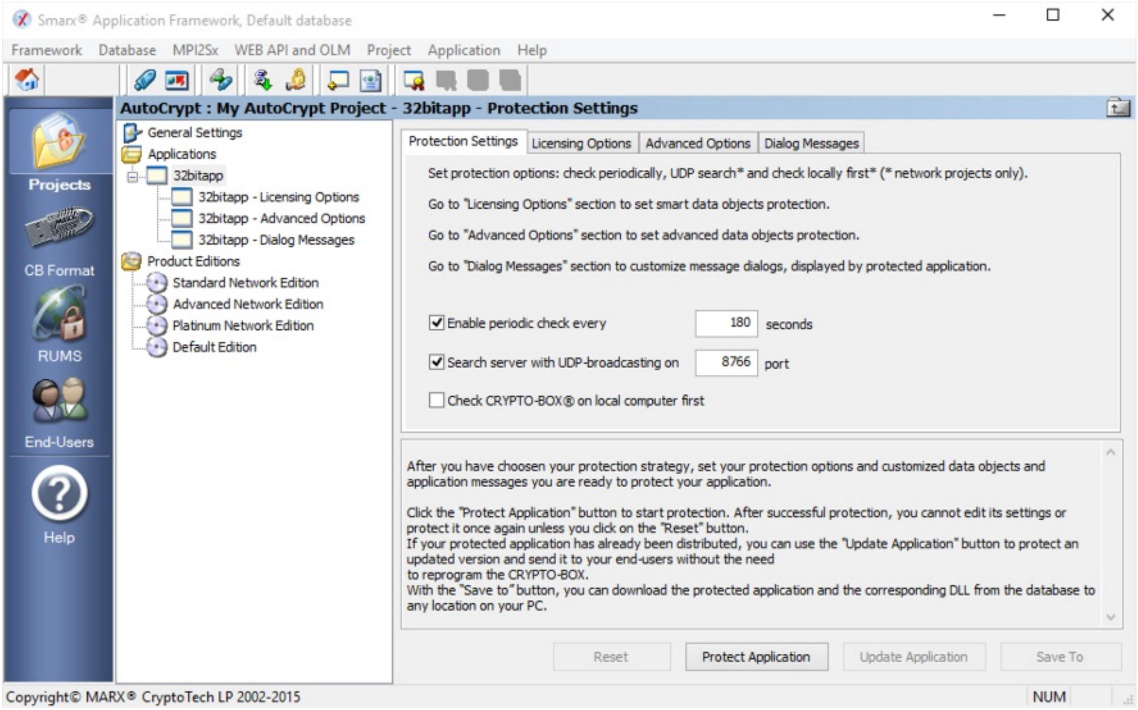
Important notes on protecting multiple applications and “sharing partitions”:

- Shared partitions can not contain Application Checksum and Application Name Hash data objects, because their values are unique to application. Thus, it is not possible to share a partition which already contains such data objects.
- If you change data objects and/or the sequence in which they are stored in memory (geometry) for an application in a shared partition, it will be automatically changed for all applications sharing that partition. Be careful not to unintentionally change settings for an application you have already distributed.
- Protection settings such as Periodic Check, Display License Status, Display License Agreement and all application dialogs (error messages, copyrights, button texts and links) are application specific and they are managed separately for each application.
- If you want to protect multiple components of one application that are running at the same time, we strongly recommend to use different partitions and not to share them!

3.3.6. Application protection settings

After selecting the application to protect you will automatically be forwarded to the protection settings. You can also select the desired application in the left navigation tree.

On the “Protection Settings” tab, you can customize protection options for your application.


Smarx® Application Framework, Default database
Framework Database MPI2Sx WEB API and OLM Project Application Help
AutoCrypt : My AutoCrypt Project - 32bitapp - Protection Settings
Projects
CB Format
RUMS
End-Users
Help
General Settings
Applications
32bitapp
32bitapp - Licensing Options
32bitapp - Advanced Options
32bitapp - Dialog Messages
Product Editions
Standard Network Edition
Advanced Network Edition
Platinum Network Edition
Default Edition
Protection Settings
Licensing Options
Advanced Options
Dialog Messages
Reset
Protect Application
Update Application
Save To
Copyright© MARX® CryptoTech LP 2002-2015
NUM

Set protection options: check periodically, UDP search* and check locally first* (* network projects only).

Go to "Licensing Options" section to set smart data objects protection.

Go to "Advanced Options" section to set advanced data objects protection.

Go to "Dialog Messages" section to customize message dialogs, displayed by protected application.

☑ Enable periodic check every 180 seconds

☑ Search server with UDP-broadcasting on 8766 port

☐ Check CRYPTO-BOX® on local computer first

After you have choosen your protection strategy, set your protection options and customized data objects and application messages you are ready to protect your application.

Click the "Protect Application" button to start protection. After successful protection, you cannot edit its settings or protect it once again unless you click on the "Reset" button.
If your protected application has already been distributed, you can use the "Update Application" button to protect an updated version and send it to your end-users without the need
to reprogram the CRYPTO-BOX.
With the "Save to" button, you can download the protected application and the corresponding DLL from the database to any location on your PC.

If the periodic check option is enabled, CRYPTO-BOX presence is checked for within the defined time-out while the application is running. If the CRYPTO-BOX is not found, the “Protection error” dialog message (see chapter *3.3.9 Defining Dialog Boxes*) will be shown. If CRYPTO-BOX is still missing the application will be terminated. The value should not be less than 60 seconds (more is recommended) for one application and increased when several applications are protected with one CRYPTO-BOX, or when network mode is used.

