CmEmbedded Guide 2.30



Version 2.30 - November 2018

© Copyright 2018, WIBU-SYSTEMS AG,
Rueppurrer Strasse 52-54, 76137 Karlsruhe, Germany

Printed in Germany
All rights reserved. No part of this documentation, the accompanying software, or other components of the described product may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the personal use of the purchaser without the express written permission of Wibu-Systems.
While the data contained in this document has been written with all due care, Wibu-Systems does not warrant or assume responsibility or represent that the data is free from errors or omissions.

Wibu-Systems expressly reserves the right to change programs or this documentation without prior notice.

WIBU®, CodeMeter®, SmartShelter®, SmartBind® and BLURRY BOX® are registered trademarks of WIBU-SYSTEMS AG.
Microsoft®, Windows®, Win32® , WinCE® are registered trademarks of Microsoft Corporation.
macOS® is a registered trademark of Apple Inc.
VxWorks® is a registered trademark of Wind River Corporation.
Intel® is a registered trademark of Intel-Corporation.

All other brand names and product names used in this documentation are trade names, service marks, trademarks, or registered trademarks of their respective owners.

Wibu-Systems is member of:

PCMCIA since 1993

USB Implementers Forum since 1997

SD Card Association since 2007

Bitkom, German Association of Information Technology, Telecommunications, and New Media since 2003

VDMA, German Engineering Federation since 2008

OPC Foundation since 2012

and also a member of the developers programs of Autodesk, Apple, HP, IBM, Intel and Microsoft.

Microsoft Gold Certified Partner

Microsoft Embedded Partner

Strategic Software Partner Industrial and Medical

Table of Contents

I About this Guide 7

1 Safety Instructions 7

2 Installation 7

3 Shipped CmDongles 7

4 Additional Help Documentation 8

5 Typographical Conventions 9

6 Support by Wibu-Systems 9

7 About Wibu-Systems 10

II How to use this document? / CmEmbedded 2.30 11

III What is CodeMeter - The Basics 12

1 Software Protection and License Management 12

1.1 CmContainer Types *15*

1.1.1 CmDongle: CodeMeter Form Factors 15

1.1.2 CmActLicense: Binding and Activation 16

1.1.2.1 *CmActLicense Binding* 16

1.1.2.2 *CmActLicense Activation* 17

2 The CodeMeter Concept 17

2.1 Product Item Options - Custom-made License Entries *19*

2.1.1 Product Code 20

2.1.2 Text 21

2.1.3 License Quantity 21

2.1.4 Activation Time 22

2.1.5 Expiration Time 22

2.1.6 Usage Period 23

2.1.7 Unit Counter 23

2.1.8 Feature Map 24

2.1.9 Maintenance Period 25

2.1.10 Linger Time 25

2.1.11 Minimum Runtime Version 26

2.1.12 Named User 26

2.1.13 Customer Owned License Information (COLI) 27

2.1.14 User Data 27

2.1.15 Protected Data 27

2.1.16 Extended Protected Data 28

2.1.17 Hidden Data 28

2.1.18 Secret Data 28

2.1.19 Access Password 29

2.1.20 Maximum Encryption Rate 29

2.2 Allocation order of licenses *30*

2.3 Module Item *30*

2.4 Security with Capital S *30*

2.5 License Models - Mapping Variety using CodeMeter *31*

2.5.1 Implementing License Models 32

2.5.1.1 *Local Single User Licenses* 32

2.5.1.2 *Concurrent-/ Floating License in the Network* 32

2.5.1.3 *Demo Versions* 32

2.5.1.4 *Modular Licenses* 33

2.5.1.5 *Leasing* 33

2.5.1.6 *Pay-per-use Licenses* 33

2.5.1.7 *Downgrade/Version Managment* 34

2.5.1.8 *Overflow* 34

2.5.1.9 *Hot / Cold Standby* 34

2.5.1.10 *Named User Licenses* 35

2.5.1.11 *Machine-bound Licenses* ... 35
2.5.1.12 *License Borrowing* ... 35
2.6 Security by Encryption ... *35*
2.6.1 Key Derivation - One License Entry - Many Keys ... 35
2.7 Cryptography ... *37*
2.7.1 Direct and Indirect Encryption ... 37
2.7.2 Symmetric Enryption ... 38
2.7.2.1 *AES - Cipher Block Chaining Mode (CBC) (recommended)* ... 38
2.7.3 Asymmetric Encryption ... 38
2.7.3.1 *ECC - Elliptic Curve Cryptography* ... 38
2.7.3.2 *ECIES - Elliptic Curve Integrated Encryption Scheme* ... 38
2.7.3.3 *ECDSA - Elliptic Curve Digital Signature Algorithm* ... 38
2.7.3.4 *RSA* ... 39
2.7.4 Additional Encryption Algorithms ... 39
3 CodeMeter Start Center ... **39**
3.1 Structure and Navigation ... *39*
3.1.1 Menu Bar ... 40
IV CmEmbedded - CodeMeter for embedded devices **41**
1 Modules and Architecture ... **41**
1.1 Module Specifics ... *42*
2 Operating System Specifics ... **43**
2.1 VxWorks ... *43*
2.2 QNX ... *43*
3 CmDongles - Platform requirements and operating systems specifics ... **44**
3.1 Android ... *44*
3.2 Linux ... *44*
3.3 QNX ... *45*
3.4 VxWorks ... *45*
3.5 Windows ... *47*
3.6 macOS ... *47*
4 Delivery packages ... **47**
5 How to Integrate CmEmbedded (libraries) ... **48**
5.1 Linux ... *48*
5.2 Windows ... *48*
6 How to identify the CmEmbedded version ... **49**
V CmDongle on embedded devices **50**
1 Licensing: Programming of CmContainer ... **50**
1.1 CmBoxPgm ... *52*
1.1.1 Using CmBoxPgm ... 52
1.1.2 Basic Commands ... 52
1.1.3 CmContainer Options ... 53
1.1.4 Commandline Syntax ... 54
1.1.5 Firm Item Options ... 55
1.1.6 FSB Entry Options ... 56
1.1.7 Product Item Options ... 56
1.1.8 Special Commands ... 66
1.2 CodeMeter License Central ... *67*
1.3 Programming by File Transfer ... *69*
2 Protection: How to encrypt using AxProtector CmE ... **72**
2.1 Automatic Software Protection using AxProtector ... *72*
2.2 AxProtector Samples ... *73*
2.3 Commandline Options for AxProtector ... *73*
2.3.1 Basic Options ... 73
2.3.2 Options for the Licensing System ... 74
2.3.3 Options for Encrypting and Decrypting ... 76

2.3.4 Runtime Options....82
2.3.5 Operational Options....84
3 CodeMeter Core API: Licensed access to applications on the embedded device....84
3.1 Code Samples....85
3.1.1 List all connected CmDongles....85
3.1.2 Encrypting using CmCrypt2....85
3.1.3 Remote Programming....86
3.1.3.1 *Create a Context File*....86
3.1.3.2 *Execute a CmContainer Update*....86
3.1.4 Access a network license....87
VI CmActLicense on embedded devices 88
1 The Adapter - how to create an individual CmActLicense binding....89
1.1 Compatibility to previous CmEmbedded Versions....89
1.2 Adapter....89
1.3 Configuration....90
1.4 General....90
1.5 Error Codes....91
1.6 License Reader....91
1.7 License Writer....92
1.8 Binding Extension....94
1.9 Dynamic Data....95
1.10 Build....97
2 Create CmActLicense license file....98
3 Licensing: Programming of CmContainer....98
3.1 CmBoxPgm....100
3.1.1 Using CmBoxPgm....100
3.1.2 Basic Commands....100
3.1.3 CmActLicense Options....101
3.1.4 Commandline Syntax....107
3.1.5 Firm Item Options....108
3.1.6 FSB Entry Options....109
3.1.7 Product Item Options....109
3.1.8 Special Commands....119
3.2 CodeMeter License Central....120
3.3 Programming by File Transfer....122
4 Protection: How to encrypt using AxProtector CmE....125
4.1 Automatic Software Protection using AxProtector....125
4.2 AxProtector Samples....126
4.3 Commandline Options for AxProtector....126
4.3.1 Basic Options....126
4.3.2 Options for the Licensing System....127
4.3.3 Options for Encrypting and Decrypting....129
4.3.4 Runtime Options....135
4.3.5 Operational Options....137
5 CodeMeter Core API: Licensed access to applications on the embedded device....137
5.1 Code Samples....138
5.1.1 Encrypting using CmCrypt2....138
5.1.2 Remote Programming....138
5.1.2.1 *Create a Context File*....138
5.1.2.2 *Exeute a CmContainer Update*....139
5.1.2.3 *Import a CmActLicense file*....139
5.1.3 Access a network license....140
VII Appendix 141
1 How to build CmEmbedded from source....141
1.1 Building from Source Code....143
1.1.1 Building on Windows....144

1.1.2 Building on Linux 144
1.1.3 Building on macOS 146
1.2 CMake *146*
1.3 CMake Example *147*
1.4 Create example *149*
2 CmDongle and SPI (Serial Peripheral Interface) 149
2.1 SPI (Serial Peripheral Interface) Integration Version 2 ..... *149*
2.2 SPI (Serial Peripheral Interface) Protocol Specification Version 2 ..... *151*
VIII Glossary 152
IX Version 154
Index 195

1 About this Guide

This preface gives you an overview of the Guide's structure, holds references for the user of the *CodeMeter* Software Development Kit (SDK), informs on typographic conventions used, and helps you when contacting the support team of Wibu-Systems.

1.1 Safety Instructions

The hardware of WIBU-SYSTEMS AG serves to protect and license digital products and has been developed, manufactured and inspected in accordance with state-of-the-art technology and recognized technical safety rules and regulations.

For further information on hardware certificates see the respective documents to be downloaded at the website of Wibu-Systems (http://www.wibu.com/en/certificates.html).

Before you use the hardware please observe the following safety instructions:

- If you follow the instructions regarding safety as described in this manual, the hardware will, in the normal case, neither cause personal injury nor damage to machinery and equipment. Connect the hardware only to matching intended interfaces. The use for other purposes, opening or own repair of the hardware may lead to damages of the product and its surroundings. Modifying the hardware affects the product safety. Caution: risk of injury!
- The hardware may warm up during operation - which is a normal operational parameter.
- Keep the hardware away from humidity and avoid strong vibration, dust, heat, and direct sunlight, in order to prevent operational interference.
- Depending on the used operating system the detection of the hardware device may take some seconds. Before disconnecting the hardware the user should wait several seconds to avoid loss of data during data saving.
- This product is not a toy, keep away from children!

Non-compliance with the safety instructions results in a loss of warranty.

1.2 Installation

For installing *CodeMeter* on Windows operating systems ( ) please insert the shipped DVD into your DVD-ROM drive. The *CodeMeter* menu automatically opens.

For macOS ( ) the installer comes as a disk image file in subfolder MacOS (DMG_file). It installs the Developer Kit. Double-click the DMG-file to open the install '`package CmInstall.mpkg`' and the uninstaller '`CmUninstall.pgk`' in the Finder. Double-click on '`CmInstall.mpkg`' starts the installation. The uninstaller uninstalls all *CodeMeter* packages.

For Linux ( ) the file '`installation_dev_en.html`' in subfolder Linux contains all information needed for an installation on Linux.



If the DVD menu should not open, please start the file `start.exe` located in the root directory of the DVD.

After selecting the favored language click on the button **"CodeMeter SDK"**. Then follow the instructions of the installation assistant to install the *CodeMeter* SDK on your computer.

For installing *CodeMeter* on other operating systems, please find the respective files in the file cabinet.

1.3 Shipped CmDongles

Together with the *CodeMeter* Software Development Kit (SDK) you received two dongles, the *CodeMeter CmDongles*.

These dongles simultaneously act as 'leading' Master *CmDongles*, so-called *Firm Security Boxes* (FSB), which allow you to program other *CmContainer*. If you later decide to go live with *CodeMeter*, you will receive your own individual *Firm Code*. You also receive a `CmFirm.wbc` file. The following table shows the location of the `CmFirm.wbc` file for different operating systems.

Operating System	Location
Windows	`%ProgramData%\CodeMeter\DevKit`

© Copyright 2018 WIBU-SYSTEMS AG

Operating System	Location
macOS	`Library/Application Support/CodeMeter`
Linux	`etc/wibu/codemeter`

For Windows and macOS import the files via drag & drop into *CodeMeter Control Center*. For Linux operating systems use the command line tool *cmu*.

Then using this *Firm Security Box* as licensor you are able to transfer license information into other *CmContainer*. Here the tools or applications *CodeMeter License Editor*, *CmBoxPgm* 52 or *CodeMeter License Central* 67 are available (see Programming of CmContainer 50).

1.4 Additional Help Documentation

In addition to this Developer Guide on Windows operating systems:

Help File	Accessible va	
CodeMeter Developer Guide	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | CodeMeter Developer Guide]
		Press "Windows" key to open Start screen | Type "CodeMeter Developer Guide" | Press "Enter" key

the following help documentation is available.

> You also find the Developer Guide on the Installation DVD. The installation packages for macOS and Linux operating systems do not contain this file since specific tools and applications are available on Windows only, e.g. *Start Center*, *AxProtector* GUI, *License Editor*. However, the current Developer Guide is downloadable from the Wibu-Systems website (http://www.wibu.com/en/manuals-guides.html).

