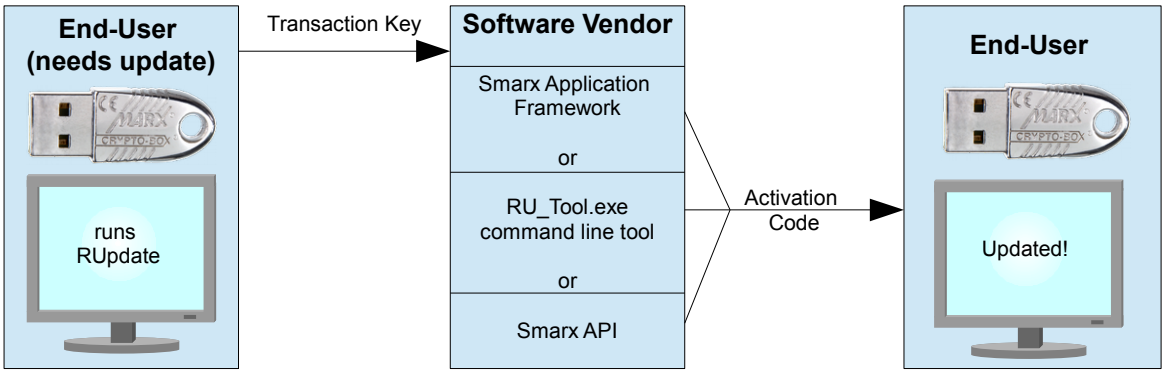
Remote Update (RUMS) AN-4

Purpose of Application: Remote CRYPTO-BOX® Updates at end-user sites
Version: Smarx OS PPK 5.75 and higher
Last Update: 15 November 2016 0-12Jun012ks(AN4_RUMS).odt
Target Operating Systems: Windows 10/8/7/Vista (32 & 64 bit)
Target Processor Platforms: Intel x86
Access to source code needed (of protection application): ☑ Yes ☐ No
Supported Programming Tools: see Smarx Compendium, chapter 2.5
Applicable for Product: CRYPTO-BOX® SC / XS / Versa

Remote Update flow of operation



- The end-user runs the RUpdate utility to generate a Transaction key and send it to the software vendor.
- The software vendor processes the Transaction Key either with the Smarx Application Framework, the RU_Tool.exe command line tool or the Smarx API to create an Activation Code with update information which he sends back to the end-user.
- The end-user receives the Activation Code and feeds it to the RUpdate utility to update the license information in the CRYPTO-BOX.

Updating the CRYPTO-BOX® Remotely

The Remote Update Management System (RUMS) is a convenient way to remotely update licensing information which were programmed inside the CRYPTO-BOX using the Smarx Application Framework (SxAF), SmrxProg command line tool or the Smarx API. The update process is controlled through encrypted configuration files which can be sent via email, no programming efforts required.

RUMS provides the following main features:

- Updating and extending license data in the CRYPTO-BOX memory, reprogramming of Rijndael Private and Rijndael Session Keys.
- Update is possible via the GUI-based Smarx Application Framework - no programming efforts are required.
- The command line version of RUMS can be controlled within 3rd party applications or batch files.
- Furthermore, software developers can use the RFP API to integrate the remote update functionality directly into the source code of their applications.

Table of Contents

1. Remote Update Management System (RUMS): Overview

Remote Update Management System (RUMS) is part of the Smarx Protection Kit (PPK) for the CRYPTO-BOX. It provides software manufacturers and distributors a convenient way to remotely update the content of a CRYPTO-BOX which is already in possession of the end user without the need to ship the CRYPTO-BOX from the end user to the software distributor and back. Thus, it allows to prolong expiration periods for software licenses, to activate or deactivate features, and more.

Remote Updates can be processed in different ways:

- Using the Remote Update Management System (RUMS) as part of the Smarx Application Framework (see **chapter 2**).
- Using RU_Tool.exe, a command line based tool (see **chapter 3**)
- Using SxFormat.exe, a command line based tool (see **chapter 4**). In contrast to RU_tool.exe, SxFormat offers one-way functionality (update does not need to be initiated by the end-user)
- Using the RFP API (Remote Update API for software developers, see **chapter 5**)

The advantages of each solution mentioned above are illustrated at the beginning of the corresponding chapter to help you select the solution which best fits your needs.

Remote Update offers the following features (Smarx Protection Kit 5.60 and up):

- Updating of existing or creation of new Data Objects in existing or new CRYPTO-BOX partitions. The Data Objects can be Expiration Date/Time, Counters, Network Licenses or individual Memory Objects.
- Creating new partitions or extension of existing partitions in CRYPTO-BOX memory.
- Update of CRYPTO-BOX Rijndael Session and Rijndael Private Keys.
- Update of AES Keys in RAM5 and RSA Keys in RAM4 zone (CRYPTO-BOX SC only)

Remote Update can be used in combination with the following solutions provided by MARX:

- AutoCrypt (automatic protection)
- Implementation with API
- Media Protection
- Document Protection

The Remote Update functionality is available as an option. Please contact your MARX distributor for more information.