The last two options are available in network mode only: search server with UDP broadcasting means that the protected application will automatically search for available CRYPTO-BOX Servers (the Server must be on the same sub-net). Default port is 8766, UDP port of the server can be changed in server configuration if required (see Smarx Protection Kit Compendium, chapter 5 for more information on network configuration). If the last option (Check CRYPTO-BOX on local PC first) is checked, the protected application will look for locally attached CRYPTO-BOX first before it starts a network search.

3.3.7. Licensing Options (Data Objects)

On the “Licensing Options” tab, you can define required licensing logic for your application by choosing licensing data objects of predefined types, such as Expiration Date, Run Counter, etc.

- To add a licensing option (data object) check corresponding check box or radio button.
- To edit the value of a data object press related “Change” button.
- To remove the data object, click the check box or radio button near data object and confirm the message that you want to delete it with "Yes".
- In network mode you will have an additional data object type "Network License" which allows you to set up the number of network licenses for the application. It defines how many instances of your application can run in the network at the same time. See White Paper “Network Licensing” more information on Network License Management. License sharing rule (an option of “Network License”) defines a condition when one license is shared by multiple instances of protected application.

You can define independent network license counters for each of your protected applications. License counters are supported by the CRYPTO-BOX SC and CRYPTO-BOX XS models. The CRYPTO-BOX Versa also offers network support, but number of network licenses are always unlimited.

For the most DataObjects you have the choice between “CDO” (Crypted DataObject) and standard DataObjects (“CDO” deactivated). CDO offers additional protection against manipulations, so we recommend to keep this option activated. If you plan to combine AutoCrypt and API implementation and additionally want to query these Crypted DataObjects via API commands, please check the sample code in our PPK and corresponding readme files for CDO compatibility. Not all API libraries offer CDO support, eg. for older or exotic compilers. Please contact our Technical Support for any questions.

The following types of licensing data objects are supported:

TEOSDO_EXPIRATION_DATE_AND_TIME “Expiration Date & Time”	exact date and time when the protected application is going to expire, for example "31 DEC 2016 23:59:59";
TEOSDO_EXPIRATION_DATE “Expiration Date”	fixed expiration date, submitted in the format “31 DEC 2016” - this data object type is obsolete and only preserved for compatibility purposes, please use “Expiration Date & Time” instead;
TEOSDO_NUMBER_OF_DAYS “Expiration Days”,	“Expiration Days” is a flexible expiration date, submitted as number of days the application is allowed to be used, starting **from the day the CRYPTO-BOX was formatted**.
“Relative Expiration Date”	The “Relative Expiration Date” specifies the number of days the application is allowed to be used **from the first run**. i.e. counter is activated on the first application launch;
TEOSDO_TIME_ALLOWED “Expiration Time”	real-time expiration, submitted as period of allowed application usage (in seconds);
TEOSDO_COUNTER “Run Counter”	number of application executions (runs);
TEOSDO_PSW_HASH “Password”	hash value, calculated from password string;
TEOSDO_APPNAME_HASH “Application Name Hash”	hash value, calculated from application name string;
TEOSDO_APP_CS “Application Checksum”	checksum, calculated from application file.
TEOSDO_NET_LICENSE “Network License”	network license counter value of the application (for AutoCrypt Network projects only). See White Paper “Network Licensing” more information on Network License Management.

If you need to have different licensing option (DataObject) settings for different customers you don't need to create a separate project for each customer. Instead you can create Product Editions of your project with different data object settings. Please refer to chapter *3.3.11 Product Editions* for more information.

3.3.8. Advanced Protection options

Password
This data object defines an application password, which will be required every time the application is launched. This feature can be useful as a "nag screen" when making demo versions or for additional security.

Application Checksum
This data object defines a checksum, calculated for the protected application file. Its value will be stored in the CRYPTO-BOX and used to check if the application was manipulated or corrupted, as well as prevent unauthorized modification. The checksum cannot be set by value, it is calculated only after successful application protection.

Application Name Hash
This data object is specific to AutoCrypt projects: it contains a hash value calculated for the name of protected application. This hash value will be stored in the CRYPTO-BOX and used to check if the application was renamed on the user side. The application name hash can't be set by value, it is calculated after successful application protection.

⚠ Application Checksum and Application Name Hash cannot be used if the application partition is shared between two or more applications (see also *3.3.5 Adding Applications to the Project*). Furthermore, it is not recommended to use these options if you plan to update your application regularly, because the new application will not be compatible with existing Checksum/Hash settings stored in the CRYPTO-BOX.

Signature (GUID)
This option is mainly for software developers: It allows you to store an individual data block (up to 16 bytes length) in the CRYPTO-BOX memory (e.g. customer specific data) and provide it directly to your protected application – with minimal efforts! The advantage of this approach: there is no knowledge about the CRYPTO-BOX API necessary – just copy the code snippet from our sample application source code to your application source code.

After license validation was successful, AutoCrypt will read the data from the CRYPTO-BOX, decrypts it and writes it to a buffer signed with a unique signature which can be identified and read by your application.

A readme file with detailed instructions and sample code can be found in the Protection Kit:
[Smarx PPK root folder]\SmarxOS\API\Win\Samples\ReadMemoryBySignature

3.3.9. Defining Dialog Boxes

During protected application execution, some message dialogs can be displayed, for example: license status, protection errors/warnings, etc. To edit these messages, click the "Dialog Messages" tab, select the proper message from the list and click on "Edit Dialog". Next, you can set the message title (caption) and the body text. If you do not wish to have messages displayed, leave the title and body empty.

If the "Display License Agreement" check box is enabled, a License Agreement will be displayed before the application starts. You may add a customer specific text here.

