



Described Version: Smarx OS PPK 5.60

Also Applicable For: Smarx OS PPK 5.65 and higher

Target Platforms: Windows 7 32/64, Vista 32/64, XP, 2000

MARX Hardware: CRYPTO-BOX[®] SC, CRYPTO-BOX[®] USB

Last Update: 9. Aug 2011

Implementation with API

SmarxOS unleashes the full power of the CRYPTO-BOX USB System, providing you with efficient, flexible and convenient ways to protect your software and data against piracy and unauthorized usage. It supports Windows 7/Vista/XP/2000, Linux, and Mac OS X. Depending on the platform (OS) and programming environment used, Smarx[®] API (CBIOS local and network API, Do API, RFP API) is supported in different formats, as:

- Static libraries
- Dynamic libraries (DLL)
- Native DLL/SO/JNILIB
- Managed DLL
- COM/ActiveX



CRYPTO-BOX[®] USB



- Quick and easy protection of Windows and Linux applications with AutoCrypt
- Individual implementations with API for all common programming languages, incl. .NET
- The CRYPTO-BOX system can be customized according to individual requirements
- Multi-platform support: Windows, Linux and Mac OS X
- Unique and stable metal case, optional with customer-specific color and labeling
- Internal secure memory of 4-64 kB
- Network and remote update capability
- AES/Rijndael encryption on-chip
- RSA support on-chip (CRYPTO-BOX SC) or on driver level (CRYPTO-BOX USB)

Table of Contents

1. Introduction.....	2
2. Using Smarx® API for different environments.....	2
3. Table of API Samples.....	3

1. Introduction

SmarxOS allows MARX customers adding hardware protection to software products running under all popular platforms:

- Windows 32 and 64 bit
- Linux 32 and 64 bit
- Mac OS X (Intel x86 and PPC hardware architecture)

Most programming environments for these platforms are supported. Smarx API (including CBIOS and DO API) is implemented for all operating systems mentioned above as one source code used for all platforms. This approach allows developers to use Smarx API even for cross-platform development. See more details below.

2. Using Smarx® API for different environments

Depending on the platform (OS) and programming environment used, Smarx API (CBIOS local and network API, DO API, RFP API) is supported in different formats, as:

- Static libraries
- Dynamic libraries (DLL)
- .NET assembly (Managed DLL)
- COM/ActiveX
- Native DLL/SO

Static libraries are the most secure way of linkage. They are provided for most of supported programming environments under Windows, Linux and Mac OSX platforms, including: Microsoft C/C++, Borland C Builder, Delphi environments, and GCC.

Dynamic libraries (DLLs) allow easy, but less secure linkage. DLL based implementation should be considered only if for some reasons no other options can be used (static library, COM). When using DLL try to improve the level of protection and licensing logic for your application (using hardware based encryption, keeping vital data in the CRYPTO-BOX, using parallel threads, etc.), making it difficult to emulate this logic by replacing the DLL. DLLs are provided for all environments of Windows (x86 and x64) platform, there is a special DLL (CBIOSVB6.DLL) for Visual Basic 6.0 environment.

For .NET developers, CBIOS4NET is the most recommended approach. It provides .NET developers with an object oriented, component based approach, simplifying integration of protection and licensing to .NET applications.

MngCBIOS.DLL was historically first CBIOS implementation for .NET environment. It is obsolete, switching to CBIOS4NET is highly recommended.

COM/ActiveX is the Windows platform specific interface standard. This interface format is universal and can be used from almost any Windows programming environment. Required Smarx OS ActiveX objects are included to CBUSetup.exe driver installation utility (see separate Application Note on CRYPTO-BOX driver installation) and are installed and properly registered together with CRYPTO-BOX USB driver.

Native DLL/native SO are specific to Java environment (Windows and Linux correspondingly).