2. Remote Update via the Smarx® Application Framework

2.1. Overview

The Smarx Application Framework (SxAF) provides quick and easy protection of applications, documents and media files. Software, documents and media files can be protected via the project types "AutoCrypt", "Document Protection" or "Media Protection" without any programming efforts. But even users considering implementation with API will profit from SxAF because it allows quick and easy configuration of CRYPTO-BOX units with licensing information for the end users.

Advantages of Remote Update with the Smarx OS Application Framework:

- Quick and easy creation of projects and processing of remote updates with a graphical interface, no programming efforts required.
- Integrated end user database.
- Overview about existing projects and processed updates.

Specialty:

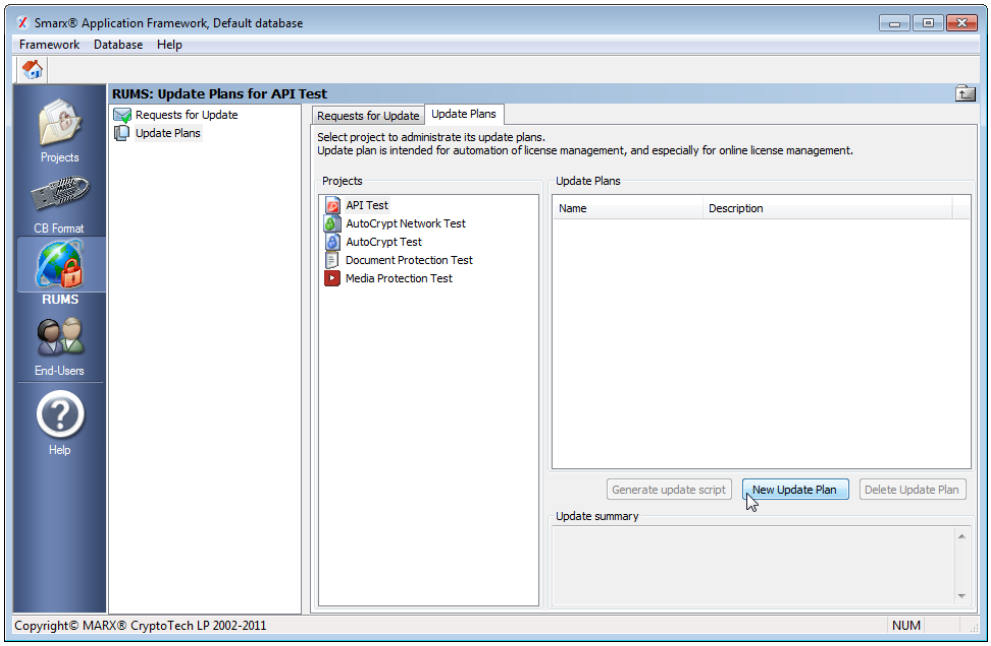
- Integrated database, incorporating existing end user databases requires some effort.

2.2. Remote Update Plans

RUMS allows you to define various Update Plans for your project according to your licensing approach, simplifying and automating additional remote license update procedures for the protected product (Update Plan can be used when proceeding Remote Update Request, see chapter *2.5*). This is especially useful if you consider intensive licensing management as a part of your marketing strategy (subscriptions, updates, extensions, etc.). Every Update Plan defines a set of updates to be completed on licensing Data Objects (which are included in the license associated with protected product). You can define as many Update Plans as you need. Consider implementing them along with Product Editions. Update Plans can really help you with license management automation.

It is not necessary to define Update Plans, consider them as a convenient option - you can process update requests manually. But smart Update Plans simplify the processing of multiple typical updates.