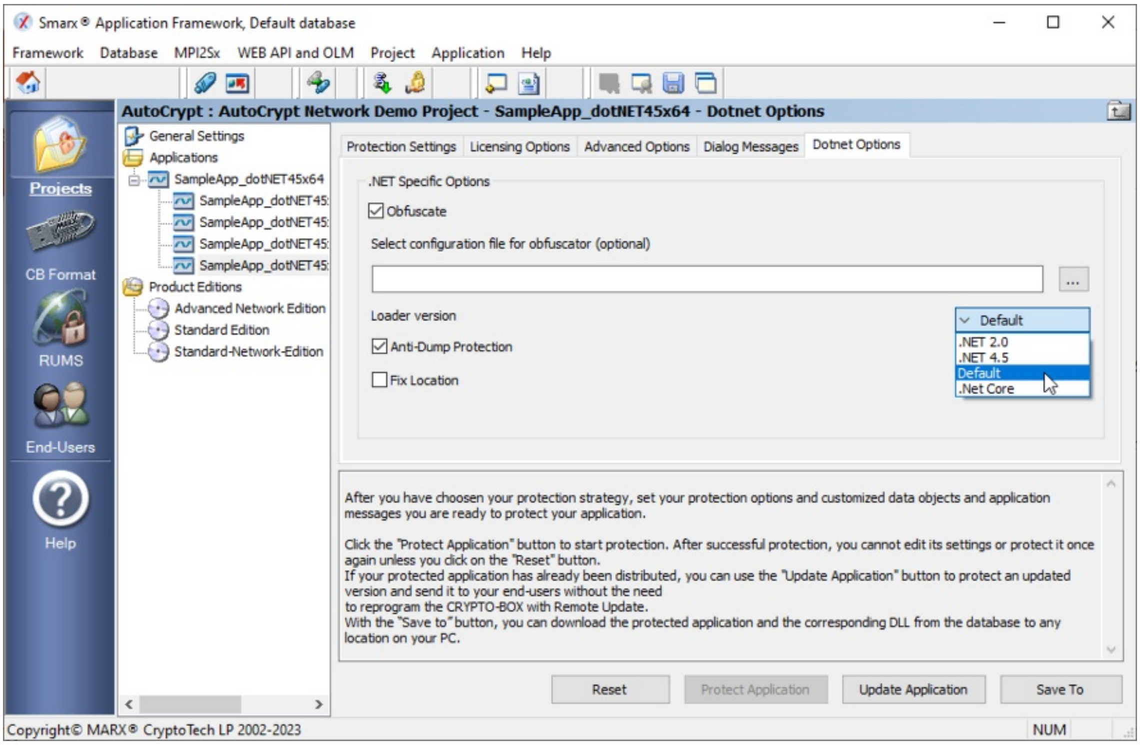
If the "Display License Status" check box is enabled, the protected application will display the license status before it starts and indicate how often or how many more days the application may be executed. If no data objects with licensing options were defined (see chapter *3.3.7*), the "Display License Status" check-box will be unavailable.

3.3.10. .Net Options

If your source application is a .NET based application, you will have an additional tab "Dotnet Options" with further settings for your application:

- The **"Obfuscate"** option offers automatic .NET application obfuscation by using the open source software "Obfuscar" in background. Refer to the Obfuscar webpage for further details.
- The **"Anti-Dump Protection"** option hinders dumping of .NET applications. We recommend to leave it activated. It should be activated only in case the protected application does not start.
- The **"Fix Assembly Location"** option resolves the issue that .NET applications might not work properly after protection with AutoCrypt if they use Location property to get path of executing assembly, e.g. Assembly.GetExecutingAssembly().Location will return empty string. This happens because the protected assembly does not exist in the file system, since our AutoCrypt loader loads it from the memory.
- The **"Loader version"** option determines which type of loader is used to protect a .NET application. This can be useful when the protected .NET application does not start. The following options are available:

  - DEFAULT – AutoCrypt will automatically detect the type of .NET application using .NET_CORE for newer .Net apps (if there is an .exe, .dll and .runtimeconfig.json with the same name in the source file folder), and .NET_20 for all other applications.
  - .NET_20 – Force using older loader for .NET 2.x and 3.x applications (some .NET 4.5+ applications might not work with these settings)
  - .NET_45 – Force using loader for .NET 4.x applications
  - .NET_Core – Force using loader for .NET Core applications (.NET 6.0 and higher)
- The **"Console Application"** option is only available when .NET_Core or .NET_SPLIT_LOAD loader is selected and has to be used when protecting a console application service.


Smarx® Application Framework, Default database

AutoCrypt : AutoCrypt Network Demo Project - SampleApp_dotNET45x64 - Dotnet Options

.NET Specific Options

☑ Obfuscate

Select configuration file for obfuscator (optional)

Loader version: Default

☑ Anti-Dump Protection

□ Fix Location

After you have choosen your protection strategy, set your protection options and customized data objects and application messages you are ready to protect your application.

Click the "Protect Application" button to start protection. After successful protection, you cannot edit its settings or protect it once again unless you click on the "Reset" button.
If your protected application has already been distributed, you can use the "Update Application" button to protect an updated version and send it to your end-users without the need
to reprogram the CRYPTO-BOX with Remote Update.
With the "Save to" button, you can download the protected application and the corresponding DLL from the database to any location on your PC.

Copyright© MARX® CryptoTech LP 2002-2023

Important notes for .Net 6.0+ (.Net Core) applications:

1. For .Net 6.0+ applications, please always specify the corresponding .dll file as the original application (see 2.2.8), not the .exe! The .exe in .Net 6.0+ is just a loader that loads the actual application which is in the .dll file. AutoCrypt protects the .dll and replaces the .exe with its own loader. If the .exe is specified, AutoCrypt will attempt to protect the associated .dll if a .NET 6.0+ application was detected.