This documentation you find via respective calls in the tools and applications after installing the SDK (Software Development Kit)

Help File	Accessible by:	
CodeMeter User Help as HTML files including the parts *CodeMeter* Runtime Kit, *CodeMeter License Server*, *CodeMeter Control Center*, *cmu* commandline program, *CodeMeter WebAdmin*, Licensing - *Field-Activation-Service*, *CodeMeter* FAQ (German and English)	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | CodeMeter Developer Guide]
		Press "Windows" key to open Start screen | Type "CodeMeter Developer Guide" | Press "Enter" key
	respective menu items, buttons or **"Start | All Programs | CodeMeter | Documentation"** [*%ProgramFiles(x86)%\CodeMeter\Runtime\help\CmUserHelp*]	
AxProtector online help as compiled HTML help in German and English	Operating System	Menu navigation
	Windows	[Start | All Programs | AxProtector | Help]
		Press "Windows" key to open Start screen | Type "AxProtector Help" | Press "Enter" key
Software Protection API as compiled HTML help in English	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | Software Protection API Help]
		Press "Windows" key to open Start screen | Type "Software Protection API Help" | Press "Enter" key
Core API as compiled help file in English	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | Core API Help]
		Press "Windows" key to open Start screen | Type "Core API Help" | Press "Enter" key
CodeMeter Java-API as HTML files in English	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | CodeMeter Java API Help]
		Press "Windows" key to open Start screen | Type "CodeMeter Java API Help" | Press "Enter" key

© Copyright 2018 WIBU-SYSTEMS AG

Help File	Accessible by:	
Programming API as HTML files in English	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Documentation | Programming API (...)] for the respective programmin languages C++, Delphi, Java]
		Press "Windows" key to open Start screen | Type "Programming API (...)] for the respective programmin languages C++, Delphi, Java" | Press "Enter" key
Samples for programming of *CmContainer* and related Sample Help documentation	Operating System	Menu navigation
	Windows	[Start | All Programs | CodeMeter | Samples] [`%CodeMeter_Samples%\`
		Press "Windows" key to open Start screen | Type "CodeMeter Samples" | Press "Enter" key

1.5 Typographical Conventions

This manual uses the following semantic markups, text emphases, and symbols:

Format definition	Information type
Italics	*Product names*
Arial Narrow Italics	*Important terms*
Arial Narrow Italics	*Properties*
"Bold double quote"	**Objects you are able to select, such as, menus, buttons or drop down items**
"*Bold Arial Narrow*"	**Command names**
`CAPITAL LETTER COURIER NEW`	`KEYS, E.G. SHIFT, CTRL OR ALT.`
`Courier New`	`Path specifications, source code or file names`

Pictogram	Description
	This symbol refers to important and essential instructions you should follow.
	This symbol refers to additional information of general interest.
	This symbol refers to an example which explains a feature.

1.6 Support by Wibu-Systems

Our customers are supported by a professional team of exceptionally qualified staff. Our direct customer contact allows us to meet customer requests as fast as possible. A comprehensive FAQ list for the *CodeMeter* end user can be found at our *CodeMeter* support page and also information about *CodeMeter* and other additional products.

Enduser Support

Wibu-Systems provides a free-of-charge user hotline for your end customers.

Developer (Customer Support)

We are available in Germany (local Baden-Wuerttemberg non-holiday) workdays (Monday through Friday) from 8 a.m. to 5 p.m. per phone (+49-721-93172-14) or per e-mail (support@wibu.com). Wibu Systems USA support is available Monday through Friday from 8 a.m. to 5 p.m. PST by phone at 800-6-GO-WIBU (425-775-6900) or by e-mail (support@wibu.us). In China contact our Shanghai office per phone +86 (0) 21-55661790 or by e-mail (info@wibu.com.cn).

Support agreements with extended services on inquiry.

Many of our distributors also provide support. Please contact your distributor to see if this service is available to you and your customers locally.

Please state your customer number which helps us to deal with your request as fast as possible.

Support Information

For best handling of your request we need the following information:

- type of protection implementation (automatic / customized)

© Copyright 2018 WIBU-SYSTEMS AG

- operating system
- version of the *CodeMeter* software installed
- *CodeMeter* variant used
- detailed error description

1.7 About Wibu-Systems

WIBU-SYSTEMS AG was founded in 1989 by Oliver Winzenried and Marcellus Buchheit with a mission to provide state-of-the-art solutions for protecting and licensing software and digital media.

Products from Wibu-Systems support virtually all operating systems and come in a broad variety of form factors, including independency and the variety of form factors, including USB, PC Card, Express Card|34, Compact Flash Card, SD Card, MicroSD-Card, and ASIC. Applications include software for desktop PCs, servers, embedded systems, mobile, smart phones, and cloud computing.

Wibu-Systems is a privately-held corporation with a worldwide staff of 100, the majority in the headquarters facility in Karlsruhe, Germany. Subsidiaries are in Seattle (USA),Shanghai and Beijing (China), Tokyo (Japan) with sales offices as well in Belgium, Great Britain, the Netherlands, Portugal and Spain, and distributors in more than 25 countries. Corporate efforts stress achieving world-class quality in the areas of security, reliability, durability, support, and customer service.

More than 6,000 independent software vendors (ISV) rely on the *WibuKey* and *CodeMeter* technologies to sell more products by reducing piracy and increasing the flexibility of their licensing models. Products include:

- *CmDongle* the hardware-based variant of the protection and licensing technology *CodeMeter* is available in many form factors for a variety of interfaces and allows for multiple ISVs to share a single *CmDongle*, easy online license transfers, and optional Flash disk in different sizes.
- *CmActLicense* is a completely software-based variant of the protection and licensing technology *CodeMeter* that protects software by binding to the characteristics of an individual PC or any target system.
- *CodeMeter License Central* creates, managers, and delivers licenses with integration into sales and ERP systems
- *SmartShelter* creates, manages, and delivers licenses with integration into sales and ERP systems
- *SmartShelter SDL* (Secure Data Layer) protects data files including audio, video, and database
- *CodeMeter Identity*, an authentication solution allows for easy and safe access to websites and hosted software applications (SaaS).

Wibu-Systems is an active member of BITKOM, VDMA, SIIA, and participates with standards organizations such as PCMCIA, USB Implementers Forum, and the SD Card Association. Additionally, Wibu-Systems is a Microsoft Gold Certified Partner, Windows Embedded Partner, and partner in developer programs of Apple, Adobe, Autodesk, Wind River, and others. Products from Wibu-Systems have received multiple industry awards including the SIIA CODiE Award for "Best Digital Rights Management" solution and the international iF Product Design Award. The company is leading different research project with universities and other companies, in parts funded by the German BMBF and BMWi. Examples include MimoSecco with the aim of developing a flexible and secure middleware solution for third party applications in the area of cloud computing and OpenID/Card which is to allow managing virtual identities by identity provider on the basis of the new German electronic ID Card.

Contact Information		
Germany	+49 (0) 721-93172-0	sales@wibu.de
USA	+1.425.775.6900	info@wibu.us
China	Shanghai +86 (0) 21-55661790	info@wibu.com.cn
	Beijing +86 (0) 10-82961560/61	info@wibu.com.cn
Japan	Tokyo +81 (3) 3582-5385	info-jp@wibu.com
Great Britain, UK	+44 (0) 20 314 747 27	sales@wibu.co.uk
Ireland	+44 (0) 20 314 747 27	sales@wibu.co.uk
The Netherlands	+31 (0) 74 75 01 495	sales@wibu-systems.nl
France	+33 (0) 173030491	info@wibu.fr
Belgium	+32 (0) 3 400 03 14	sales@wibu.be
Spain	+34 (0) 91 414 8768	sales@wibu.es
Portugal	+34 (0) 91 414 8768	sales@wibu.es
Other countries	+49 (0) 721-93172-0	sales@wibu.com

© Copyright 2018 WIBU-SYSTEMS AG

2 How to use this document? / CmEmbedded 2.30

How to use this document?

CodeMeter is the technology of Wibu-Systems providing secure protection and effective license management of software and digital content. This *CmEmbedded Guide* divides into separate parts.

If you are new to *CodeMeter*, the section CodeMeter Basics 12 outlines the general concept of *CodeMeter* installed on your development system, and how it meets protection, licensing, and security requirements. Moreover, basic terms are introduced.

The section CmEmbedded - CodeMeter for Embedded Devices 41 gives an overview of *CmEmbedded* modules and architecture and comprises platform requirements and operating systems specifics, lists available delivery packages, and shows how to integrate *CmEmbedded* libraries.

Once you decided which licensing system, i.e. *CmDongle* or *CmActLicense,* you will employ, two separate sections with complete information will guide you how to do this.

CmDongle

If you employ *CmDongle* 50, the section describes how to program licenses and license options to the *CmDongle*, how to protect embedded applications, if supported, using *AxProtector* CmE, and how to use additional licensed accesses to your application via *CodeMeter Embedded Core* API (including code samples).

CmActLicense

If you employ *CmActLicense* 88, the section describes how to create the Adapter required for the individual *CmActLicense* binding, create a *CmActLicense* license file, program licenses and license options to the *CmActLicense*, how to protect embedded applications, if supported, using *AxProtector* CmE, and how to use additional licensed accesses to your application via *CodeMeter Embedded* Core API (including code samples).

The Appendix 141 explains how to build *CmEmbedded* from source using CMake and describes the communication between the ASIC on the *CmDongle* using SPI (Serial Peripheral Interface).

The Guide closes with a glossary and an index.

- *CodeMeter Embedded* for the productive use is delivered as static library.
  This library is always built holding all available *CmEmbedded* function modules and is able to handle all supported *CmContainer* accesses via *CmDongle* (File I/O, USB and SPI, Serial Peripheral Interface), *CmActLicense*, LAN and Runtime Bridge.

  By default, the source code package also holds all function modules. These can be deactivated, if required, allowing a modular customized configuration.
- *CmEmbedded* supports the new ***Universal Firm Codes*** for *CmDongle* and *CmActLicense* licensing systems.
- *CmActLicense* bindings are linked during the build process. User of the *CmActLicense* function no longer require the source code package to use *CmEmbedded*. Only the ***CmActAdapter*** is delivered as sample code, becomes adapted to the target hardware by the customer, and then is linked to *CmEmbedded*.
- For communication with the CmASIC - a dongle solution as chip - next to USB also SPI (Serial Peripheral Interface) communication is available.
- The performance of 'concurrent access' by several *CmEmbedded* simultaneous instances to a single system. These coordinate access to the *CmContainer* via shared memory thus provide consistent status (license cache).
- The communication between *CmEmbedded* and *CmDongle*, by default, is encrypted, The encryption is automatically performed without the necessity of any further action by the user.

  This function can be deactivated only, if *CmEmbedded* is built from source.

  
  Please note that a simultaneous mixed encrypted and -not-encrypted operation of CmEmbedded instances is not supported.
- The build system of the source code package has been switched to CMake.

For more detailed information see the separate Release Notes document available from the Wibu-Systems website after registration.

© Copyright 2018 WIBU-SYSTEMS AG

3 What is CodeMeter - The Basics

This section introduces *CodeMeter* as secure hardware and software-based software protection and licensing technology for digital contents 12 . Also the the CodeMeter Concept 17 is explained, and describes the CodeMeter Start Center 39 as the communication center in the developing environment.

3.1 Software Protection and License Management

With *CodeMeter* Wibu-Systems offers a secure hardware and software-based software protection and licensing technology for digital contents for smartphones, embedded systems, desktop PCs, server and cloud computing.



In the following parts of the document the term *CmDongle* will be used representing all *CodeMeter* hardware form factors. *CmActLicense* represents the pure software and activation based variant of the protection and licensing system *CodeMeter*. If there is a technical reference to both variants, the term *CmContainer* is used.
Moreover, throughout this document the terms "licensor" and "licensee" are used. The term "licensor" may be replaced by "developer" or "vendor", while "licensee" refers to the software "enduser" or a user of digital content.

The protection effect is accomplished by the fact that a *CodeMeter* protected software functions only with the corresponding copy protection hardware (*CmDongle*) or the software and activation based variant *CmActLicense*. *CmDongle* is available as USB version (*CmStick/M /ME /I /T /C*), as PC Card (*CmCard/M,* Cardbus, 32 Bit), as Express Card|34 (*CmCard/E*), as Compact Flash Card (*CmCard/CF*), as SD and MicroSD-Card, and as ASIC.

Protection of Copyrights and License Rights

In a user-friendly way, *CodeMeter* technically safeguards the compliance with copyrights. In doing so, *CodeMeter* presents a technology which provides software protection by hard encryption but simultaneously also allows for the secure mapping of licensing strategies. The protection is based on encryption and decryption operations which are securely performed inside the *CmContainer*.

You integrate this protection into your software once; using effective tools and interfaces, and then deliver the same program customized to your customers or to various license models. Subsequently, the software runs only with the correspondingly programmed *CmContainer*. What our competitors today call "Protect Once, Deliver Many™" Wibu-Systems has been offering as a matter of course since the company was founded in 1989.

As the figure below shows, *CodeMeter* meets all requirements for a secure and effective technology in the realm of software protection and license management.