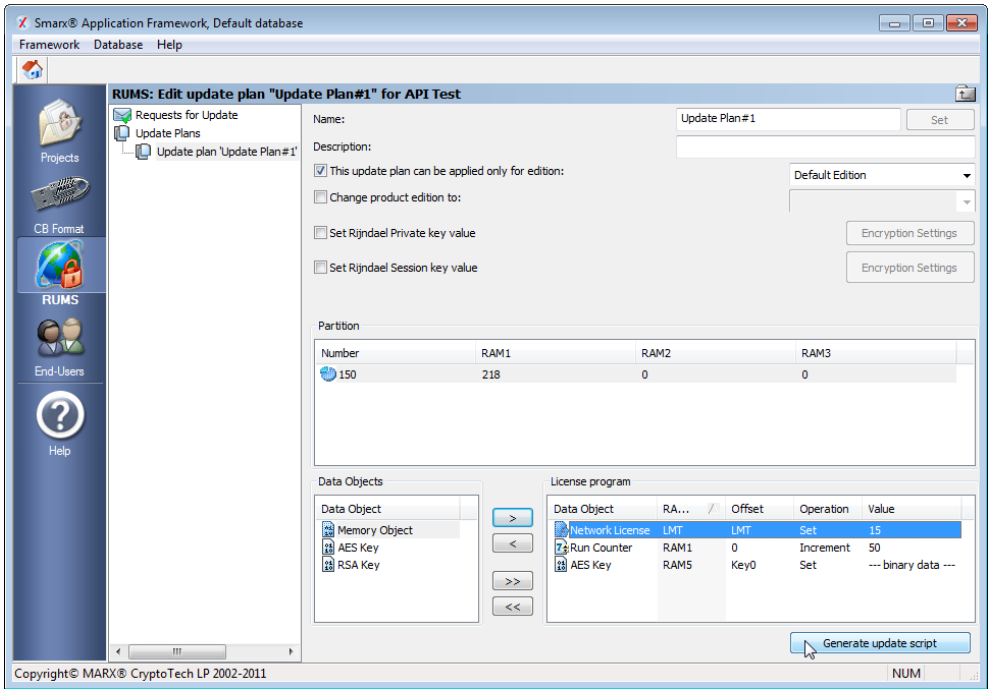
To create an Update Plan, start the Smarx Application Framework and select “RUMS” from the left menu bar, then click the "Update Plans" tab. In the following window, select the project for which you want to create an Update Plan on the left, then click the "New Update Plan" button on the right:



The following page allows you to define required license updates for every application (in case of AutoCrypt) or partition (in case of Implementation with API) included into the project (if your project includes more than one application).

Select an application from the “Application Modules” list-view if you have an AutoCrypt project, or an partition from the "Partition" list-view if you have an API project, and then choose licensing Data Objects of the current application you want to be included in the update sequence:
Use the “>” button to add selected Data Object to the update sequence. The “<” button removes selected Data Object from the update sequence. The “>>” button adds all licensing Data Objects to the update sequence and sets default values for all Data Objects (depending on the edition selected in the “Change product edition” combo-box). The “<<” button removes all Data Objects from the update sequence. Double-click on any Data Object to select the operation and new value. This data will be included in the update sequence.

The “Generate update script” button on the lower right displays the “Update Script Generation” dialog. The Update script will include all license changes of the Update Plan and can be used later by the RU_Tool (see chapter 3) or the Online License Management (OLM) component. For more information on OLM have a look at the Smarx Compendium, chapter 6.



After you have specified your Update Plans, you may continue with creating the Remote Update Utility (see next chapter).

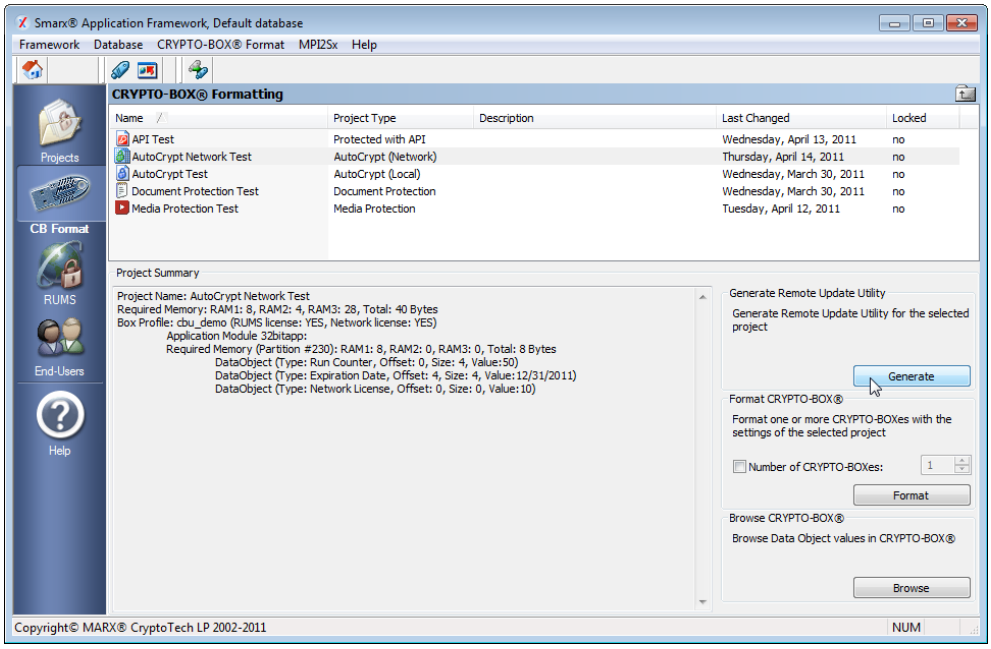
2.3. Creating the Remote Update Utility for the end user

A prerequisite for updating a CRYPTO-BOX is an existing project in the Smarx Application Framework (SxAF) which was used to format the CRYPTO-BOX (see Smarx Compendium chapter 4 for detailed information on how to create projects).

The end user who is in possession of the CRYPTO-BOX needs to create a Transaction Request first. To do so, he will need the Remote Update (RUpdate) Utility which is shipped to him by the software distributor (preferably together with the protected software and CRYPTO-BOX).