2. For .Net 6.0+ applications, the target file must have the same name as the original file (see 2.2.8), otherwise the protected application will not start!

3. In case of .Net 6.0+ applications you can specify the same folder for both the original application and the protected application. In that case AutoCrypt replaces the original application with the protected files and moves the original files to the _backup folder. If you choose a different destination path, always remember to copy the associated runtime configuration files (.json files) to the destination directory, otherwise the protected application will not start!

4. .Net 6.0+ applications can be protected only with STANDARD or DOTNET_Core option, with all other settings the protected application will not start!

5. AutoCrypt cannot protect .Net 5.0 applications out of the box! Either upgrade to a newer .Net version (6.0 or higher), or use the workaround described in the AC_Tool Readme file to adjust the .Net version in your runtime configuration file. See chapter 4.1 for more information.

3.3.11. Product Editions

Every SxAF project (AutoCrypt, Implementation with API or Document/Media Protection) is associated with some license: a set of data objects with initial values defining licensing for software product or document(s) protected with this SxAF project.

When protecting software or media products, you may want to define more than one license per project, or in other words to have more than one edition of your product.

Some examples for product editions:

- Advanced Local Edition: for 1 year
- Unlimited Local Edition: unlimited

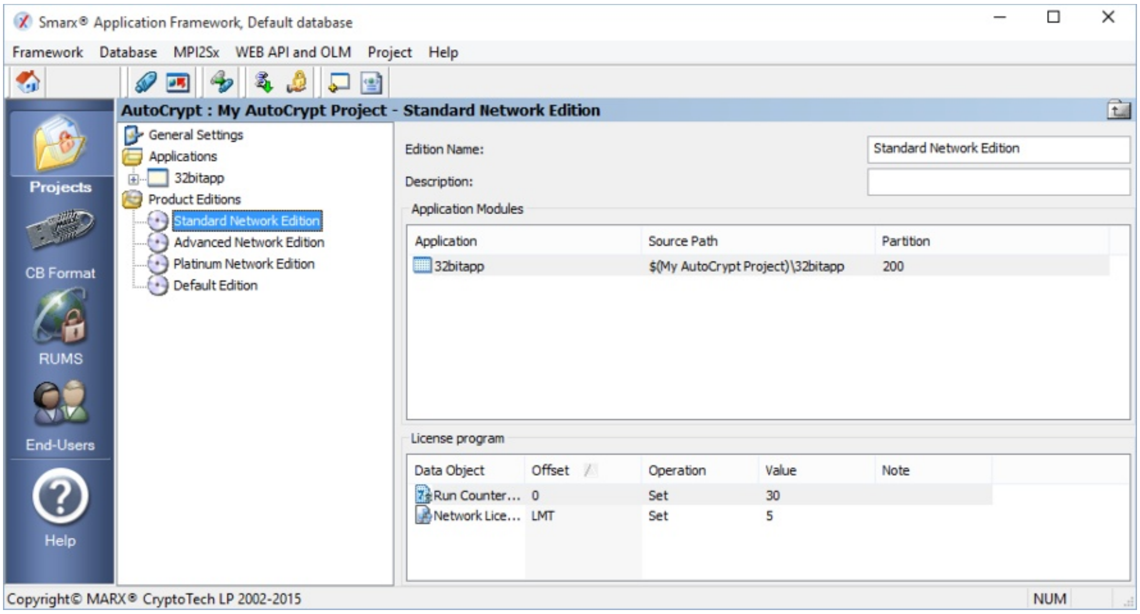
or

- Standard Network Edition: 5 network licenses for 6 months
- Advanced Network Edition: 10 network licenses for 1 year
- Platinum Network Edition: 15 network licenses, unlimited time

And so on.

Such editions could help with your marketing and pricing strategies. All you have to do is to set up the desired Product Editions for your project and format your CRYPTO-BOX modules with those corresponding settings using CRYPTO-BOX Format (see chapter *3.4 CRYPTO-BOX® Format: Configuring and Programming*). By default only standard (default) edition is present.

To add a new Product Edition, click the "Product Editions" tab on the left navigation tree. Choose "Add Edition" to create a new one or double-click on an existing partition to change its settings.



AutoCrypt : My AutoCrypt Project - Standard Network Edition

Edition Name: Standard Network Edition

Description:

Application Modules

Application	Source Path	Partition
32bitapp	$(My AutoCrypt Project)\32bitapp	200

License program

Data Object	Offset	Operation	Value	Note
Run Counter…	0	Set	30	
Network Lice…	LMT	Set	5	

You can only change data objects in the Product Edition settings which were initially created in your project settings. Therefore, please be sure you have created the corresponding data object in your project (see chapter *3.3.7*) before you create Editions with different settings.

3.3.12. Protecting the Application

To protect the application, click the “Protect Application” button.

Now the original application file will be downloaded from the SxAF database and protected. The protected application and its DLL will be uploaded to the database. If you have multiple applications to protect inside your project, you have to repeat this step for every single application.

ℹ AutoCrypt will automatically compress and encrypt the protected application (AES Rijndael Private Key of the CRYPTO-BOX is used for that, see chapter *3.3.4 General Project Settings*). Additionally, it will be protected against debugging.

If the application has already been protected, its settings cannot be edited or protected once again unless clicking on the "Reset" button. The “Save to” button, allows to download the protected application and the corresponding DLL from the database to any location on the computer.

All information concerning the protected application is stored in the SxAF database. Next, you will use "CB Format" to program the CRYPTO-BOX with your project settings (see chapter *3.4 CRYPTO-BOX® Format: Configuring and Programming*).