© Copyright 2018 WIBU-SYSTEMS AG

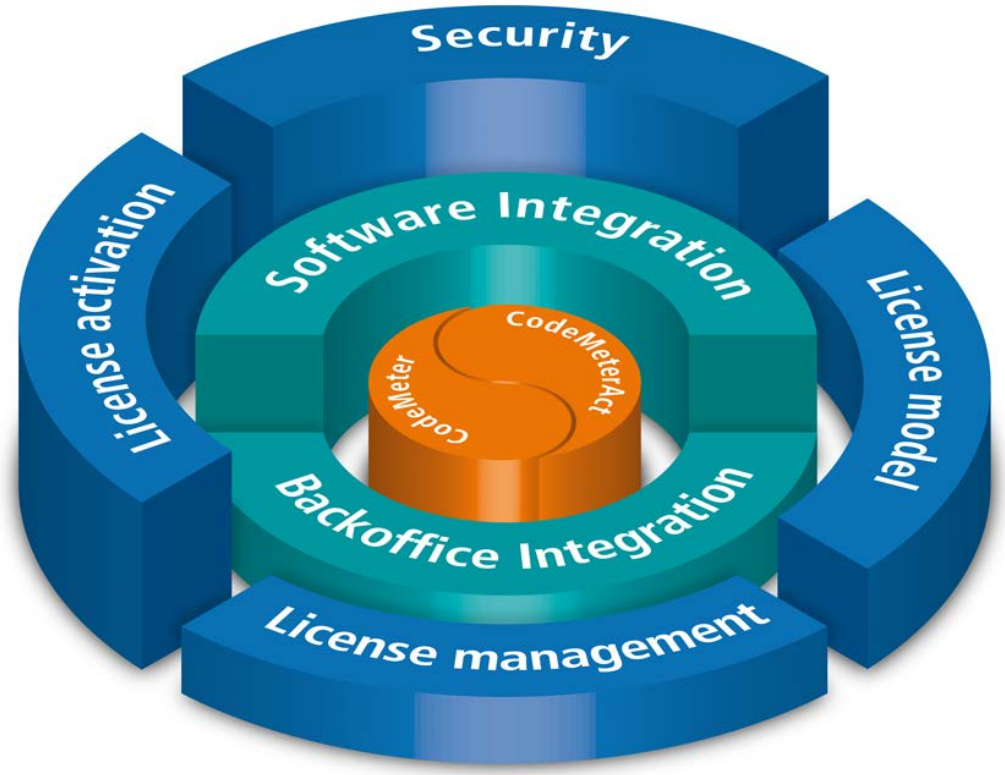


Figure 2: Overview - Software Protection and License Management using *CodeMeter*

Security

- For protection *CodeMeter* uses state-of-the-art encryption algorithms including AES (Advanced Encryption Standard*)* bit key length and ECC (Elliptic Curve Cryptography) with 224 bit key length for asymmetric encryption and signatures, and RSA with 2048 bit key length for asymmetric encryption.
  For ECC Wibu-Systems only supports the P-224 curve variant secp224r1 with a key length of 224 bit as recommended by the U.S. American NIST (National Institute of Standards and Technology).
- All keys used are safely stored in the *CmContainer.* The recipient is not able to read out the keys from the *CmContainer.* In addition, the option of using alternating keys exists, i.e. at runtime of the application further information is integrated into encryption and decryption operations. The keys may also be randomly generated within the *CmContainer.*
- A secure leading dongle, the *Firm Security Box* (FSB) allows programming of licenses into the *CmContainer.* The FSB is unique for each licensor.
- In Hacker's contests, software protected by Wibu-Systems has successfully met the challenges of the international hacker's scene.
- The *CmDongle* is additionally protected against all known analytical hacking methods (e.g. electron beam microscope, DPA) and the communication between *CmContainer* and the PC is completely encrypted.
- Parts of the protected application (source code and resources) are decrypted only, if accessed. This "on demand decryption" effectively protects against memory dumping and the extraction of unprotected versions.
- *CodeMeter* provides multi-layered, combinable and interconnected protection:
  - Automatic protection of applications using *AxProtector* as secure basic protection without changing the source code including runtime checks, effective anti-debug mechanisms, modification of resources, and locking of the *CmContainer* if crack attempts are detected.
- Additional technical sophisticated security mechanisms integrated in *CodeMeter* technologies, tools, and interfaces which are constantly developed and advanced
- Manipulation-proved protection of usage periods, activation and expiration times of applications by using the *CmContainer* internal clock and a certified time stamp mechanism.

License Mapping

- Programming of license entries into the *CmContainer* with a variety of options:
  - tag licenses with describing information

© Copyright 2018 WIBU-SYSTEMS AG

- define the number of simultaneous users and network access models using built-in network support (LAN and WAN)
- implement activation and expiration times of a license with relative or absolute dates, or a usage period with a variable start time
- create and display user-specific information
- program independent counter to be decremented for defined actions
- use a *Feature Map* to release single modules of an application while only a single license entry is allocated, or to manage versions
- use maintenance periods to grant software support and service for defined time periods
- use additional binary information via diverse data fields also to locate alternative key sources

➲ Variable combinations of license options make up for mapping any imaginable license strategy:

License strategy	License model
Standard License Models	Single User License Floating | Concurrent Licenses Demo Versions Modular Licenses
Feature-based Licenses Models	Leasing Software Assurance Pay-per …
Extended License Management	Downgrade | Version Management Overflow Licenses Cold | Hot Standby Licenses Named User Licenses Machine bound Licenses License Borrowing Volume Licenses

➲ The *CodeMeter* SmartCard Chip with 60/384 kByte memory allows the programming of up to 6,000 license entries into a single *CmDongle*.

➲ Vendor-independent use and management of license entries by unique and secure separation of individual license container in a *CmDongle.* Thus several software vendors are able to share a single *CmDongle.*

Licensing Management

➲ Efficient ticket system *CodeMeter License Central* in a *Desktop* and *Internet* edition. The input of order, customer and item number creates matching tickets to be used for further tasks in the sales and production departments.

➲ Integration of license management in sales and support processes by *CodeMeter License Central Internet* including interfaces: Internet gateway to the customer, connectors to ERP and CRM systems, and connectors to online shops.

➲ Data transfer via SOAP (XML-based) including only minimal customization in the online shop or the ERP system. In most cases, existing license generators and customer-specific order fields are instantly transferable.

License Activation

➲ Next to local programming, also secure programming, editing or deleting of complete license contents and options in a *CmContainer* via file transfer.

➲ File-based remote programming using CmFAS *(CodeMeter Field Activation Service)* or SOAP-based using *CodeMeter License Central.*

Software Integration

➲ Automatic integration of the protection into the software as basic protection via automatic encryption of executable source code without changing the source code using *AxProtector.*

- easy-to-use graphical interface including the most important options for the encryption of different project types (Windows 32-bit/64-bit,, Android Native, Linux).
- creation and further use of a commandline for *AxProtector* commandline.

➲ Additional requirements (encryption and decryption of data, personalization, read-out additional data) are met by *CodeMeter Core API* holding extensive functions. Using the interactive *CodeMeter API Guide* quickly provides you with the matching source code.

Back Office Integration

➲ Easy and fast creation and programming of licenses when developing a software, or testing license strategies using the graphical *CodeMeter License Editor* interface if only a small number of *CmDongles* is in use.

© Copyright 2018 WIBU-SYSTEMS AG

- Commandline programming applying scripts and batch files for mass production and test automation using *CmBoxPgm*. Process programming is simultaneously applied in one pass to several *CmContainer.*
- Create, manage and deliver licenses with the efficient ticket system *CodeMeter License Central* in a *Desktop* and *Internet* edition.
- Additional requirements not met by the existing tools to create, program, and manage licenses can be integrated into own applications using the basic *Programming API* (HIP, *High Level Programming API*).

3.1.1 CmContainer Types

3.1.1.1 CmDongle: CodeMeter Form Factors

CmDongle is available in a large variety for different interfaces:

Form factor	Description	
	CmStick	Standard Edition for the USB interface plastic case without additional flash memory[1)]
	CmStick ME	Metal Edition, in a classy metal case without additional flash memory[1)]
	CmStick/M	Version of both editions with additional flash memory to directly start the software mobile from the *CmDongle*
	CmStick/T	Version of both editions with internal battery without additional flash memory[1)]
	CmStick/C	Compact-robust small edition without additional flash memory[1)]
	CmStick/I	USB Flash Disk Module with with a 2x5 socket of 2.54 mm standard grid size
	CmStick/CI	USB Flash Disk Module with a 2x4 socket of 2.00 mm grid size
	CmCard	PC Card, 32-bit, with Flash Memory
	CmCard/E	CmCard as Express Card with 34 standard interface
	CmCard/CF	CF Card (Compact Flash) with Flash Memory
	CmCard/SD	Secure Digital Memory Card
	CmCard/Micro SD	Micro Secure Digital Memory Card
	CmCard/CFast	Industrial CFast Memory Card (2, 4, 8 ,and 16 GB)
	CmASIC	ASIC for integration in own hardware

Figure 3: *CmDongle* Form Factors

[1)] This form factor can alternatively be configured as Human Interface Device (HID). For requirements and details see here.

© Copyright 2018 WIBU-SYSTEMS AG

3.1.1.2 CmActLicense: Binding and Activation

CmActLicense represents the software-based variant of the protection and licensing technology *CodeMeter*. Here licenses and the keys responsible for encrypting and decrypting are saved to a *CmActLicense* license file which is cryptographically safeguarded and signed. This virtual *CmContainer* is unique and bound only to a specific computer or device.

The unique binding is guaranteed by a digital "finger print" calculated from specific hardware features of a computer or a device. This ensures that *CmActLicense* licenses are valid only for the identified computer or device and are not transferable.

3.1.1.21 CmActLicense Binding

Binding Schemes

Structuring which hardware features are used in which way for binding a license is done by using binding schemes. These schemes are divided in three categories: dynamically weighted using *CodeMeter SmartBind*, explicitly using *Binding Extension* and without binding using the *None* binding scheme.

***CodeMeter* SmartBind**

SmartBind is only available with *CodeMeter Runtime* and not available with *CodeMeter Embedded*.

***CodeMeter* Binding Extension**

In cases in which the binding of licenses is to be designed to be bound to vendor-specific features of a device or own secure features of a separate target system - for example in the embedded field - the binding scheme Binding Extension[102] is available.

When using these hardware features the vendor together with the installation program of his software additionally delivers a signed plugin. *CodeMeter License Server* on demand loads this plugin and provides functionality to detect the features. This way all imaginable features may be used a binding features for *CmActLicense* licenses, e.g. of a end-user computer or of a embedded target system.

For more information see the separate document "CmActLicense Binding Extension" you get from Wibu-Systems on request.

If you use the binding scheme *Binding Extension* for individual binding of a *CmActLicense* to an own hardware you are able to create and deliver pre-calculated license[101] files when the binding value is known. The step to create a license request file on the target system then is only optional at a later activation.

None Binding

Using the binding scheme None[102] allows you to deliver protected software without the binding to a specific computer or device.

This is the case, for example, it the binding of a license is time-limited but is to be valid for any computer or device, e.g. for test and demo reasons. Here Wibu-Systems offers the "Trial License[105]" license model allowing you create demo licenses which are valid for a maximum of 90 days. These licenses expire after this period and are not re-importable.

An additionally use case is creating time-unlimited and re-importable licenses for any computer or device. This is relevant, if primarily preventing reverse engineering is wanted. Here Wibu-Systems offers the "Protection Only[106]" license model.

For both 'None-Bind' based license models a separate license entry in the *Firm Security Box*[19] (FSB) is required and as evaluation *Firm Code* `6000010` is part of the Software Development Kit.

Additional Options for *CmActLicense* licenses

In addition to the binding schemes, you are also able to set further options when activating *CmActLicense* licenses. The following table lists these options.:

Option	Description
Operating Systems	This option allows you to define the operating system(s) on which *CmActLicense* license can be used.
Virtual Machines	This option allows you to enable the use of *CmActLicense* licenses on virtual machines.
Multiple License Reimport	This option allows you to define that a *CmActLicense* activation file is unlimited re-importable on a computer or device.
CodeMeter Runtime	This option allows you to set a minimum required *CodeMeter* Runtime version.

© Copyright 2018 WIBU-SYSTEMS AG

3.1.1.22 CmActLicense Activation

Largely, activating *CmActLicense* licenses is based on the standardized *CodeMeter* procedure for file-based remote programming of *CmDongles*[69]. The procedure is based on the transfer of license request and license update files.

License request files (context files) hold the current license information status at the customer and license update files are used by the vendor to provide updates and activations.

However, in the case of *CmActLicense* license before activating licenses first the actual hardware features of a computer or a device have to be detected. Here the vendor creates a license information file (`*.WibuCmLIF`) for *CmActLicense* using *Universal Firm Codes* (UFC) or (`*.wbb`) file for *CmActLicense Firm Codes*. This file corresponds to an empty license container however holds specifications on binding schemes[16] and additional activation options[16] to be used for unique binding of a license to the computer or the device.

By importing the empty license container by the customer two things happen. Firstly, the necessary information on the computer or the device are detected and, secondly, the basis for binding the license using a unique, digital "finger print" is prepared. The initial license request file the customer creates then holds all necessary license information the vendor needs to program a *CmActLicense*-license which is uniquely bound to this computer or device and can only be activated for this computer or device. The transfer of these binding and activation information is provided by the license update file the customer imports.