The software distributor generates the RUpdate Utility by starting SxAF, choosing "CB Format", selecting the corresponding project and clicking "Generate Remote Update Utility".

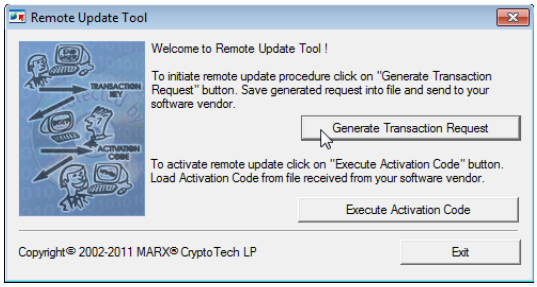
Generating the Remote Update Utility assumes a valid RUMS license. This license can be obtained from your MARX distributor in the form of an updated hardware profile (.TRX file). The profile needs to be imported to the SxAF project by selecting the "Import profile" button (under the "General Settings" section of the project). If no RUMS license is available, the "Generate" button is deactivated.



Now the RUpdate Utility (3 files) will be stored in the selected folder. These files need to be provided to the end user.

2.4. Initiating Remote Updates

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:



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. Proceeding with Update Requests

After the software distributor has received the end user’s request file (*.rutr), it needs to be loaded into Remote Update Manager for further processing. The distributor starts the Smarx Application Framework and selects “RUMS” from the left menu bar, then presses the “Load” button to open the transaction request he

received from the end user. After that, the request will be decrypted. Now, it is possible to create a list of updates for several Data Objects. There are three possibilities:

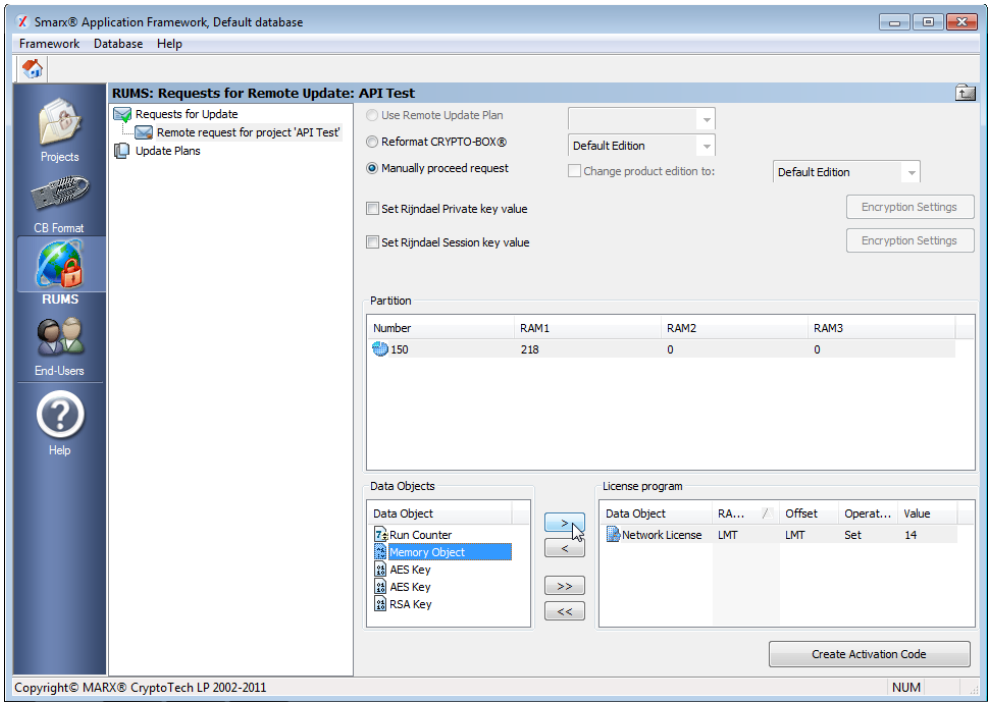
- The option "Use Remote Update Plan" allows you to use Update Plans with predefined update settings (see chapter *2.2. Remote Update Plans*). Once you have created your desired Update Plans, you do not need to spend any time choosing data objects, operations and values anymore. All you have to do is to select a suitable Update Plan and press the “Create Activation Code” button (see chapter *2.7. Create Activation Code*.
- If you have created different Product Editions for your project (see Smarx Compendium, chapter 4 for more details on Product Editions), you may change the existing Product Edition to another one (for instance, changing a “Default Edition” with some limitations into a “Gold Edition” with extended features or usage time). To do so, select the option “Reformat CRYPTO-BOX” and continue with chapter *2.7. Create Activation Code*.
- Select and edit Data Objects manually using the option “Manually proceed request” (see next chapter for details).

2.6. Proceeding with Update Requests Manually

Select an application (for AutoCrypt projects) or partition (for API-based projects) in the window on the middle of the page, then select Data Objects to update below.