Later, various protection options that were set by AutoCrypt can be remotely updated using the Remote Update Management System (RUMS). See Application Notes for RUMS at marx.com

ℹ You can automatize the protection of your applications with AutoCrypt by using AC_Tool.exe. This utility is a command line version of AutoCrypt which can be controlled within other applications using command line switches. See chapter 4.1 *AutoCrypt - Command Line Version* for more details.

⚠ If your protected application has already been distributed, you can protect an updated version and send it to your end-users without the need to reprogram the CRYPTO-BOX. To protect your updated application, open the AutoCrypt project and select “Update” in the “Application” menu or click on the corresponding toolbar icon.

3.3.13. Generating XML Script for Usage with Command Line Tools

If you want to integrate application protection and CRYPTO-BOX formatting with your own administration/ distribution strategy, the command line tools AC_Tool (for protecting applications, see chapter *4.1*) and SmrxProg (for configuring CRYPTO-BOX units, see chapter *4.2*) can be called up by applications or scripts.

The option “Generate script for AC_Tool and SmrxProg“ in the “Project“ menu allows you to export project data to an XML script file for further usage with AC_Tool.exe and SmrxProg.exe. Select the application you want to export from the database by enabling the appropriate checkbox and select a folder where the application will be stored. Then click “Export” and choose a folder for the XML file.

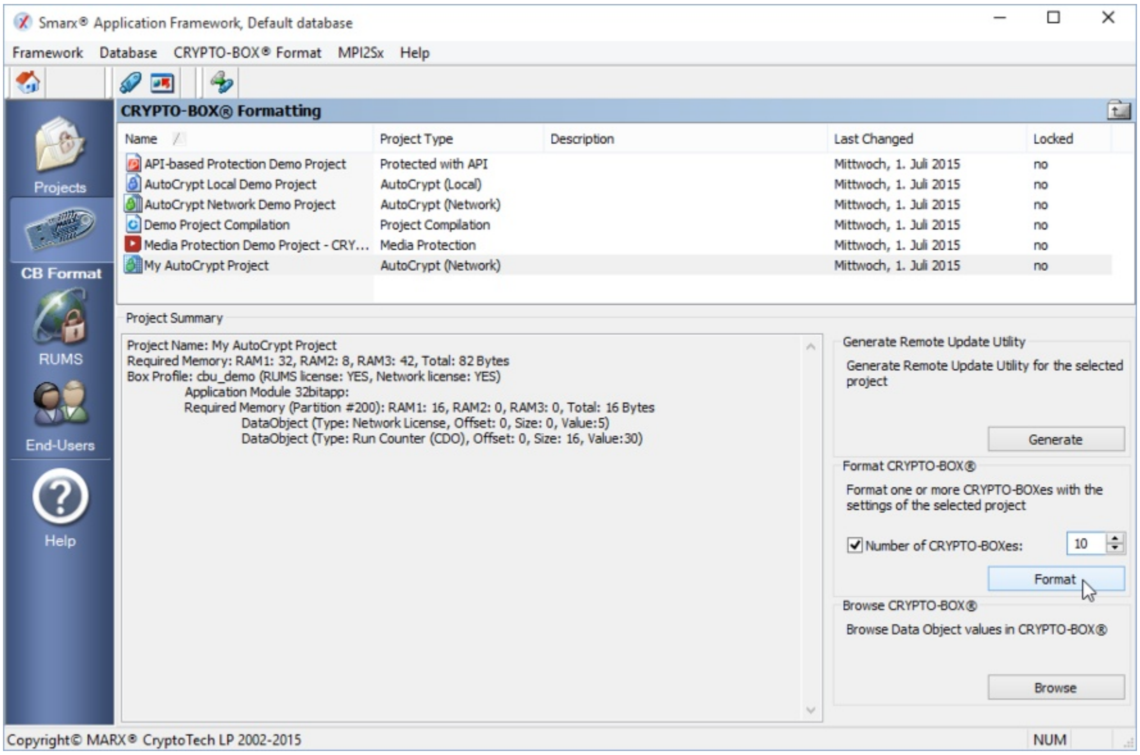
3.4. CRYPTO-BOX® Format: Configuring and Programming

CRYPTO-BOX Format as part of Smarx Application Framework provides CRYPTO-BOX formatting for AutoCrypt projects stored in the SxAF database (LM/db).

You can start CRYPTO-BOX Format using the “CB Format” button in the Smarx Application Framework.

3.4.1. Selecting projects to format

Select an existing project in the upper window. The Project summary field below allows you to review the project settings.



CRYPTO-BOX® Formatting

Name	Project Type	Description	Last Changed	Locked
API-based Protection Demo Project	Protected with API		Mittwoch, 1. Juli 2015	no
AutoCrypt Local Demo Project	AutoCrypt (Local)		Mittwoch, 1. Juli 2015	no
AutoCrypt Network Demo Project	AutoCrypt (Network)		Mittwoch, 1. Juli 2015	no
Demo Project Compilation	Project Compilation		Mittwoch, 1. Juli 2015	no
Media Protection Demo Project - CRY…	Media Protection		Mittwoch, 1. Juli 2015	no
My AutoCrypt Project	AutoCrypt (Network)		Mittwoch, 1. Juli 2015	no

Project Summary

```
Project Name: My AutoCrypt Project
Required Memory: RAM1: 32, RAM2: 8, RAM3: 42, Total: 82 Bytes
Box Profile: cbu_demo (RUMS license: YES, Network license: YES)
        Application Module 32bitapp:
        Required Memory (Partition #200): RAM1: 16, RAM2: 0, RAM3: 0, Total: 16 Bytes
                DataObject (Type: Network License, Offset: 0, Size: 0, Value:5)
                DataObject (Type: Run Counter (CDO), Offset: 0, Size: 16, Value:30)
```