The following figure illustrates this process:

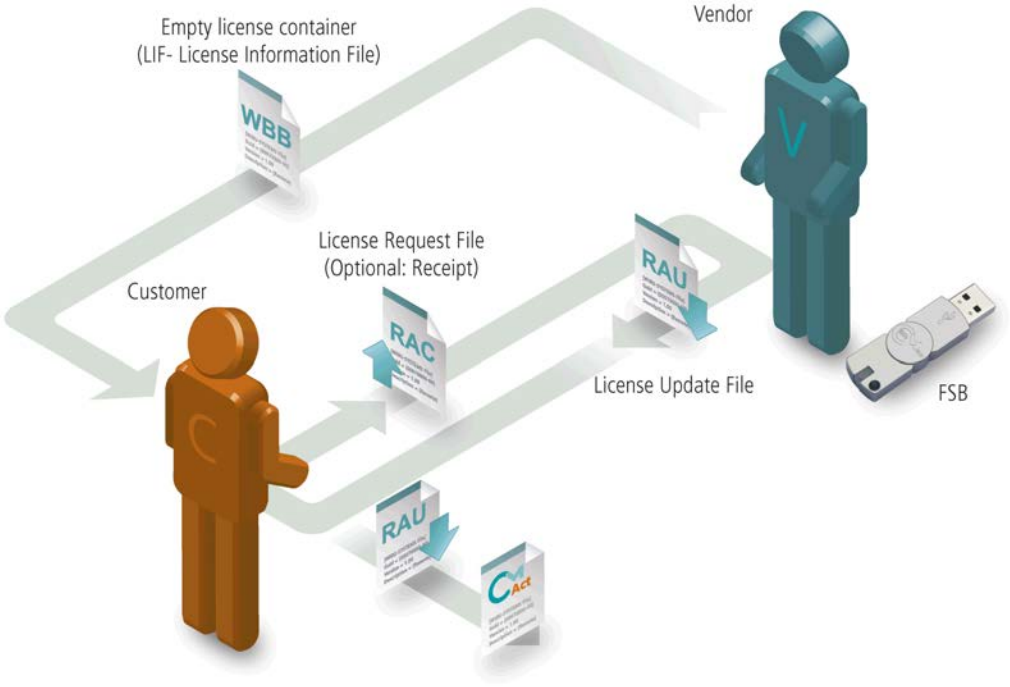


Figure 4: *CmActLicense* - Activation by file-based remote programming (CmFAS, *CodeMeter Field Activation Service*)

3.2 The CodeMeter Concept

In *CodeMeter* a license is identified by two unique numbers: *Firm Code* and *Product Code*.

The *Firm Code* you receive from Wibu-Systems. This number individually identifies each licensor and is uniquely one-time assigned.

The *Product Code* is a number you are free to choose. This allows you to identify products you want to protect and license.

> If you want to protect and license more than one product, you can use a *Product Code* for each single product. Comprehensive products can also have several *Product Codes* at the same time, e.g. programs with a variety of modules.

Analog to a file cabinet, the entries in a *CmContainer* are hierarchically structured in several logical areas.

© Copyright 2018 WIBU-SYSTEMS AG

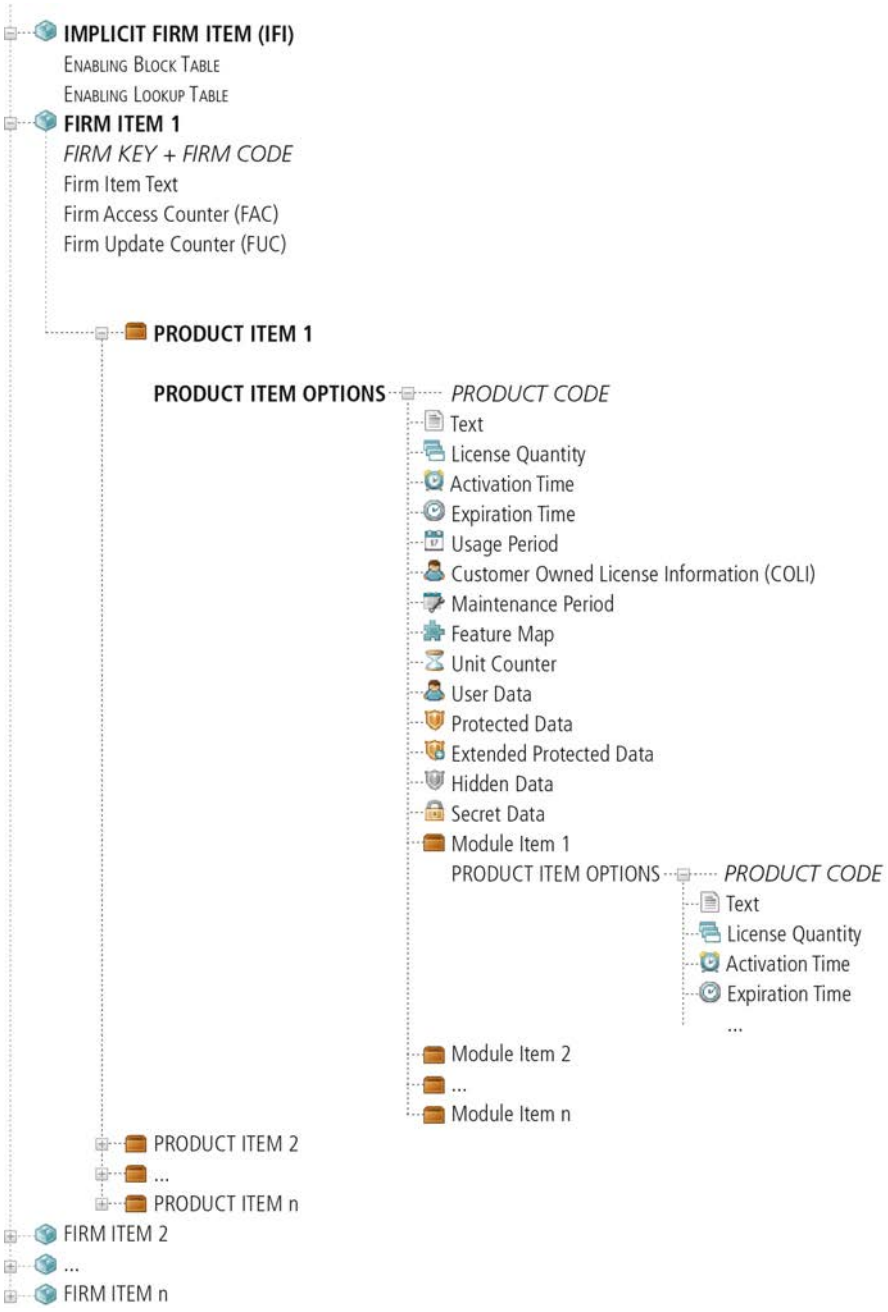


Figure 5: *CmContainer* License Entry Organization

At the top level, you find the *Firm Items*. *Firm Items* represent license container which separately hold the *Firm Code* for each single licensor.

Firm Item Options (FIO)

Further options - the *Firm Item Options* (FIO) - label each license container, and count how often it has been addressed by an update or an access (Firm Item Text, Firm Update Counter, Firm Access Counter).

Each licensor owns a separate individual license container and only s/he is able to create, edit or delete license entries for products for his/her *Firm Item*.



This is the reason why licenses in a *CmDongle* can be organized vendor-independent. Several vendors may share a *CmDongle* and save costs and efforts. The licensee has the advantage that s/he has all his/her licenses available in a single *CmDongle* using only one port. A *CmDongle* may hold up to 6,000 license entries.

Implicit Firm Item (IFI)

The *Implicit Firm Item* at the *Firm Item* level is a special license container. This logical area of the entry structure is freely accessible for each *CmContainer* owner. The only prerequisite here is that s/he has a valid password for accessing the *CmContainer*.

© Copyright 2018 WIBU-SYSTEMS AG

Product Items, the License Entries

The license entries for the actual products locate at the level of the *Product Items*. The *Firm Item* level can hold one or more license entries, i.e. *Product Items*.

At the *Product Item* level of single license entries also the *Product Item Options* locate. They hold the *Product Code* which uniquely defines a license entry. And also further options defining the actual characteristics of a license, such as, how many licenses may be simultaneously used on a network, how long a license is valid, which functions are accessible and billed, etc. Moreover, several other data fields are available holding additional binary information and differ in their access privileges (for an overview and the description see here) 19.

These optional characteristics are combinable in a variety of ways and constitute the basis for the mapping of any imaginable license strategy (see license models 31).

It is imperative for the entry structure of a *CmContainer* that a *Firm Item* level can be created only with a uniquely identified *Firm Code* and that license entries can not be created, edited or deleted outside this license container. This is ensured by a *Firm Security Box* (FSB) which is bound to your *Firm Code*. The FSB and the *Firm Code* are issued to you by Wibu-Systems.

Firm Security Box

The *Firm Security Box* (FSB) represents a form of a Master-*CmContainer* required to program licenses with your *Firm Code* into a *CmContainer*.

This way Wibu-Systems ensures that only you as the owner of the *Firm Security Box* are able to program other *CmContainer* using your *Firm Code*. The programming process is safeguarded by cryptography and the required keys are safely stored in your FSB.

Firm Key for *CmDongle*

And finally, in the case of the hardware-based *CodeMeter* variant *CmDongle*, Wibu-Systems assigns you a *Firm Key*. The *Firm Key* is a secret key and influences almost all encryption and decryption operations of licenses, their authentication, and also the creation, update and deletion of license entries at the *Product Items* level. The *Firm Key* is initially delivered in and with your *Firm Security Box*. However, if you feel a higher security need, and want to define the *Firm Key* in the case of the hardware-based *CodeMeter* variant *CmDongle* for yourself, you are free to do so.



When using an individual *Firm Key* you must ensure that you very safely store this *Firm Key*. If you loose this key, even Wibu-Systems is not able to restore it.

The *Firm Key* is stored in the *Firm Security Box* (FSB) in the *Product Item Option* (PIO) *Secret Data*.



For security reasons, you are not able to retrieve the *Firm Key* from the *Firm Security Box* (FSB).

3.2.1 Product Item Options - Custom-made License Entries

Each license entry at the *Product Items* level can hold differently combined *Product Item Options* (PIO). These PIOs allow you to define individual license models for separate customers.

This is an important feature that can save the developer lots of time and money. Why? Because the developer no longer needs to spend time altering installations on a customer by customer basis. Instead, all customers receive the same software and the license options are defined in the *CmContainer*. See the table below for the properties of the single options:

Product Item Option	Description	Read Access	Write Access	Integrated in Encryption
Text	256 double byte character, used for display in *CodeMeter WebAdmin*	√	√	X
License Quantity	Number of simultaneously usable licenses, use for floating | concurrent licenses on the network	√	with FSB	X
Activation Time	Use for time-limited versions	√	with FSB	√
Expiration Time	Use for time-limited versions	√	with FSB	√
Usage Period	Use for time-limited versions	√	initially at first start	√
Customer Owned License Information	128 character, use for customer-specific data (e.g. name of the licensee)	√	with FSB	X
Feature Map	32-bit mask, use for activating features or for version management	√	with FSB	√

© Copyright 2018 WIBU-SYSTEMS AG

Product Item Option	Description	Read Access	Write Access	Integrated in Encryption
Maintenance Period	Used for time-limited software service agreements	√	with FSB	√
Linger Time	Used for controlling time on re-start	√	with FSB	x
Named User	Using required credentials for license access	√	with FSB	x
Minimum Runtime Version	Minimum CodeMeter Runtime Version (Major- + Minorversion)	√	with FSB	x
Unit Counter	Counter, use for pay-per-use, pay-per-click, pay-per-print, or pay-per-start versions	√	reducing √ / incrementing with FSB	√
User Data	256 byte data, use for saving configuration data	√	√	x
Protected Data	256 byte for saving additional data	√	with FSB	x
Extended Protected Data	128 types of up to 256 bytes[1)]	√	with FSB	x
Hidden Data	128 types of up to 256 bytes data, use as key source[1) 2)]	with password	with FSB	as separate key
Secret Data	128 types of up to 256 bytes data, use as key source[1)]	x	with FSB	as separate key
Access Password	String to access a *Product Item*	√	with FSB	x
Maximum Encryption Rate	Integer	√	with FSB	x
Module Item	A *Product Item* may hold 0 to n *Module Items*	√	with FSB	x
Universal Data	Saving of keys: AES key length 256 bit; RSA key length 4096 bit	x	with FSB	as separate key

[1)] 128 types (index 0-127) available for the ISV while indices 128-255 are reserved for Wibu-Systems

[2)] The reading and writing of data from or into a *CmContainer* is featured also without FSB access at runtime using WUPI functions, if the *CmContainer* is specially prepared.

Table 2: Overview *Product Item Options* (PIO)

In order to modify these license options, in most cases your *Firm Security Box* is required. This way, Wibu-Systems ensures that your customer is not able to change the license you sold. Only the options *Text* and *User Data* can be modified without a *Firm Security Box*. At the same time, with *CodeMeter* tools and interfaces used to check and query licenses guarantee that your software is used with the proper license information.

3.2.1.1 Product Code

The PIO *Product Code* serves as the unique identification of a license entry.

- The *Product Code* is a 32 bit value and can freely chosen by the licensor.
- The definition and programming of the *Product Code* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter* tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the Product Codes
CmBoxPgm	Programming the PIO /p 57
CodeMeter License Central	Programming the PIO
Query/Check	
AxProtector	*Product Code* has to be defined.
Core API	***CmAccess*** and handle editing using ***CmCrypt***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.1.2 Text

The PIO *Text* serves for labeling a license entry.

- The *Text* option may hold up to 256 double byte of information, e.g. name of the product or user as displayed in *CodeMeter WebAdmin*.
- Write and read access is not limited i.e. a *Firm Security Box* (FSB) is not required.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO *Text*
CmBoxPgm	Programming the *PIO* /pt 63
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetProductItemText* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.3 License Quantity

The PIO *License Quantity* serves to define single user licenses or the number of simultaneously used licenses on a network. With its use you can implement different license models, such as, single user, concurrent / floating licenses, or terminal server sessions.

At the same time, you have to define access modes to organize license allocation, i.e. how do started instances and allocated licenses of the protected software correspond to each other in a network environment.


These modes are not saved in the *CmContainer* but you define them when encrypting your software.

- The License Quantity option may hold up to 4 bytes and holds the information of the number of licenses available on a network.


Up to *CodeMeter* Version 6.0 setting the option to a value of 0 created a local only license.


Please note that local licenses on server operating systems (termin al server) cannot be used via remote desktop connections.

Since *CodeMeter* Version 5.10, the programming of a license with a License Quantity value of 0 in the case of a not installed server operating system (Windows Server 2008, etc.) allows the start of the application via remote desktop connection.

However, if the computer has been assigned a License Quantity value of 0 and an installed server operating system, the start of the application via remote desktop connection is not supported.