Select the Data Object you want to change from the left part of the window and click the “>” button to add the Data Object to the update sequence (right window). A new window opens which allows you to change the Data Object settings.

Repeat the steps above for other applications/partitions or Data Objects.



2.7. Create Activation Code

After you have selected the desired updates, click “Create Activation Code” to generate an activation code and save it as an external file (*.ruac). This file has to be sent to the end user.

2.8. Executing the Activation Code on the End User Side

As soon as the-end user receives the activation code, a Remote Update can be performed. 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) permits only a ONE time update and prevents multiple unauthorized activations.

3. RU_Tool.exe - Command Line Utility for Remote Update Management

3.1. Overview

RU_Tool.exe allows you to perform remote update of the CRYPTO-BOX hardware. Duplicating RUMS functionality (GUI-based component to perform CRYPTO-BOX updates in Smarx Application Framework, see chapter *2. Remote Update via the Smarx® Application Framework*), RU_Tool can be integrated to your application to automate customer specific scenarios of remote update.

In opposite to RU_Tool.exe, SxFormat.exe (see chapter *4*) does not require any data (Transaction Request – see chapter *3.3*) from the end-user to perform an update the CRYPTO-BOX.

With RU_Tool, the following actions can be performed from command-line:

- extracting the Remote Update Utility for the end user
- processing of remote update requests obtained from the end user and generation activation codes for remote update
- retrieving CRYPTO-BOX serial number (BoxName, hex value) from remote update request

RU_Tool.exe can also be used to add new partitions to an existing CRYPTO-BOX at the end user side. Please see the readme.txt file in RU_Tool folder, Appendix B, for possible use case scenarios.

After Protection Kit installation, RU_Tool can be found in the "Smarx Tools" section in the Smarx PPK Control Center or in the folder “\MARX CryptoTech\SmarxOS PPK\Tools\RU_Tool”. The readme.txt file in this folder contains the latest information on RU_Tool, as well as a description of its return codes.

RU_Tool can be used in combination with other command line tools: SmrxProg.exe (for CRYPTO-BOX formatting), AC_Tool.exe (for automatic protection) and Doc_Tool.exe (for protection of documents). See Smarx PPK Control Center, section "Smarx Tools" for more details.

Advantages of Remote Update with RU_Tool.exe:

- Can be adapted to nearly every application, database or specific distributions strategy.

- High grade of automation, for instance automatic creation of specific update codes directly from the distribution system of the software distributor.
- Allows complete redistribution of the CYPTO-BOX memory (Creating new partitions and Data Objects).

Specialty:

- Update script (XML file) needs to be created or generated - can be done either via the Smarx Application Framework (SxAF), manually, or generated by separate tools (may require additional programming efforts).

3.2. Creating the Remote Update Utility for the end user

The end user who is in possession of the CRYPTO-BOX needs to create an update request first. For this, the Remote Update Utility is required. This can be done either with the Smarx OS Application Framework (SxAF, see chapter *2.3. Creating the Remote Update Utility for the end user* for more information) or with the RU_Tool.exe itself by running it with "-extract" parameter.

Parameter description:

RU_Tool.exe -extract <TRX-file> <EXE-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)
<EXE-file>	EXE file name for RUpdate utility

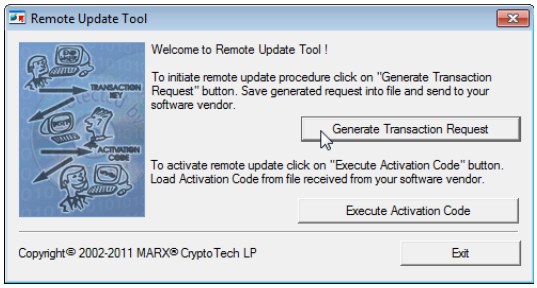
Result:

<EXE-file> file for RUpdate utility will be extracted together with *.409 and *.407 files containing RUpdate text resources in English and German. Results will be displayed on console and output to RU_Tool.LOG file.

Generating the Remote Update Utility assumes a valid RUMS license. This license can be obtained from your MARX distributor in the form of an updated hardware profile (.TRX file). If no RUMS license is available, an error message will be displayed.

3.3. Initiating Remote Updates

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:



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.

The RUpdate Utility has a graphical interface but can be controlled via command line switches as well. Have a look at the readme.txt (Appendix A) in RU_Tool folder for more explanations.

3.4. Retrieve CRYPTO-BOX Serialnumber (BoxName)

If required, the software distributor can obtain the CRYPTO-BOX Serialnumber from the update request.

Parameter description:

RU_Tool.exe -getboxname <TRX-file> <RUTR-file> [<LOG-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)
<RUTR-file>	Remote update request file from where the BoxName will be retrieved
<LOG-file>	Optional log-file where the BoxName will be stored in. Without <LOG-file> being specified the BoxName will be displayed in console output.