Generate Remote Update Utility

Generate Remote Update Utility for the selected project

Format CRYPTO-BOX®

Format one or more CRYPTO-BOXes with the settings of the selected project

☑ Number of CRYPTO-BOXes: 10

Browse CRYPTO-BOX®

Browse Data Object values in CRYPTO-BOX®

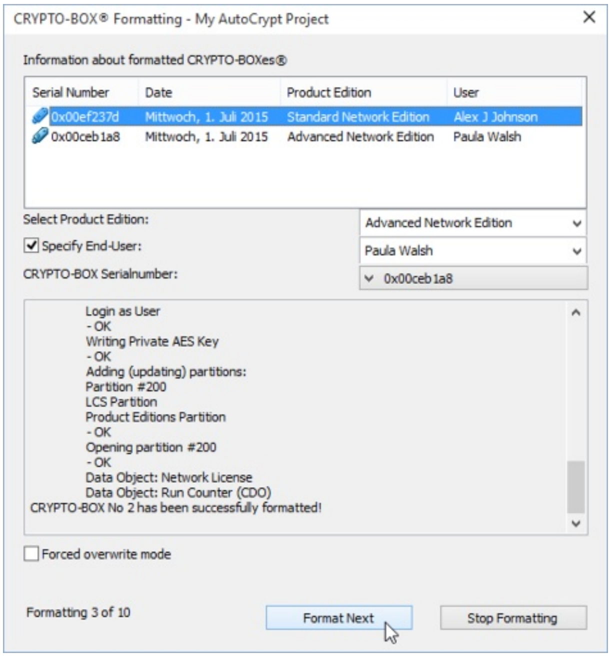
3.4.2. Formatting CRYPTO-BOX units

In the lower right corner, select the number of CRYPTO-BOXes you want to format. Plug in the first CRYPTO-BOX and click the “Format” button. In the next window, you will see the following options:

- **Information about formatted CRYPTO-BOXes** - shows you the CRYPTO-BOX units you already formatted, including Serialnumber, selected Product Edition and assigned End-User.
- **Select Product Edition** - choose the desired Product Edition (see chapter *3.3.11* for more information on Product Editions).
- **Specify End-User** - check this box if you want to assign the formatted CRYPTO-BOX to a certain End-User (see chapter *3.6* for more information on End-User Management). This allows you to identify the end-user later during Remote Update. If you have not specified any End-Users earlier, this field will be grayed out.
- **CRYPTO-BOX Serialnumber** - displays the Serial number of the currently attached CRYPTO-BOX.
- **Forced overwrite mode** - will delete all existing partitions from the CRYPTO-BOX, except the project specific ones.

Click the “Format Next” button to start formatting the first CRYPTO-BOX. A message box shows the status of the formatting process. After all CRYPTO-BOX modules have been formatted, click “Stop Formatting” to close the window.

If you get an error message during formatting that Administrator login to the CRYPTO-BOX has failed, check if you specified the correct hardware profile in your project. See chapter *3.3.4* for more information.



CRYPTO-BOX® Formatting - My AutoCrypt Project

Information about formatted CRYPTO-BOXes®

Serial Number	Date	Product Edition	User
0x00ef237d	Mittwoch, 1. Juli 2015	Standard Network Edition	Alex J Johnson
0x00ceb1a8	Mittwoch, 1. Juli 2015	Advanced Network Edition	Paula Walsh

Select Product Edition: Advanced Network Edition

☑ Specify End-User: Paula Walsh

CRYPTO-BOX Serialnumber: 0x00ceb1a8

```
          Login as User
          - OK
          Writing Private AES Key
          - OK
          Adding (updating) partitions:
          Partition #200
          LCS Partition
          Product Editions Partition
          - OK
          Opening partition #200
          - OK
          Data Object: Network License
          Data Object: Run Counter (CDO)
CRYPTO-BOX No 2 has been successfully formatted!
```

□ Forced overwrite mode

Formatting 3 of 10

The command line tool SmrxProg allows you to automate CRYPTO-BOX formatting. See chapter *4.2* for details.

Do not change the project settings after you have already formatted CRYPTO-BOX units for your end users. If you need to update licensing information later we recommend you to use Remote Update (RUMS, see chapter 3.5).

3.5. Creating Remote Update Tool

If you want to allow updates of your end user's CRYPTO-BOX later, you need to generate the Remote Update Tool. This program encapsulates the project and CRYPTO-BOX specific data and can be distributed to the end user together with CRYPTO-BOX. Click the "**Generate**" button on the CB Format main screen and select the path and filename the Remote Update Utility should be extracted to. A detailed description of RUMS can be found in the RUMS Application Notes at marx.com → Support → Documents.

3.6. End User Management

The Smarx Application Framework allows the software distributor to assign CRYPTO-BOX units which were formatted with corresponding AutoCrypt projects settings to particular end-users. Assignment can be done during CRYPTO-BOX Format (see chapter *3.4*). This, however, assumes that the end-users are already defined in the database. Click the "End-Users" button on the SxAF main screen to add end-user descriptions to the database.

There is no need to fill in all the fields for every end-user. For user selection in CRYPTO-BOX Format and for identification during Remote Update, it suffices to fill in the name fields.

If you want to take customer information and licensing information from your own, customized database instead of using the SxAF database, you can implement CRYPTO-BOX configuration into your own system by using command line based tools. See chapter *3.3.13 Generating XML Script for Usage with Command Line Tools* for more information.