For example, if you have installed Windows 7, Windows 8, etc., the access via remote desktop connection works but not, for example, with a Windows Server 2008 system.

- The definition and programming of the *License Quantity* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO License Quantity
CmBoxPgm	Programming the *PIO* /plq 59
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetAbsoluteLicenseQuantity* 51 or *SetRelativeLicenseQuantity* 51
Query/Check	
AxProtector	License handling and License Options

© Copyright 2018 WIBU-SYSTEMS AG

Query/Check	
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.4 Activation Time

The PIO *Activation Time* serves as an activation date in terms of a "valid from…" to implement license models which define a start time of a protected application.

If you additionally define an *Expiration Time* [22], you are able to implement time-limited license models, e.g. leasing, subscription, etc.

- An Activation Time defines a split second value in intervals between `January 1st, 2000, 0:00:00` and `December 31st, 2099, 23:59:59`. This value is always saved in the time zone format UTC (*Universal Time Coordinated*) and is independent of a time zone or a daylight savings time interval.
- Access to the license is granted only if the Box Time and the *Certified Time* in the *CmContainer* are later than the defined Activation Time. For the fail safe and manipulation safe control mechanism see here.
- The *Activation Time* is part of the key derivation [35]. This key is derived each time an encryption, decryption or authentication operation is involved. A manipulation of the *Activation Time* which is not permitted, e.g. setting an earlier date, leads to deviant derivation results and the licensed access is prevented.
- The licensee is not able to directly change the *Activation Time*, i.e. setting it to an earlier date.
- The definition and programming of the *Activation Time* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.
- The licensor may set an *Activation Time* as either an absolute value, or a relative value. E.g. start on `January 1st, 2010` or start 30 days from initial installation.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO Activation Time
CmBoxPgm	Programming the *PIO* ***/pat*** [57]
CodeMeter License Central	Programming the number of days the application allowed to run from the first start
Programming API	Call class ProductItemParamSet [51] and subsequently SetAbsoluteActivationTime [51] or SetRelativeActivationTime [51]
Query/Check	
AxProtector	Select options in advanced runtime settings
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.5 Expiration Time

The PIO *Expiration Time* serves as an expiration date in terms of a "valid until…" to implement license models which define an end time of a protected application.

If you additionally define an *Activation Time* [22], you are able to implement time-limited license models, e.g. leasing, subscription, etc.

- An *Expiration Time* defines a split second value in intervals between `January 1st, 2000, 0:00:00` and `December 31st, 2099, 23:59:59`. This value is always saved in the time zone format UTC (*Universal Time Coordinated*) and is independent of a time zone or a daylight savings time interval.
- Access to the license is granted only when the *Box Time* and the *Certified Time* in the *CmContainer* precede the defined *Expiration Time*. For the fail safe and manipulation safe control mechanism see here.
- The *Expiration Time* is part of the key derivation [35]. This key is derived each time an encryption, decryption or authentication operation is involved. A manipulation of the *Expiration Time* which is not permitted, e.g. setting an earlier date, leads to deviant derivation results and the licensed access is prevented.
- The licensee is not able to directly change the *Expiration Time*, i.e. setting it to a later date.
- The definition and programming of the *Expiration Time* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.
- The licensor may set an *Expiration Time* as either an absolute value, or a relative value; e.g. stop on `January 1st, 2010` or stop 30 days from first access.

© Copyright 2018 WIBU-SYSTEMS AG

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO *Expiration Time*
CmBoxPgm	Programming the *PIO* /pet 58
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetAbsoluteExpirationTime* 51 or *SetRelativeExpirationTime* 51
Query/Check	
AxProtector	Selecting options in advanced runtime settings.
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.6 Usage Period

The PIO *Usage Period*, defined as a fixed period of time a license can be used, allows to implement license models not bound to a fixed start time. This allows implementing 'real' demo versions.

3.2.1.7 Unit Counter

The PIO *Unit Counter* serves for implementing license models which bill a software according to its actual use, e.g. pay-per-use, pay-per-click, etc.

You define an initial value of the counter and which software action is to decrement this counter by how many units. Actions may comprise, for example, the number of calls of specific software functions, number of print jobs, etc. At the same time, you may also use an *Unit Counter* for time-limiting a license by checking the software by a fixed interval and decrementing the counter on each check by a defined value.

- An *Unit Counter* may assume integer values between `0` and `4294967294` (Hex: `FFFFFFFE` (32 bits). Up to Firmware Version 1.18 integer values between `0` and `16777215` (24 bis) are valid.
- The number of units by which the *Unit Counter* is decremented (delta value) may be set by you as value between `1` and `9999`. The decrement takes place safe from manipulation inside the *CmContainer*.

  For security reasons the user of an application decreases this value of an Unit Counter. Increasing this value is not possible.

- The *Unit Counter* is part of the key derivation 35. This key is derived each time an encryption, decryption or authentication operation is involved. A not permissible manipulation of the *Unit Counters*, e.g. increasing the counter or decreasing the delta value, leads to deviant derivation results and the licensed access is prevented.
- If the *Unit Counter* reaches a value of `0`, the access is prevented. Only special operations which ignore an *Unit Counter* are still executable.
- The licensor may set an *Unit Counter* to an absolute value or add a value to an existing one (relative).

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO *Unit Counter*
CmBoxPgm	Programming the *PIO* /64puc 64
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetAbsoluteUnitCounter* 51 or *SetRelativeUnitCounter* 51
Query/Check	
AxProtector	If an *Unit Counter* exists in the *CmContainer*, *AxProtector* automatically will decrement it when the software is started. However, you are able to change the decrement. On the *AxProtector* page "Runtime settings" you are able to check the license also at runtime using an existing *Unit Counter*.
Core API	***CmCrypt***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.1.8 Feature Map

The PIO *Feature Map* serves for implementing license models which activate specific functions (modules, features) or versions of an application.

If you do not want to use individual *Product Items* for different modules of a program, you are able to assign a *Feature Code* within a *Product Item*. The *Feature Map* represents 32 bits allowing you to individually assign and activate up to 32 *Feature Codes*.

Using the *Feature Map* is also an option to manage versions. Here an individual *Feature Code* is assigned to each program version.

Version Management using *Feature Code*

Each new major version is coded as one bit. If your customer is allowed to use several versions, then activate the corresponding bits by setting the bits to a value of 1.

In combination with the PIO *License Quantity* you are now able to implement a downgrade privilege in the network. Up to the defined number of licenses your customer is able to use the current version, or the activated previous versions. However, in total no more than the number of licenses you defined in the PIO *License Quantity*.

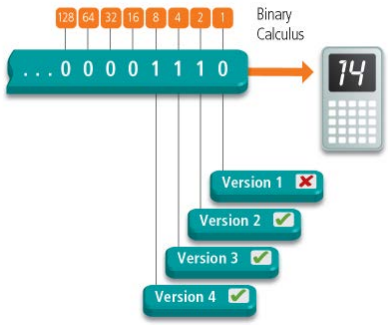


Figure 6: Version Management using PIO *Feature Code*

As the figure shows, the binary value of "`1110`" or the decimal value of `14` activates versions 2 through 4 but not version 1. In *AxProtector* and *IxProtector* you are able to specify the *Feature Codes*.

> Even if you do not use the *Feature Map* for implementing license models, the value of the *Feature Map*, the *Feature Code* does become part of the key derivation 35 and thus of encryption and decryption operations. Then the *Feature Map* has a *Feature Code* of 0 and the *Product Item Option* is not activated.

The PIO *Feature Map* has the following properties:

- Up to 32 features are independently manageable. Each feature is mapped by an individual single bit.
- The Feature Code is part of the key derivation 35. This key is derived each time an encryption, decryption or authentication operation is involved. A not permissible manipulation of the Feature Codes, e.g. setting a corresponding bit in the *Feature Map*, leads to deviant derivation results, and the licensed access is prevented.
- The licensee is not able to directly change the Feature Code, i.e. adding new features and activate them.
- The definition and programming of the *Feature Map* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the PIO *Feature Map*
CmBoxPgm	Programming the *PIO* /pfm 59
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetFeatureMap* 51
Query/Check	
AxProtector	Must be defined
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.1.9 Maintenance Period

The PIO *Maintenance Period* serves to store a time-span into the *CmContainer*, e.g. `12.01.2011` until `03.31.2012`. Then the use of a license is limited to software versions which have been created, i.e. released, within this *Maintenance Period*. The *Release Date* is stored in the protected application and at runtime a check is executed whether the date is within the defined period. An option exist to choose between a required *Maintenance Period* and a check of the *Release Date* only if a *Maintenance Period* exists (default setting). If the *Release Date* does not locate within the *Maintenance Period* the use of the software is not covered by the license.


Requires *CodeMeter*® Firmware 1.18 or higher.

This allows you to implement license models which map the granting of support and services when using the software.

- The *Maintenance Period* option holds two 32-bit values: start and end of the *Maintenance Period*.
  For both values the specification is possible either as date values or as integers in the customary *CodeMeter*® format (seconds since 01.01.2000). This covers the currently time horizons of *CodeMeter*® up to a maximum of February 2136.
- The definition and programming of the PIO *Maintenance Period* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO* *Maintenance Period*
CmBoxPgm	Programming the *PIO* /pmd 62
CodeMeter License Central	Not yet implemented
Programming API	Call class *MaintenancePeriodParamSet* 51 and subsequently *MaintenancePeriodPIO* 50
Query/Check	
AxProtector	May be activated and checked
Core API	***CmCrypt2***, ***CmAccess2*** and in Managing API ***GetBoxContents***

3.2.1.10 Linger Time

The PIO *Linger Time* serves for defining a time period in seconds for how long a license remains allocated after the license of the protected application has been de-allocated or the protected application has been closed.

This allows you to implement license models which are to time-control the restart of protected applications.

- The *Linger Time* option is specified in number of seconds.
- The definition and programming of the PIO *Linger Time* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The behavior of the *Linger Time* depends on the selected access mode you defined in the runtime settings in *AxProtector*.

Access mode	Linger Time behaviour
Normal user limit	Each license lingers since in this mode each started instance allocates a license. It does not make a difference whether the *CmContainer* is found locally or on a network.
Station Share	For each PC a license lingers since in this mode several started instances on the PC allocate a single license.
Exclusive Mode	A license lingers since in this mode the protected application is allowed to start once on a PC. In a server client environment then the client will not be able to use a server license for the defined time.
No user limit	A license does not linger since in this mode any number of instances can be started locally or on a network without the allocation of additional licenses.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO* *Linger Time*
CmBoxPgm	Programming the *PIO* /plt 60
CodeMeter License Central	Not yet implemented
Programming API	Call class *LingerTimeParamSet* 51 and subsequently *LingerTimePIO* 50

© Copyright 2018 WIBU-SYSTEMS AG

Query/Check	
AxProtector	May be ignored
Core API	***CmAccess2***

3.2.1.11 Minimum Runtime Version

The PIO *Minimum Runtime Version* serves for defining a minimum *CodeMeter*® Runtime version required to be installed.

This allows you to implement license models which requires a specific *CodeMeter*® Runtime version for features implemented by a specific runtime version.

- The *Minimum Runtime Version* options holds Major and Minor version information and optional the Build Number.
- The definition and programming of the PIO *Minimum Runtime Version* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO Minimum Runtime Version*
CmBoxPgm	Programming the *PIO* **/pmrt** 62
CodeMeter License Central	Not yet implemented
Programming API	Call class ***MinVersionParamSet*** 51 and subsequently ***MinVersionPio***.
Query/Check	
AxProtector	Option -D 74 (section licensing systems)
Core API	***CmAccess2*** checks the current runtime with the programmed required *Minimum Runtime Version*.

3.2.1.12 Named User

The PIO *Named User* restricts use of a license to a predefined set of credentials..

Possible credentials types are:

- The system login username
- The system login username and the domain name
- A user-defined text, specified during license programming. It may hold a text up to 127 bytes of data (UTF-8).

For easy verification of the PIO contents the plaintext of the credentials used for creation can be included in the PIO (privacy/safety implications given).



Currently, only one *Named User* entry can be defined for each *Product Item*.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Not yet implemented
CmBoxPgm	Programming the *PIO* **/pnmu** 63
CodeMeter License Central	Not yet implemented
Programming API	Call class ***NamedUserParamSet*** and subsequently ***NamedUserPIO***
Query/Check	
AxProtector	---
Core API	***CmAccess2*** and in the Managing API ***GetInfo***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.1.13 Customer Owned License Information (COLI)

The PIO *Customer Owned License Information (COLI)* serves for the display of additional personalized license information in *CodeMeter WebAdmin*, e.g. name of the licensee or serial number.

- The *Customer Owned License Information* option may hold up to 256 bytes of data (UTF-8).
- The definition and programming of the PIO *Customer Owned License Information (COLI)* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CmBoxPgm	Programming the *PIO* */pcoli* 58
Programming API	Use classes *CustomerOwnedLicenseInfoParamSet* 51 and *CustomerOwnedLicenseInfoPio* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.14 User Data

The PIO *User Data* serves for saving visible data. For example, you are able to store configuration data.

- The *User Data* option may hold up to 256 bytes.
- Write and read access is not limited i.e. a *Firm Security Box* (FSB) is not required. At runtime of the protected application this PIO can be changed by anyone.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO* *User Data*
CmBoxPgm	Programming the *PIO* */pud* 64
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetUserData* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.15 Protected Data

The PIO *Protected Data* serves saving additional visible data in binary format. For example, you are able to store specific information on the customer.

- The PIO *Protected Data* option may hold up to 256 bytes.
- The definition and programming of the PIO *Protected Data* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO* *Protected Data*
CmBoxPgm	Programming the *PIO* */ppd* 63
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetProtectedData* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.1.16 Extended Protected Data

The PIO *Extended Protected Data* serves for saving additional but secure data in binary format.