3.5. Preparing Remote Update Data to Reprogram the CRYPTO-BOX®

XML script file:

An XML file has to be prepared, containing information about the updates to be applied to the CRYPTO-BOX.

The readme.txt in RU_Tool folder, Appendix B, shows some use-case scenarios on xml update script generation.

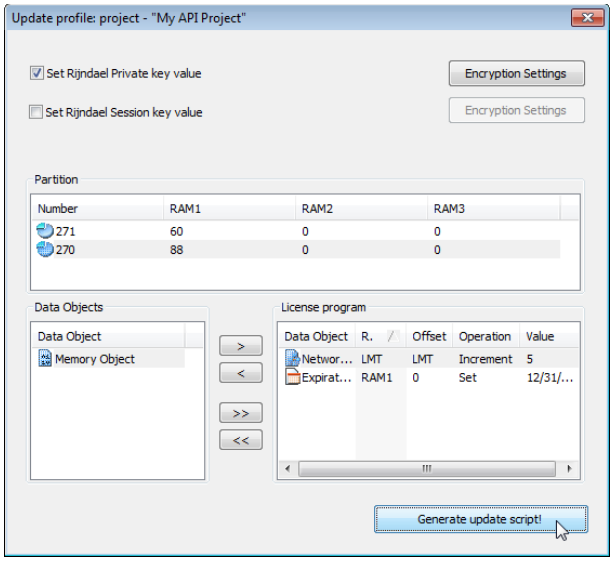
The AC_Local_Update.xml file in RU_Tool folder can be used as template.

Creating the XML script file automatically:

Especially when implementation with API is used, it may be difficult to prepare the update sequence manually using description in the readme.txt in RU_Tool folder and *AC_Local_Update.xml* as template. The

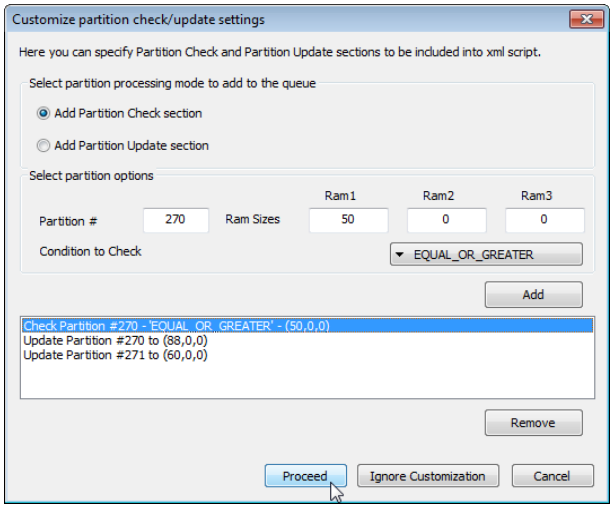
Smarx Application Framework (SxAF) simplifies this procedure:

- Start the SxAF and open an existing "AutoCrypt" or "Implementation with API" project or create a new one (see Smarx Compendium chapter 4 for more information).
- Under "General Settings", click on “Import profile” to import the hardware profile (.TRX file) from the CDROM you received along with your customer specific CRYPTO-BOX modules from MARX.
- Define the partitions and Data Objects you want to update later.
- Click on the "CB Format" tab in the left navigation bar and format the CRYPTO-BOX units for your end users. If the CRYPTO-BOX is already at the end user side you can skip this step.
- Switch back to the "Projects" tab and open your project again.
- In the upper menu bar, click on "Project" → "Generate update script for RU_Tool".
- Select an application from the “Application Modules” list-view if you have an AutoCrypt project, or an partition from the "Partition" list-view if you have an API project, and then choose licensing Data Objects of the current application you want to be included in the update sequence
- Use the “>” button to add selected Data Object to the update sequence. The “<” button removes selected Data Object from the update sequence. The “>>” button adds all licensing Data Objects to the update sequence and sets default values for all Data Objects. The “<<” button removes all Data Objects from the update sequence. Double-click on any Data Object to select the operation and new value. This data will be included in the update sequence.



- Click the “Generate update script!” button on the lower right to continue.
- Now you will be asked if you want to customize check conditions before the update will be performed. If you choose "Yes", a new window will open:
- With "Add Partition Check section" you can check if a certain partition with a certain size is already available on the CRYPTO-BOX - if not the update will not be processed. This can be useful if you have CRYPTO-BOX units with different settings for different applications in the field and you want to allow to perform updates for a certain CYPTO-BOX unit only. The screenshot below demonstrates how it can be

used: the update will be performed only if an partition #270 with RAM1 size of minimum 50 bytes already exists in the target CRYPTO-BOX ("EQUAL_OR_GREATER").



- With "Add Partition Update section" you can add new partitions or update the size of existing partitions. This allows you to update the internal partition structure of the CRYPTO-BOX directly by your end user.
- Click "Proceed" to continue to the Save dialog. Choose the name for the XML file and click "Save".

We strongly recommend you run a test to check if the generated XML file works correctly before you send the activation code to your end users!