4. Command Line Tools

4.1. AutoCrypt - Command Line Version

The command line version of AutoCrypt, AC_Tool.exe, allows you to protect Windows applications and DLLs. The protection process can be controlled within other applications or batch-files. AC_Tool.exe can be found here:

- If you use the **PPK**, open the Control Center and go to the menu point "Drivers & Tools" → "Command Line Tools", or open the folder [SmarxOS PPK root]\Tools\ AC_Tool;
- If you use the **AutoCrypt Wizard Package**, see [AutoCrypt Package root folder]\Tools\ AC_Tool.

AC_Tool is used in combination with SmrxProg, which performs CRYPTO-BOX programming (see chapter *4.2)*.

SmrxProg is also available for Linux and macOS platforms. Refer to the readme file in the "Smarx OS 4 Linux" and "Smarx OS 4 Mac" package for further details.

Parameter description:
AC_Tool.exe <TRX file> <XML file>
where:

<TRX file>	TRX file provided to you by MARX distributor with your customer specific CRYPTO-BOX hardware (cbu_demo.trx for demo CRYPTO-BOX shipped with the Evaluation Kit).
<XML file>	XML file with application protection settings and CRYPTO-BOX configuration also used by SmrxProg for further CRYPTO-BOX programming (see AC_Test.xml as an example) Also initial XML script file can be generated for the active SxAF project (menu "Project" → "Generate Script for SmrxTools"). Even if not using SxAF for protection and license management it still can be useful for XML prototype file creation.

Short explanation on how to use AC_Tool:

- Take an XML file that was exported from AutoCrypt Wizard (see chapter *2.4*) or from SxAF (see chapter *3.3.13*). You can also use a text editor to customize XML data (AC_Test.xml, AC_Local.xml or AC_Network.xml may be used as prototypes).
- If you are going to protect a .NET based application, look at AC_Dotnet.xml for useful hints on .NET configuration options **and check chapter 4 of the readme file in AC_Tool folder for further explanations on the <DOTNET_CONFIG> section!**
- Place the TRX file (you received it with your first CRYPTO-BOX delivery), the XML file created in the previous step and the AC_Tool.exe file into the same directory.
- Run the following command from the console: AC_Tool.exe <TRX file> <XML file>.
- Results will be displayed on the console and output to the AC_TOOL.LOG file.

Have a look at the readme.txt file in AC_Tool folder for more information.

4.2. SmrxProg - Command Line based CRYPTO-BOX® Formatting

SmrxProg is a command line utility for CRYPTO-BOX formatting (programming) through command line switches. Duplicating CB Format functionality (GUI-based component of CRYPTO-BOX formatting in Smarx Application Framework, see chapter *3.4*), SmrxProg can be efficiently used for customer specific scenarios of CRYPTO-BOX programming.
SmrxProg supports AutoCrypt project files (.xml files) generated either with AutoCrypt Wizard or AutoCrypt SxAF.

SmrxProg supports:

- (Re)programming CRYPTO-BOX® Label.
- Creating partitions in CRYPTO-BOX memory (supports partition numbers from 101 to 65535, maximum number of partitions per CRYPTO-BOX is 32).
- Programming Data Objects and network licenses to particular partitions.

- Executing extended partitions operations, like update, delete etc. (see "Extended script format" section in the SmrxProg readme.txt for details).
- (Re)programming of CRYPTO-BOX encryption keys (Private/Session AES Key/IV).
- (Re)programming User Password (UPW) (see "Extended script format" section in the readme.txt).

SmrxProg.exe can be found here:

- If you use the **PPK**, open the Control Center and go to the menu point “Drivers & Tools” → “Command Line Tools”, or open the folder [SmarxOS PPK root]\Tools\ SmrxProg;
- If you use the **AutoCrypt Wizard Package**, see [AutoCrypt Package root folder]\Tools\ SmrxProg.

Parameter description:
SmrxProg.exe <TRX file> <INI file>
or
SmrxProg.exe <TRX file> <XML file>

where:

<TRX file> TRX file provided to you by MARX distributor with your customer specific CRYPTO-BOX hardware (cbu_demo.trx for demo CRYPTO-BOX shipped with the Evaluation Kit).

<INI-file> INI file with CRYPTO-BOX configuration (see Test.ini as example).

<XML file> XML file with application protection settings and CRYPTO-BOX configuration also used by SmrxProg for further CRYPTO-BOX programming (see AC_Test.xml as example).
Also a prototype XML file can created with SxAF for the active project (menu “Project” → ”Generate Script for SmrxTools”). Even if not using SxAF for protection and license management this option can still be useful for automatic creation of the prototype XML script.
Another option is to use Partition Editor (PE) utility for extended XML (script) file generation.

Short explanation on how to use SmrxProg:

- Take an XML file that was exported from AutoCrypt Wizard (see chapter *2.4*) or from SxAF (see chapter *3.3.13*). You can also use a text editor to customize the XML file (AC_Test.xml may be used as prototypes).
- Place the TRX file distributed by MARX, the XML file file obtained on the previous step, and SmrxProg.exe into the same directory.
- Run the following command from the console:
  SmrxProg.exe <TRX file> <XML file>
- Results will be displayed on the console, and directed to the SMRXPROG.LOG file.

Have a look at the readme.txt file in SmrxProg folder for more information and return code description.

5. Distributing Protected Applications to your End Users

After selecting the protection and licensing strategy that best suits your needs, and after protecting your software and formatting the CRYPTO-BOX units, it is time to send everything to the end user. At this point, it is important to include the CRYPTO-BOX drivers (and CBIOS Network Server if network protection is used) with your product. MARX provides a special program, called CBUSetup which can be added to your installation scripts or batch files.