- The PIO *Extended Protected Data* comprises 256 types. Each type may have a length up to 256 bytes.

  Of the types 128 (index 0-127) are reserved for the ISV and 128 (index 128-256) for Wibu-Systems.

- The definition and programming of the PIO *Extended Protected Data* (write access) requires a *Firm Security Box* (FSB). However, the read access is not limited.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO Extended Protected Data*
CmBoxPgm	Programming the *PIO* /ped 58
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetExtendedProtectedData* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in Managing API ***GetBoxContents***

3.2.1.17 Hidden Data

The PIO *Hidden Data* serves for saving additional secure - but only with a password readable - data in binary format. For example, you are able to store individual key constants for decryption operations.

- The PIO *Hidden Data* comprises 256 types. Each type may have a length of up to 256 bytes.
  The default and optimal entry length equals 242 bytes which is shorter than the maximum entry length of 256 bytes. Using this default length optimizes hardware resource performance in the *CmContainer*. Reading data is automatically done across entries, i.e. when an entry is completed by the maximum length automatically the next entry is read.

  Of the types 128 (index 0-127) are reserved for the ISV and 128 (index 128-256) for Wibu-Systems.

- The definition and programming of the PIO *Hidden Data* (write access) requires a *Firm Security Box* (FSB). The read access is feasible only with a valid password.
- The reading and writing of data from or into a *CmContainer* is featured also without FSB access at runtime using WUPI functions, if the *CmContainer* is specially prepared.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO Hidden Data*
CmBoxPgm	Programming the *PIO* /phd 59
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetHiddenData* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in the Managing API ***GetBoxContents***

3.2.1.18 Secret Data

The PIO *Secret Data* serves for saving additionally secure - but invisible - data in binary format. For example, you are able to store individual keys for decryption operations .

- The PIO *Secret Data* comprises 256 types. Each type may have a length of up to 256 bytes.

© Copyright 2018 WIBU-SYSTEMS AG


Of the types 128 (index 0-127) are reserved for the licensor and 128 (index 128-256) for Wibu-Systems.

- The definition and programming of the PIO *Secret Data* (write access) requires a *Firm Security Box* (FSB). A read access is not possible.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	Programming the *PIO* *Secret Data*
CmBoxPgm	Programming the *PIO* /psd 63
CodeMeter License Central	Programming the PIO
Programming API	Call class *ProductItemParamSet* 51 and subsequently *SetSecretData* 51
Query/Check	
AxProtector	---
Core API	***CmAccess*** and in the Managing API ***GetBoxContents***

3.2.1.19 Access Password

The PIO *Access Password* allows you to restrict access to a *Product Item* level exclusively to accesses using a password.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	not yet implemented
CmBoxPgm	Programming the *PIO* /papwd 57
CodeMeter License Central	Activate this option
Programming API	Call class *ProductItemParamSet* 51
Query/Check	
AxProtector	---
Core API	Managing API ***GetBoxContents*** and ***CmGetInfo***

3.2.1.20 Maximum Encryption Rate

The PIO *Maximum Encryption Rate* allows you to specify a maximum number of encryptions performed per 30 seconds on accessing a programmed license entry. This security feature obstructs potential hackers to draw conclusion from encrypted data by using brute force methods.

The following references show you which *CodeMeter®* tools and interfaces you may use to create, edit or delete this PIO or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	not yet implemented
CmBoxPgm	Programming the *PIO* /pmer 61
CodeMeter License Central	Activate this option
Programming API	Call class *ProductItemParamSet* 51
Query/Check	
AxProtector	---
Core API	Managing API ***GetBoxContents*** and ***CmGetInfo***

© Copyright 2018 WIBU-SYSTEMS AG

3.2.2 Allocation order of licenses

On designing license strategies, the allocation order defines the criteria according to which a license is accessed, if several licenses with the same *Firm Code:Product Code* pair and a *Feature Code* are available.

Paramount here is that the first license to be allocated provides the least restriction by way of its Product Item Options. This results in a weighted license allocation order displayed below:

- License has no restrictions
- License has an *Expiration Time* or an activated *Usage Period* - here the license with the longest remaining time is allocated first
- License has an *Unit Counter* - here the license with the highest counter is allocated first
- License has a non-activated *Usage Period*
- License with a not reached *Activation Time* - here a forced license access is required
- License with a consumed *Unit Counter*, reached *Usage Period* and reached *Expiration Time* - here a forced license access is required

3.2.3 Module Item

Module Items allow the organizational grouping of different license entries required for a product. This is especially helpful when using the license transfer feature.

Module Items represent *Product Items* located below the level of a *Product Item*. Thus it can have all license properties, such as a common *Product Item*. A *Module Item* inherits all PIOs (*Product Item Options*) from the super-ordinated *Product Item*. All *PIO*s explicitly set in the *Module Item* overwrite the values inherited from the parent *Product Item*. However, adding additional license properties to a *Module Item* is possible.

A *Product Item* may hold 0 to n *Module Items*. A *Module Item* itself cannot have *Module Items*.

The following references show you which *CodeMeter*® tools and interfaces you may use to create, edit or delete this PIO, or how to conduct queries or checks.

Create/Edit/Delete	
CodeMeter License Editor	---
CmBoxPgm	Programming the *PIO* /pmi 62
CodeMeter License Central	---
Programming-API	Calling class *ProductItemParamSet* 51 and then *ProductItemParamSet* 51
Query/Check	
AxProtector	----
Core API	***GetBoxContents*** reads the original properties of the *Product Items* used as *Module Item*; ***CmGetInfo*** reads the actual effective properties.

3.2.4 Security with Capital S

The following table shows additional advantages featured by hardware-based protection with *CodeMeter*® (*CmDongle*).

Advantage	Description
Firmware runs protected in the hardware	The firmware, i.e. key storing and calculation, and the related encryption and decryption are safely protected and run in the Smartcard chip of the *CmDongle*. The hacker cannot analyze the chip because it represents a black box.
Hardware can be locked	If you detect an attack within your software (this is done automatically by our tools), you are able to send a lock command to the *CmDongle* directly from within your software. This command locks all your licenses, i.e. those at your *Firm Item* level. You are able to reactivate these licenses by remote programming. However, until reactivation the *CmDongle* behaves as if those licenses (and the keys involved) were not present. The hacker does not have a second try.
Counters cannot be set back by a backup	Counters are safely stored in the Smartcard chip of the *CmDongle*.The values of the counters cannot be manipulated from the outside and cannot be reset by installing a backup.
Deleted licenses cannot be set back by a backup	Licenses which have been deleted in a *CmDongle* no longer exist. By transfer of a receipt, the developer is sure that the license does not exist in the current *CmDongle* and also is

© Copyright 2018 WIBU-SYSTEMS AG

Advantage	Description
	irretrievable.
Expiration Time and Usage Period are checked against the internal clock	All times and dates used, such as *Expiration Time* and *Usage Period*, are checked against the clock running internally in the Smartcard Chip. The recorded times cannot be manipulated; the internal clock cannot be set back. Consequently, an expired license is irretrievable. For further security, the developer can update the internal clock via a certified time server.
Certified Time via time server and locking	Additionally, Wibu-Systems provides globally spread time server which supply a certified time and impede time manipulation. Wibu-Systems is able to lock lost *CmDongles*. This locking is stored in these *CodeMeter®* time server, and as soon as the affected *CmDongle* is trying to update this time it is locked.
License Portability	The user wants the convenience of using software legally purchased on different computers (home, office, etc). The developer wants to make sure that his/her programs are not used illegally on multiple computers. With *CodeMeter*, both the user and developer are winners; since the license is contained on the *CmDongle*, the user can move it by simply relocating the *CmDongle*. And the developer knows that while his/her program may be installed on more than one system, it can only be used on one of them at a time.
Security against license loss by viruses and other malware	Programming (create, edit, delete) of a license in a *CmDongle* is secured by cryptography. Only you with your FSB are able to delete entries. No virus is able to destroy the user's licenses.

Table 3: Advantages of *CodeMeter®* Hardware

3.2.5 License Models - Mapping Variety using CodeMeter

As described, each license entry may have *Product Item Options* combined in any way. This allows you as a licensor to map your license strategy using a variety of license models. The following table shows the basic license models from which you are able to build your individual license strategy.

License model	Description
Single-user	The license is stored on a PC (*CmActLicense*) or in a *CmDongle* connected to the PC. The software runs on the same computer/machine, or in the cloud.
Network	The license is stored on a central server in the network. It is used by PCs as a floating license. In embedded applications its main use is as an emergency license. It has little significance in the cloud.
Feature-on-demand	Individual licenses are used to activate specific products and modules. This allows you to generate extra turnover through the sale of add-ons. In embedded applications, service technicians can connect a suitable CmDongle to access hidden service functions.
Perpetual license	The license is issued permanently and never expires.
Demo version	The user can only access the functionalities you specify for a limited time.
Rental, leasing, subscription	You specify how long the license is valid for. You can use *CodeMeter License Central* to automatically extend the license
Pay-per-use	Billing is based on the number of units used. You can decide whether the billing unit is based on time or function. In the cloud billing with this type of license is usually volume-based.
Software assurance	This is a perpetual license which includes a service level agreement. Users have automatic access to updates as soon as they are available.
Downgrade right	The license covers the right to optionally use older versions of a program. With this license a key customer can make sure the same version of a program is used throughout the company. They can decide when to update to the new version.
Grace Period license	The license covers the right to optionally use the next version of a program. This means you can still sell the current version even though a new version has been announced.
Volume license (with control)	You specify the number of licenses a key customer can activate.
Volume license (without control)	The key customer is sent an activation code which they can use as often as they wants. The number of licenses appears in the contract but is not controlled (*CmActLicense* only).
Version licensing	It is possible to choose whether the license covers one or several versions of the same software.
Cold standby	The user owns a spare license which they can use if there is a problem with their current license. They have to activate the license before it can be used.
Hot standby	The user owns a spare license which they can use immediately if there is a problem with their current license.
High availability licenses	The user owns a redundant license server ("2 out of 3" principle).
Overflow licenses	The user can activate more licenses than they own. Usage is monitored though and can be subsequently billed.

© Copyright 2018 WIBU-SYSTEMS AG

License model	Description
License borrowing	The user can borrow a license from a license server to use on a local computer (*CmActLicense*) or in a *CmDongle* for a fixed time. When the license expires, it is automatically returned to the license server and can no longer be locally accessed. It is also possible to manually return the license before the expiry date.
User-specific licenses	The license is associated with a specific user name.
Computer-specific licenses	The license is associated with a specific computer name.
Time zone licenses	The license can only be used in the geographical region (time zone) specified by you.

Table 4: Mapping License Models using *CodeMeter*

3.2.5.1 Implementing License Models

This section briefly describes a series of examples showing you how to implement different license models using *CodeMeter*®. For the necessary programming you use the respective tools 50.

All you need for building these examples is your valid *Firm Code*. With it you are allowed to create your *Firm Item* level. Then you store your actual license entries at this level and configure the available *Product Items*.

Of course you are able to alter or combine these example models so that they exactly match your license strategy.

3.2.5.11 Local Single User Licenses

This license is exclusively available locally on a PC to which the matching *CmContainer* is connected.

Define a freely chosen *Product Code* at the Product Item level. Set the Product Item Option *License Quantity* for example in *CmBoxPgm* 59, to the value `local`.

3.2.5.12 Concurrent-/ Floating License in the Network

This license is centrally provided by a server and allows the concurrent use of licenses for a specified number of clients. Only *CodeMeter Runtime* can act as license server. *CodeMeter Embedded* can act as client if the LAN option is enabled.

Define the number of licenses simultaneously used in the network by the *Product Item Option* License Quantity.
Activate *CodeMeter License Server* on the favored PC to which the *CmContainer* is connected. *CodeMeter License Server* is already integral part of the *CodeMeter* runtime environment. You activate the server in *CodeMeter WebAdmin*. In *CodeMeter WebAdmin* you are also able to monitor the number of allocated licenses allocated by single PC. The license access knows different access modes.

- UserLimit**:** for each started instance of your software exactly one license is allocated.
- StationShare**:** for each PC, the application can have any number of multiple instances, only one license per PC is allocated.
- NoUserLimit**:** the software can be started but no license is allocated. Even if all license have been already allocated.

When operating a *CmContainer* on a virtual machine the license must be directly available in the session. A sharing between different sessions is not possible.
When operating a *CmContainer* on a terminal server, or in multi-user mode on Windows XP or Windows Vista you can avoid license infringement by setting *CodeMeter License Server* in *AxProtector* to the minimum version 3.20.
Then *CodeMeter* automatically handles sessions, and each session is interpreted as a separate PC including all access modes.

3.2.5.13 Demo Versions

Time Limit

If time-limiting the license of a demo version, you define a fixed *Expiration Time* or an *Usage Period* .

If using a fixed Usage Period, the moment the protected application is started for the first time determines when the testing period ends. This allows you to implement 'real' demo versions where the time-span is not limited to a previously defined date.

Define an Expiration Time or an Usage Period at the Product Item level.

© Copyright 2018 WIBU-SYSTEMS AG

In *CodeMeter* the Expiration Time or Usage Period are checked against the internal clock in the *CmContainer* (for time synchronization see *here*).

Runtime Limit 'Start x-times'

If limiting the license of a demo version by the number of allowed software starts, you define an Unit Counter at the *Product Item* level.

e.g. The Unit Counter is decremented each time the software is started (value = runtime / time unit). In the software you then decrement the Unit Counter per time unit by a value of 1.

Functional Limit

Demo versions may also differ from standard versions by different functional scopes (so-called crippling). In this case, you may license a functional limited demo version (see modular licenses 33).