3. Table of API Samples

The following table contains programming environments currently supported by Smarx API:

Platform	SmarxOS Interface	Language	Programming Environment
Windows	CBIOS	C/C++	MSVC 6.0, MSVS 2003
			MSVS 2005, 2008, 2010 (Win32/64)
			Borland 5, 6, BDS2006, RAD2007-2011
			MinGW
			Qt4
		C#	MSVS 2003
			MSVS 2005, 2008, 2010 (Win32/64)
			MSSQL 2005, 2008
		Delphi	Delphi 5, 6, 7, 2005, BDS2006, RAD 2007-2011
		Java	JDK 1.6
		VB	MSVB 6.0, MSVS2003
			MSVS 2005-2010 (Win32/64)
		VBA	MS Access
			MS Excel
		Other	LabVIEW 8.5
			MS Visual FoxPro 9.0
			Micro Focus Visual COBOL 2010 R3
			Intel Visual Fortran XE 2011

API Samples

Platform	SmarxOS Interface	Language	Programming Environment	
			Digital Mars D	
			F#	
			REALbasic 2006	
			DarkBASIC	
			Scala	
		XSMRXCOM	C#	MSVS 2005/2008
		DO	C/C++	MSVC 6.0, MSVS 2003
				MSVS 2005, 2008, 2010 (Win32/64)
				Borland 5, 6, BDS2006, RAD2007-2011
				MinGW
			C#	MSVS 2003
				MSVS 2005, 2008, 2010 (Win32/64)
				BDS2006
			Delphi	Delphi 5, 6, 7, 2005, BDS2006, RAD 2007-2011
			Java	JDK 1.6
			VB	MSVB 6.0, MSVS2003
		MSVS 2005-2010 (Win32/64)		
		Other	MS Visual FoxPro 9.0	
		RFP	C/C++	MSVC 6.0, MSVS 2003
				MSVS 2005, 2008, 2010
				Borland 5, 6, BDS2006, RAD2007-2011
				MinGW
			C#	MSVS2003
				MSVS 2005-2010 (Win32/64)
			Delphi	Delphi 6, 7, 2005, BDS2006, RAD 2007-2011
			Java	JDK 1.6
			VB	MSVS2003
				MSVS 2005-2010 (Win32/64)
		CBIOS Extended	C#	MSVS 2005
				MSVS 2005 DataProtection4MediaFiles
	C/C++		MSVC 2005	
			MSVC 2005 MFC ACAD DWFViewer Demo	
			MSVC 2005 Data Encryption Demo	



API Samples

Platform	SmarxOS Interface	Language	Programming Environment
			Msvc 2005 Secured PDF Viewer Demo
			Msvc 2005 Secured Web Viewer Demo
Linux	CBIOS	C/C++	GCC
			Qt 4
		Java	JDK 1.6
		Delphi	Kylix
	Basic	REALbasic 2006	
	DO	C/C++	GCC
	RFP	C/C++	GCC
Mac OS X	CBIOS	C/C++	GCC
			Qt 4
		Java	JDK 1.6
	Basic	REALbasic 2006	
	DO	C/C++	GCC



All samples can be found in the Smarx OS Protection Kit (PPK) - see PPK Control Center, section "Software Protection with Smarx API for Developers" for more information.
Please contact us if you need libraries or sample code for further environments!

CRYPTO-BOX Data Sheet

	CRYPTO-BOX SC (CBU SC)	CRYPTO-BOX USB (CBU)
		
Controller-Chip	8/16 bit RISC Smart Card Processor	8 Bit microcontroller with USB interface
Chip Certification	EAL4+ / ISO 7816	WHQL (Microsoft)
Supported Operating Systems	Windows 7/Vista/XP/2000, Linux, Mac OS X	Windows 7/Vista/XP/2000, Linux, Mac OS X
Hardware integrated 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, min. 32KByte free	4, 32 or 64 KByte
Read/write performance of internal memory	ca. 80kByte/s	ca. 1,3kByte/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	
Programming the memory	typically more than one million cycles, 100,000 guaranteed	
Data retention time	minimum 10 years	
Compliance & Certifications	FCC, CE (TÜV Rheinland - Germany), RoHS, USB-Logo	
Dimensions	15 x 6 x 38 mm	14 x 8 x 36 mm
Weight	10,5g	9,2g
Temperature	0°C to +60°C / 32°F to 140°F	
Humidity	0% to 95% relative humidity	

CRYPTO-BOX Certifications



All brands, trademarks and registered trademarks are the property of their respective owners.

CRYPTO-BOX Evaluation Kit

www.cryptotech.com/eval

MARX Software Security GmbH

Vohburger Strasse 68
85104 Wackerstein, Germany
Phone: +49 (0) 8403 / 9295-0
Fax: +49 (0) 8403 / 1500
contact-de@cryptotech.com

MARX CryptoTech LP

3355 Annandale Lane
Suwanee, GA 30024 U.S.A.
Phone: (+1) 770 904 0369
Fax: (+1) 770 904 3893
contact@cryptotech.com

www.cryptotech.com