Information on CRYPTO-BOX driver installation can be found in the PPK Control Center under the menu point “Drivers & Tools” → “CRYPTO-BOX Drivers”. For CBIOS Network Server installation, see “Drivers & Tools” → “Network Server”.
Furthermore, separate Application Notes for driver and network server installation are available at marx.com → Support → Documents.

6. FAQ – Frequently Asked Questions

1. **Which files can be protected with AutoCrypt?**

AutoCrypt provides protection for Windows 64/32Bit executables and DLLs as well as .NET-based applications.

Protection of RAD XE 64Bit applications with AutoCrypt was added to PPK 5.90. Since PPK 7.0, protection of applications based on the Windows Presentation Foundation (WPF.NET) is supported, and since PPK 8.17 .NET 6.0 (or higher) applications can be protected.

If your application cannot be protected with AutoCrypt: please contact us - in almost every case we will find a solution!

2. **Is it possible to automate the protection process? I have a lot of applications to protect, and it will be cumbersome to do it manually one by one?**

The command line version of AutoCrypt, AC_Tool.exe provides a high grade of automation: the protection process can be controlled within other applications or batch-files. See section 4.1 for details.

With SmrxProg.exe there is another command line tool available which takes care of CRYPTO-BOX formatting. Please refer to the section 4.2 for more details.

3. **I protected my application successfully, but when I use “CB Format” to configure the CRYPTO-BOX I always get an error “Failed to format CRYPTO-BOX ...”.**

Please make sure that you have selected the correct CRYPTO-BOX hardware profile. The standard profile “cbu_demo” works only with the CRYPTO-BOX contained in the Evaluation Kit. For your customer specific CRYPTO-BOX, choose "Import profile" to import the profile (.trx file) you received from MARX. Refer to section 3.3.4 for more details.

4. **I protected my application successfully, but when I run the protected file, it does not start anymore, or shows some error message!**

Sometimes the protected application is not compatible with AutoCrypt.

In this case we need more details on your protected application. Please contact us or use the Support Ticket System (valid Support Level Option required) to get in touch with us. In almost every case we will find a solution.

5. **My Antivirus software claims that the protected application is infected by a virus!**

Sometimes, the protected application is treaten as infected (false positive). The reason is that AutoCrypt uses similar methods to encrypt the application which are also used by malware. In such cases you can do the following:

- If your application is .NET 4.x based, try our DOTNET_SPLIT_LOAD loader. See explanation of the “Expert settings” in chapter 2.2.8 if you use AutoCrypt Wizard. When using von AC_Tool, check chapter 4.1 in this document and chapter 3 in AC_Tool readme file for further details.
- Define an exception rule or exclude the protected application in the settings of your Antivirus application
- Most manufacturers of Antivirus solutions offer the possibility to report and upload false positives (whitelisting).
- Check the settings of your Antivirus software whether you can disable some detection methods which often lead to false positive detections.

6. **The protected application works fine on the same computer where I did the protection. But when I copy it to another computer, I always get an error “CRYPTO-BOX not found”, “Protection DLL is**

missing or corrupted" or no error message at all.

If you receive a "*CRYPTO-BOX not found*" error please check the following:

- Make sure that you formatted the CRYPTO-BOX with your project settings! If you use AutoCrypt Wizard, see section2.3, for AutoCrypt SxAF see section 3.4. If you use the SmrxProg command line tool, see section 4.2.
- Is the CRYPTO-BOX connected and is the red LED light "on"? If not, please check if the CRYPTO-BOX drivers are installed (see section 5). Our "MARX Analyzer" diagnostic tool checks for installed MARX hardware and software components and provides troubleshooting options. See MARX Analyzer Application Notes for details.

If you receive a "*Protection DLL is missing or corrupted*" message:

During the protection process, a file fmteos.dll (resp. fmteos64.dll for 64 bit applications) is generated in the same folder where AutoCrypt has stored your protected application. Please make sure to deliver this file together with your protected application.

If your protected application is .NET 6.0 or higher and you did not get any message after starting the application, then most probably the Helper.dll is missing. This file will be created automatically in the target folder during protection. Please add this file to your package, too.

7. **I protected my application with AutoCrypt. But now I have an updated version of my application and I would like to protect it with AutoCrypt so that it works with an existing CRYPTO-BOX which is already in the field. How do I do that?**

a) For AutoCrypt Wizard:

Open the project and go to the "Application Settings" screen (see chapter *2.2.8*). Select the updated application under "Source file to protect", and protect the application as described in chapter *2.2.9*). Make sure to leave all other project settings untouched!

b) For AutoCrypt SxAF:
Open the existing project in the Smarx Application Framework (SxAF), and choose the application you want to update in the tree view under "Applications". Now click the button "Update Application", select the new application you want to protect, and click "OK". Then click the "Protect Application" button to protect and store the new protected application.

c) For AC_Tool:
Just protect the new application with he same XML file as the old one.

IMPORTANT: Do not use the "Application Checksum" DataObject if you plan to preserve compatibility with further updates!

8. **I protected several components/modules of my application (several EXE and/or DLL files) with AutoCrypt. But the protected components do not work correctly: sometimes I get error messages, and licensing (execution counter, network licenses) seems to work incorrect!**

Please note that certain protection and licensing options cannot be used at the same time for all components/modules:

- Use the "periodic check" option only for one protected module (for example, only for the EXE, not for the EXE and further DLLs).
- Do not share partitions for applications/modules, if they are running at the same time – this may lead to unexpected behavior and error messages. See section *3.3.5* for further information on sharing partitions.
- If you share license options such as execution counters for multiple applications/modules, it will be decremented for each application when it is started! Therefore its recommended to use a separate partition in that case.