3.2.5.14 Modular Licenses

Modular licenses allow you to variably license special parts of a protected application (modules or functionalities). You have two options to implement modular licensing: by using different *Product Codes*, or the *Product Item Option Feature Map*.

Different Product Codes

e.g. Define a *Product Code* for each module (functionality) of the application. In this way, you may activate up to 6,000 different modules, and separately define further modular license options, such as, Expiration Time or License Quantity (network licenses).

Feature Map

e.g. Define a *Feature Code*. Each bit value in the *Feature Map* then exactly stands for a single module (functionality). By programming the respective *Feature Map* you activate the single modules (functionalities) you wish, for example, for demo purposes.

Licenses for single modules may also be distributed and span several *CmContainer*. For example, a standard basic version runs machine-bound while the service technician with his/her *CmContainer* gets access to extended functions.

3.2.5.15 Leasing

Licenses in the realm of leasing allow the definition of a period of time in which the use of a software is licensed.

e.g. Define an *Expiration Time* or an *Usage Period* at the *Product Item* level.

In *CodeMeter* the Expiration Time or Usage Period are checked against the internal clock in the *CmContainer* (for time synchronization see here).

3.2.5.16 Pay-per-use Licenses

Licenses in the realm of pay-per-use are based on billing for the actual starts of a software or its modules, for example, pay-per-click, pay-per print, pay-per-start, etc. This guarantees maximum flexibility, e.g. acquiring additional customer which prefer to pay the software on a per-use basis.

e.g. Define an Unit Counter at the *Product Item* level. Before or after the respective action in your application you decrement the counter by one or more units. When the counter equals the value of 0 the license becomes invalid. You may also limit the use by defining a number of allowed action calls.

For different actions you may optionally use the same Unit Counter or several Unit Counter.

© Copyright 2018 WIBU-SYSTEMS AG

3.2.5.17 Downgrade/Version Managment

Downgrade

A downgrade license model grants the license to use a former instead of the current licensed version of the same product.

e.g. Define a Feature Map at the *Product Item* level. Each bit in the Feature Map then represents a version. For example, you now may simultaneously grant a floating license for three PCs including a downgrade privilege, i.e. the licensee is able to start either the old or the new version on all three PCs, but both versions together only on a maximum of three PCs at the same time.

In total the number of started applications cannot exceed the number you defined in the PIO License Quantity.

Version Management

e.g. Define a Feature Map at the *Product Item* level. Each bit in the Feature Map represents a version you are able to separately activate or deactivate.

If you simultaneously set the PIO *License Quantity* to a value of 1, the user is able to use only one of the activated versions at a time. Of course this also works on a network environment with more than one license.

3.2.5.18 Overflow

Overflow license models cover the provision of additional pay-per-use licenses for ensuring a short-term increase of license requests.

e.g. Define two *Product Codes* at two different *Product Item* levels for the main and the overflow license entry. The main entry holds no Unit Counter and a License Quantity according to the number of licenses acquired. In contrast, the overflow entry holds a high Unit Counter and a License Quantity according to the desired number of overflow licenses.
Now, all main license entries that are allocated, use the overflow entries for the software. Then you are able to decide for yourself whether you show this in the software and eventually slow down software performance. In addition, you may monitor the Unit Counter on a regular basis to record how often (or how long) overflow licenses have been used.

If implementing overflow licenses, you are still able to protect using *AxProtector*. Use *AxProtector* with the *Product Code* of the main entry and set the access mode to NoUserLimit.

3.2.5.19 Hot / Cold Standby

License models in the realm of system reliability and stability (so called "mission critical" applications) may require cold and hot standby licenses.

Cold Standby

By cold standby we mean the practice of keeping a second non-activated *CmDongle* next to the *CmDongle* in use. In the case the first *CmDongle* fails, the second backup *CmDongle* is used.

e.g. Define a Usage Period at the *Product Item* level.
Deliver your customer this backup *CmDongle* with a Usage Period of a couple of days. When the license entry is used the first time, the Usage Period starts. This license allows the user to bridge the failure but is not a full-fledged second license.

Hot Standby

The hot standby practice also has a backup *CmDongle* ready but it operates parallel to the actual *CmDongle*. Only when the system fails; the backup *CmDongle* is used.

e.g. Define two separate *Product Codes* for the main and the backup license for two separate *CmDongle*. The main license holds no Unit Counter. The backup license is implemented with a very high Unit Counter for the second *CmDongle* connected to a second PC. Connect the *CmDongle* with the main license and without Unit Counter to the license server, and the *CmDongle* with the backup license and very high Unit Counter to the backup server. Using the server search list you define the sequence of the license to be allocated. In the case that the first server fails, the second server with the backup licenses is used automatically. Checking the Unit Counter on a regular basis avoids misuse.

© Copyright 2018 WIBU-SYSTEMS AG

3.2.5.110 Named User Licenses

Named user licenses cover the use of a software bound to a named user who additionally has to successfully authenticate him/herself to the system.

e.g. Define a *Protected Data* field at the *Product Item* level and save the User ID to it.
In the software you then check whether the separately saved User ID is identical to the User ID calculated for the actual user.

3.2.5.111 Machine-bound Licenses

In some cases it may be necessary to bind a *CmContainer* to a specific PC, machine or user.

e.g. Define a *Protected Data* field at the *Product Item* level and save the ID to it.

3.2.5.112 License Borrowing

The license borrowing model allows for the use of software applications on a PC not connected to the license server controlling access to the protected application. The license is borrowed for a limited time. However, the total number of licenses available in the network is not affected. This license mobility is required, for example, when licenses have to be available on a separate laptop on the road, or at the home office.

e.g. For license borrowing you require prepared *CmContainer*, one at the server and one at the client side.
The licensee borrows and returns licenses using the "Borrowing" tab in *CodeMeter Control Center*. In *CodeMeter WebAdmin* the license allocation is displayed.

3.2.6 Security by Encryption

The security of *CodeMeter*® is based on encryption. The software or modules or data in the software to be protected are encrypted by the developer before shipping. The key for decryption is part of the license the developer generates for the end-user. On the user's side, parts of the software are decrypted only when needed (on demand decryption). After use, these parts then are re-encrypted.

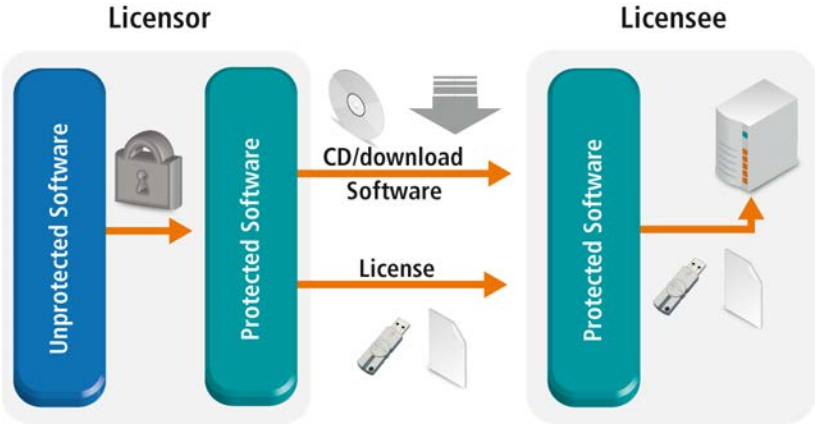


Figure 7: Security by Encryption

3.2.6.1 Key Derivation - One License Entry - Many Keys

The software is encrypted at runtime on the user's PC. Also at runtime, the communication between the software and the license stored in the *CmContainer* is encrypted. A common practice among hackers is to use a "record / playback" tool at the interface in order to discover the encryption key. This is prevented by *CodeMeter*® because Wibu-Systems uses the concept of alternating keys. As the figure below shows these keys are generated in the *CmContainer* by a derivation, our so-called "secret sauce".

© Copyright 2018 WIBU-SYSTEMS AG

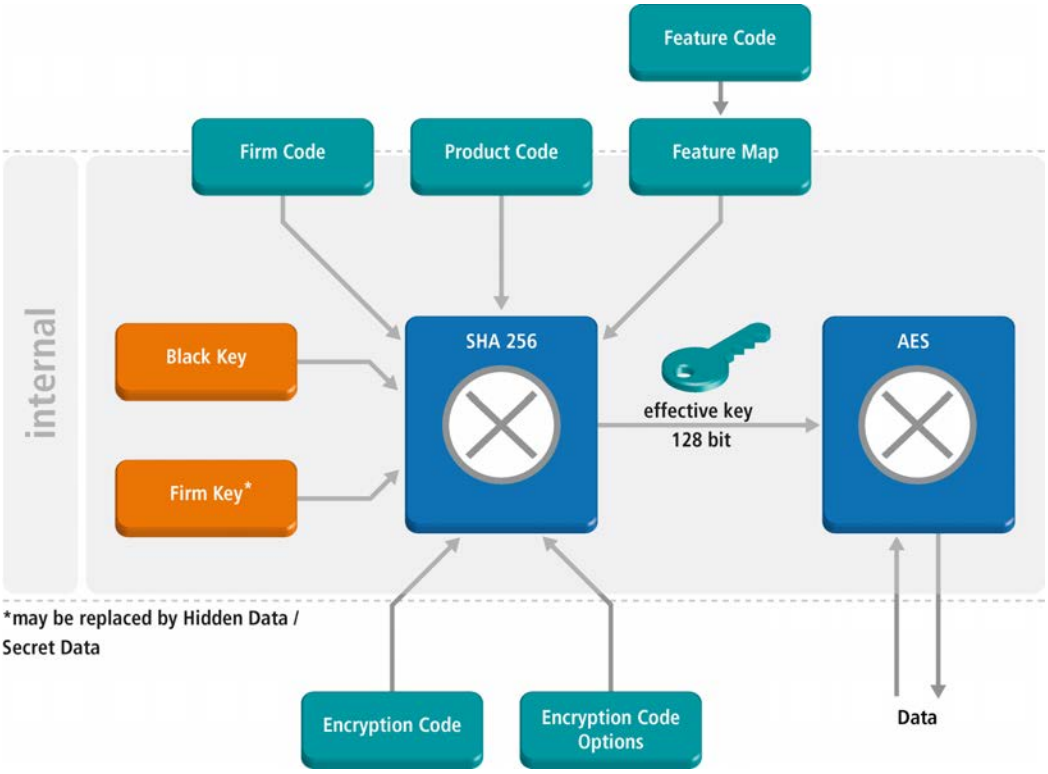


Figure 8: Key Derivation *CmDongle*

In the case of *CmActLicense*, the activation and software-based variant of *CodeMeter*, instead of *Black Key* and *Firm Key* the information on the *Product Items* and an individually derived *Product Item Secret Key* exist as parameters.

Effective key

The "effective" key for encryption and decryption operations is composed of several *CmContainer* internal and external parameters. Within the *CmContainer* the key is then calculated by a hash function (SHA 265).

Black Key, *Firm Key* in *CmDongle*

At first, there exist two non-readable parameters inside the *CmDongle*. The *Black Key* is a secret key known only to Wibu-Systems. And the *Firm Key* is initially delivered for *CmDongle* by Wibu-Systems but can be changed afterwards by the licensor.

Please note that this is possible only in the case of *CmDongle*, the hardware-based variant of *CodeMeter* but not for *CmActLicense*, the activation and software-based variant of *CodeMeter*.



In the case of *CmDongle*, alternatively, also a *Secret Data* or *Hidden Data* field may be used instead of the *Firm Key*.

Firm Code, *Product Code*, *Feature Code*

Additionally, there exist four more parameters located in the *CmContainer*. However, for a license they can be programmed from the outside into the *CmContainer*: *Firm Code*, *Product Code*, *Maintenance Period* with a defined *Release Date*, and *Feature Map* - where feature activating is done via single bits defining the *Feature Code*.

Encryption Code

These first parameters are static and are integrated for single license entries into the key derivation. In contrast, the two additional parameter *Encryption Code* and *Encryption Code Options* (ECO) are dynamic and get integrated at runtime as a variable for a given protected application.

The *Encryption Code* is a fix value (you are free to choose) which is stored for encryption and decryption operations.

Encryption Code Options

Moreover, the runtime *Encryption Code Options* complete the composition of the parameters. The *Encryption Code Options* contain information about required *Product Item Options* used in the key derivation and how they are checked. They cover the following options (for the detailed description of how to use the single options see in the *CodeMeter API Guide* online help the section "**Functions | Encryption API | CmCrypt2**"):

- PIO *Unit Counter*, *Activation Time* and *Expiration Time*
- Access modes of started instances to available licenses

© Copyright 2018 WIBU-SYSTEMS AG

- Value by which an *Unit Counter* is decremented at a specific action (delta mask),
- Check whether a *Certified Time* update has been occurred since the *CmContainer* was connected or activated,
- Decrement of a special counter by a value of 1 when a debugger has been detected (*Firm Access Counter*, FAC).

Hash Function (SHA 256)

The calculation resulting from all the parameters via a hash function (SHA 256, Secure Hash Algorithm) represents the effective key. Within the symmetric encryption method AES (Advanced Encryption Standard) the effective key then is used to encrypt and decrypt data.

The hash value is used as a kind of "finger print" to ensure that data has been actually encrypted and decrypted with the same effective key. If illegal manipulation has been attempted, and thus changes of the parameters have occurred in between encryption and decryption operations; a completely different hash value is the result, hence, the operation is not executed, and the license access results in a failure.

This process of the key derivation takes place within the *CmContainer* and the derived keys are used by all *CodeMeter®* tools and interfaces when data is encrypted or decrypted.

3.2.7 Cryptography

In general, cryptographic methods and operations comprise the following objectives:

- *integrity*: contents must not be altered.
- *confidentiality*: reading the actual content by unauthorized persons is practically prohibited.
- *authentication*: the sender of a message proves the identity to the receiver.
- *non-reputability*: the sender of a message cannot deny sending it, nor the receiver receiving it.