3.6. Generate Activation Code

Run RU_Tool with "-update" parameter to generate the Activation Code with encrypted update information which has to be sent to the end user.

Parameter description:

RU_Tool.exe -update <TRX-file> <XML-file> <RUTR-file> <RUAC-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 updates information (see AC_Local_Update.xml as example)
<RUTR-file>	Remote update request file
<RUAC-file>	File name for remote update activation code

The Activation code will be saved to <RUAC-file>; results will be displayed on console and output to RU_Tool.LOG file.

3.7. Executing the Activation Code on the End User Side

The .RUAC file has to be sent to the end user. He will run the RUpdate Utility, load the .RUAC file and execute remote update. The unique transaction code (generated for each request) permits only a ONE time

update and prevents multiple unauthorized activations.

The Remote Update Utility has a graphical interface, but can be controlled via command line switches as well. Have a look at the readme.txt (Appendix A) in RU_Tool folder for more explanations.

4. SxFormat.exe - Command Line Utility for updating the CRYPTO-BOX®

4.1. Overview

SxFormat.exe allows you to perform remote update of the CRYPTO-BOX hardware. In opposite to RU_Tool.exe (see chapter 3), SxFormat.exe does not require any data from the end-user to perform an update the CRYPTO-BOX.

SxFormat is generated with the SmrxProg.exe command-line tool. The update information are specified via XML files.

After Protection Kit installation, SmrxProg.exe can be found in the "Smarx Tools" section in the Smarx PPK Control Center or in the folder “\MARX CryptoTech\SmarxOS PPK\Tools\SmrxProg”. The readme.txt file in this folder contains the latest information on SmrxProg.exe, as well as a description of its return codes.

Advantages of Remote Update with SxFormat.exe:

- Can be adapted to nearly every application, database or specific distributions strategy.
- High grade of automation, for instance automatic creation of specific update codes directly from the distribution system of the software distributor.
- Allows complete deletion of the CYPTO-BOX memory and creation of new partitions and Data Objects).
- Update does not need to be initiated by the end-user (as required with RUMS and RU_Tool).

Specialty:

- Less control about the update execution because no data about the CRYPTO-BOX of the end-user will be collected.
- Update script (XML file) needs to be created or generated - can be done either via the Smarx Application Framework (SxAF), manually, or generated by separate tools (may require additional programming efforts).

4.2. Preparing Remote Update Data to Reprogram the CRYPTO-BOX®

XML script file:

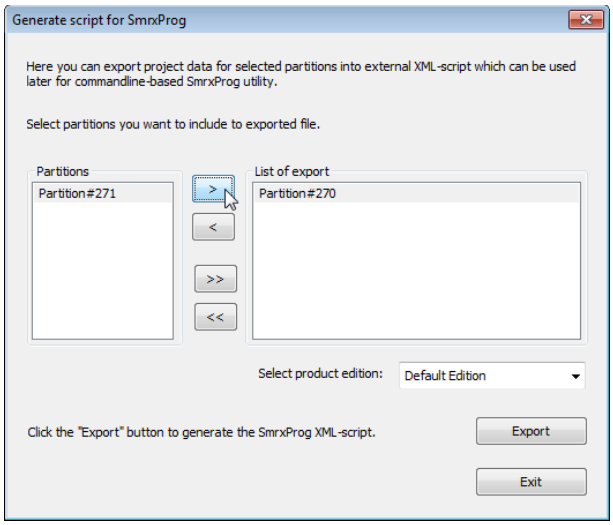
An XML file has to be prepared, containing information about the updates to be applied to the CRYPTO-BOX. The readme.txt in SmrxProg folder, Appendix A, shows some use-case scenarios on xml update script generation.

The AC_Local_Update.xml file in RU_Tool folder can be used as template.

Creating the XML script file automatically:

Especially when implementation with API is used, it may be difficult to prepare the update sequence manually using description in the readme.txt in SmrxProg folder and *AC_Local_Update.xml* as template. The Smarx OS Application Framework (SxAF) simplifies this procedure:

- Start the SxAF and open an existing "AutoCrypt" or "Implementation with API" project or create a new one (see Smarx Compendium chapter 4 for more information).
- Under "General Settings", click on “Import profile” to import the hardware profile (.TRX file) from the CDROM you received together with your customer specific CRYPTO-BOX modules from MARX.
- Define the partitions and Data Objects you want to update.
- In the upper menu bar, click on "Project" → "Generate script for SmrxProg".
- Use the “>” button to select the applications (in case of AutoCrypt project) or partitions (in case of API project) you want to write into the target CRYPTO-BOX. The “<” button removes selected objects from the update sequence. The “>>” button adds all objects to the update sequence. The “<<” button removes all objects from the update sequence.



- Click the “Export” button on the lower right to continue to the Save dialog. Choose the name for the XML file and click "Save".

The “Extended Script Format” (see SmrxProg readme file, Appendix A) allows you to incorporate certain checks of the target CRYPTO-BOX into the script file to refuse formatting of CRYPTO-BOX units which does not meet certain conditions. This allows you to prevent end-users from updating CRYPTO-BOX units which are not authorized to receive this update. Corresponding XML templates in extended script format can be created using instructions in chapter *3.5*.

We strongly recommend you to make a test run and check if the generated XML file works correctly before you send the Activation Code to your end users!

5. Remote Update with Remote Update API (RFP API) for developers

Using the Remote Update API (part of the Smarx API) offers software developers maximum flexibility. This allows for you to integrate the update process directly into your application. MARX provides libraries and samplecode for several compilers, for instance:

- Static library for C++ (Visual Studio, 32 and 64Bit)
- CBIOS.RFP as part of CBIOS4NET (object oriented, component based API for .NET developers)

A detailed documentation of the available API commands can be found in the corresponding Developer's Guide for your compiler, check the "Documentation" tab in the Smarx PPK Control Center.

Advantages of Remote Update with the Remote Update API:

- Can be fully integrated into the application
- Maximum flexibility

Specialty:

- Programming efforts required

Check the "Documentation" tab in the Smarx PPK Control Center for proper documentation of the Remote Update API for your compiler (Remote Update API Reference for C++ or CBIOS4NET API Reference for C#).

6. FAQ (frequently asked questions)

1. When using the command line tools to generate the Remote Update Utility or SxFormat.exe, I always get an error message “Error: Decryption of 'C:\...\CBU_XSN-xxxxx.trx' failed - your CRYPTO-BOX(R) firmware 2.2 or higher should be attached”!

To generate these tools it is necessary to attach a CRYPTO-BOX to the computer which matches to the specified TRX-file.

2. The Remote Update functionality does not work for me: In SxAF the option “Generate Remote Update Utility” is grayed out, and with the command line tools I always get “Error: RUMS not licensed”!

The Remote Update functionality is available as an option. Please contact your MARX distributor for more information.

3. Why does Remote Update always require to be initiated the update by the end-user? This means too much efforts for me and my end-users!

With "SxFormat" you can update the CRYPTO-BOX without the need to have the transaction request from the end-user. For more details please refer to chapter *4*.

4. Is Remote Update resp. SxFormat available for Linux or Mac OS X?

The tools and libraries for Remote Update are currently available for Windows only. On request we can offer Remote Update functionality for other platforms – please refer to our Technical Support.

5. I'm using the RUMS option in SxAF for updating CRYPTO-BOX units remotely. Now I need to write additional data (partitions and data objects) to the CRYPTO-BOX units of my end-users. But RUMS only allows to update existing partitions! How can I add new data?

Please use the command line based RU_Tool.exe in that case. A detailed description can be found in chapter *3* of this document.

6. Is the Remote Update API available for other compilers than C++ and C#?

If you need libraries for other compilers or platforms please contact our Technical Support.

Remote Update (RUMS) AN-4

CRYPTO-BOX® Data Sheet

	CRYPTO-BOX SC (CBU SC)	CRYPTO-BOX XS/Versa (CBU XS/Versa)
		
Controller chip	RISC Smart Card Processor with USB Interface	RISC Smart Card Processor with USB Interface
Chip certification	EAL4+	EAL4+
Supported operating systems	Windows, Linux, Mac OS X, iOS, Android	Windows, Linux, Mac OS X, iOS, Android
In hardware implemented algorithms	AES 128 Bit, RSA (up to 2048 Bit key length), others (for example: ECC) on request	AES 128 Bit in hardware, RSA up to 2048 Bit key length on driver level
Memory size (complete)	72KByte, ca. 32KByte free	4, 32 or 64 KByte
Internal memory read/write performance	ca. 80kByte/s	ca. 12kByte/s
Password (PIN/PUK)	up to 16 Byte length	
Case & LED	Designer metal housing, cast zinc, with LED display of operating status, eye for key ring/lanyard	
Connector	USB Type A	
Memory programming	minimum 100,000 write cycles	
Data retention time	minimum 10 years	
Conformity & Certifications	FCC, CE, RoHS, WEEE, USB logo	
Dimensions	14 x 7 x 32,5 mm / 0.55” x 0.28” 1.28”	14 x 7 x 32,5 mm / 0.55” x 0.28” 1.28”
Weight	7,5g	7,5g
Temperature	-10°C to +70°C / 14°F to 158°F	
Humidity	0% to 95% relative humidity	

CRYPTO-BOX Certifications

Evaluation Kit

www.marx.com/eval

MARX Software Security GmbH
Vohburger Strasse 68
85104 Wackerstein, Germany
Phone: +49 (0) 8403 / 9295-0
Fax: +49 (0) 8403 / 9295-40
contact-de@marx.com

MARX CryptoTech LP
489 South Hill Street
Buford, GA 30518 U.S.A.
Phone: (+1) 770 904 0369
Fax: (+1) 678 730 1804
contact@marx.com

www.marx.com