CodeMeter® provides many cryptographic methods which meet these objectives.



AxProtector and *IxProtector* apply AES (Advanced Encryption Standard) with a key length of 256 bits in symmetric encryption and decryption operations.
Asymmetric encryption and decryption operations are executed by ECC (Elliptic Curve Cryptography) 224-bit and RSA with 2084-bit.
For ECC Wibu-Systems supports the P-224 curve variant secp224r1 with a key length of 224 bit as recommended by the U.S. American NIST (National Institute of Standards and Technology).

The *CodeMeter Core API* provides the ***CmCrypt2*** function to apply the various encryption and decryption algorithms. They include symmetric methods but also asymmetric methods for signatures and public key structures.

3.2.7.1 Direct and Indirect Encryption

In *CodeMeter®* there is a basic distinction between when an encryption operation is direct or indirect. This influences system operational performance.

Direct Encryption / Decryption

In the case of direct encryption the operation takes place in the*CmContainer.* The data to be encrypted has an exact data length of 16 bytes and is encrypted in the cryptographic unit in the *CmContainer*.



Using direct encryption does make sense for random-based checks or for encryption / decryption sequences with a short length.

Indirect Encryption

In the case of indirect encryption, first, a part of the data is directly encrypted in the *CmContainer*, and subsequently this result is integrated as an initialization vector into the remaining operation, which takes place in the PC memory.



The minimum length of data is 16 bytes, the maximum length is 4 GByte.

© Copyright 2018 WIBU-SYSTEMS AG

3.2.7.2 Symmetric Enryption

In the case of symmetric cryptographic methods for encryption operations the same key is used (see *Encryption API*).

AES

CodeMeter® applies the standard algorithm for symmetric encryption of data: the AES (Advanced Encryption Standard).



In *CodeMeter*® AES is applied with a key length of 128 bits = 16 bytes.

3.2.7.21 AES - Cipher Block Chaining Mode (CBC) (recommended)

The AES algorithm in Cipher Block Chaining Mode XORs each block of plain text with the previous cipher text block before being encrypted. This way, each cipher text block is dependent on all plain text blocks processed up to that point. Also, to make each message unique, an initialization vector must be used in the first block. A one-bit change in a plain text affects all following cipher text blocks. The encryption is sequential, i.e. it cannot be parallelized.



Wibu-Systems recommends this mode since CBC is the most commonly used mode of operation.
Although *CodeMeter* supports also other older cryptographic modes. However, their use is not recommended.
For more information see the technical guideline TR-02102-1 "Cryptographic Mechanisms: Recommendations and Key Lengths" (courtesy translation) edited by the German Federal Office for Information Security, BSI (Bundesamt für Sicherheit in der Informationstechnik) (BSI).

3.2.7.3 Asymmetric Encryption

Along with symmetric encryption, *CodeMeter*® also provides the option to asymmetrically encrypt and decrypt data by private and public keys, and to create and verify signatures for authentication checks. The *CodeMeter API Guide* provides the necessary API functions and function blocks: Authentication API, Encryption API, Blocks

3.2.7.31 ECC - Elliptic Curve Cryptography

ECC (Elliptic Curve Cryptography) is an approach to public key cryptography based on elliptic curves. Here both communicating parties have different keys: a public key and a private key. The private key is kept secret, while the public key may be widely distributed. The private key never leaves the *CmDongle*. From it the public key can be calculated to be deposited and saved authentically with the opposite party.

For ECC Wibu-Systems only supports the P-224 curve variant secp224r1 with a key length of 224 bit as recommended by the U.S. American NIST (National Institute of Standards and Technology).

3.2.7.32 ECIES - Elliptic Curve Integrated Encryption Scheme

The ECIES (Elliptic Curve Integrated Encryption Scheme) is a public key encryption scheme which allows data to be sent to the private key (in the *CmDongle*) owner when the public key is known. Generally, data is encrypted using the function ***CmCryptEcies*** and is decrypted with ***CmCrypt*** and the algorithm `CM_CRYPT_ECIES_STD`.

3.2.7.33 ECDSA - Elliptic Curve Digital Signature Algorithm

The ECDSA (Elliptic Curve Digital Signature Algorithm) is a signature algorithm generating a hash value (digest) from a document. This digest then is signed with the private key. In contrast to ECIES, here the private key is used for creating the signature, while the public key is used for verification.

See the relevant functions ***CmCalculateDigest***, ***CmCalculateSignature***, ***CmValidateSignature*** and ***CmGetPublicKey*** of the *Authentication API* needed for performing the authentication procedures.

© Copyright 2018 WIBU-SYSTEMS AG

3.2.7.34 RSA

The RSA algorithm is named after its inventors (Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman) and is suitable for signing as well as encryption.

3.2.7.4 Additional Encryption Algorithms

Certified Time Encryption

This encryption operation refers to the *Certified Time* feature and presents a special function for the *CodeMeter*® time server.

SHA - Secure Hash Algorithm 256

The SHA algorithm is a cryptographic hash algorithm creating a 256 bit (32 byte) checksum to be used as a "finger print". In the realm of asymmetric encryption, the SHA-256 algorithm is used for preparing a signature in order to calculate a control value of constant length for the data to be signed.

3.3 CodeMeter Start Center

CodeMeter Start Center serves as communication center on your system on which you develop and comes with the *CodeMeter* Software Development Kit. It allows the access to basic *CodeMeter*® tools, applications, and interfaces.

3.3.1 Structure and Navigation

You access *CodeMeter Start Center* via the **"Start | All Programs | CodeMeter"** start menu (Press "Windows" key to open Start screen | Type "CodeMeter Start Center" | Press "Enter" key). The user interface is divided into two areas: an upper menu bar, and a lower display window, allowing access to single applications.



Figure 9: *CodeMeter Start Center*

© Copyright 2018 WIBU-SYSTEMS AG

3.3.1.1 Menu Bar

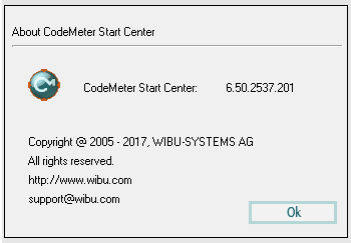
File Menu

Element	Description
Language	*CodeMeter Start Center* provides several language settings for the user interface. Currently, you may choose from eight languages: Chinese, German, English, Spanish, French, Japanese, Italien, and Russian.
Exit	The **"Exit"** menu item closes *CodeMeter Start Center*. Alternatively, you may close the window via the **"x"** control or the **"Exit"** button in *CodeMeter Start Center*.

Tools Menu

The tools menu contains the button item of the *CodeMeter Start Center* window and also provides an alternative way to open *CodeMeter WebAdmin* to view existing licenses in *CmContainer.*

Help Menu

Element	Description
Search for Updates	Finds software updates available on the Wibu-Systems Internet sites.
Developer Guide	Displays this document as a PDF file.
Wibu-Systems Homepage	Links to the Wibu-Systems homepage.
Mail to Support	Opens an e-mail addressed to Wibu-Systems Support.
About	Opens a windows holding version information. 

Lower Display Window

Element	Description
AxProtector	*AxProtector* 72 automatically protects your software. *AxProtector* places an envelope around your compiled software without altering the source code of the application.
API Guide	*CodeMeter API Guide* provides a way for you to integrate software protection into your source code. It can generate source code for you and explains the *CodeMeter Core API* and the *Software Protection API* (WUPI).
CodeMeter License Editor	*CodeMeter License Editor* is an intuitive graphical tool for creating, editing and deleting licenses in *CmDongles*. It supports locally connected *CmDongles* and also file-based remote programming 69.
Samples	Finds samples for various programming languages explaining how to integrate the interfaces (*Software Protection API* (WUPI), *Core API*). Click the button to open the related directories and get a short overview of existing examples.
Exit	Use the **"Exit"** button to close *CodeMeter Start Center*.

© Copyright 2018 WIBU-SYSTEMS AG

4 CmEmbedded - CodeMeter for embedded devices

This section gives an overview of CmEmbedded modules and architecture[41] and module specifics[42] you have to consider on employing. Moreover, it comprises platform requirements and operating systems specifics[44], lists available delivery packages[47], and shows how to integrate CmEmbedded[48] libraries.

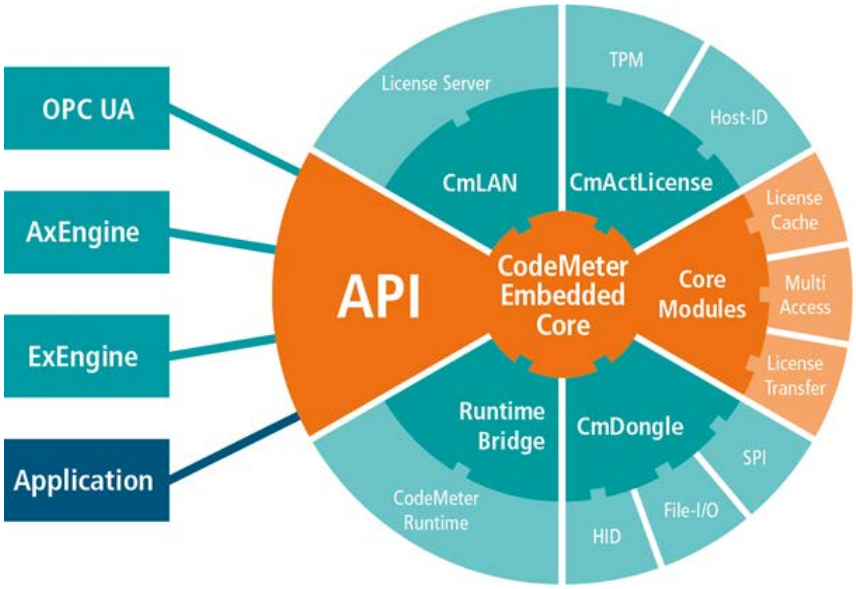
4.1 Modules and Architecture

CmEmbedded consists of different function modules. The core is always available and has the additional modules for multiple access and license caching.

The core does also provide the *CodeMeter Core* API which is used by the *CodeMeter* tools, such as, the AxEngine or own implementations in the user software. The core is extended by the modules for accessing different *CodeMeter* containers, such as, *CmDongles*, *CmActLicense*, or LAN access.

The default delivery package contains all functions. That gives the user the choice which Licensing system (*CmDongle*, *CmActLicense*) will be used.

For implementations with limited resources, it is possible to build *CmEmbedded* versions consisting only of the needed function modules. This will decrease the size of the library.



CmDongle

Standard module in *CmEmbedded*. Includes all functions for the use of secure elements like USB dongle, microSD-, SD-, CF- and CFast-Cards and ASICs via SPI or USB.

CmActLicense

Module for *CmActLicense* software based licenses. The binding of the *CmActLicense* adapter needs to be implemented by the customer. See chapter below.

CmLAN

Module for accessing licenses from a CodeMeter License server via network

Concurrent (License Cache)

Module allowing multiple processes to access safely license data of specific *CodeMeter* Container. The safety is guaranteed over locking mechanism concept and all processes coordinates between each other via shared memory.

Runtime Bridge

Module mostly for developers running a *CmEmbedded* protected executable on a computer also running the *CodeMeter* runtime. The container access is then tunneled to the runtime.

Automatic Protection

In the embedded environment there are two possible ways of automatic protection. They differ in the architecture.

© Copyright 2018 WIBU-SYSTEMS AG

AxProtector

AxProtector encrypts the binary and adds the *AxEngine* and *CmEmbedded*. The result is a self decrypting ELF (Executable and Linkable Format) file. *AxProtector* is available for Linux and Android platform.

ExProtector

ExProtector encrypts the binary and adds just the credentials. The decryption is done by an external *ExEngine* with *CmEmbedded*. The needed procedure is integrated into the loader of the operating system, such as the *libc* library. The adaption into the operating system is highly individual for the used release of the operating system and needs always to be done customer-specific.

By nature, *ExProtector* can just be used by operating systems which provide the ELF loader in source code, such as Linux. The steps for an integration are available described in a separate document "MIPSel Linux ELF Loader Fixed Key Adaptation" available on request.

For VxWorks there is a ready integrated solution available. The adaption to the loader is integrated in the delivered package and can be used out-of-the-box.

4.1.1 Module Specifics

CmDongle

CmDongle under *CmEmbedded* is a standard module. This module brings all the required functionalities to support and manage the communication of connected devices. Devices cover secure hardware like USB dongle, SD Cards and ASICs via SPI or USB.

Encrypted communication and CmDongle

 Linux:

On Linux, encrypted communication uses shared memory and keeps the specified communication key in shared memory. The communication key locates in the ram until the session is expired or the shared memory is manually deleted.

 Windows

On Windows, the communication key stay alive until the last process is terminated (license cache with multi processes mechanism). So the communication key is negotiated each time the program (one process or multi-process) is launched.

The encryption requires to consider another *CodeMeter* security mechanism which manages the use to a defined number of executions in a defined time. Then after a defined time span an error message is issued which serves to protect the communication encryption against brute force attacks (*FULA*, *Function Use Limit Administration*). The *FULA* default values are: number of encryption calls (`r`): `50`; call time-span (`tr`): `1200s` (`20`min.); lock-time after exceeding (`ts`): `300s` (`5`min.)

Here on Windows and for all libraries without license cache problems may occur, if some important issues are not considered:

- On Windows the shared memory access is not persistently saved to the file system. Thus Wibu-Systems recommends to use *CmEmbedded* in a service / application which is not constantly started and exited.
- Generally, problems may occur if machines perform an aggressive power management for USB ports. Since on hibernating the firmware does not update the *BoxTime*, the 5 minutes lock time-out runs slowly or never expires. Then CmDongles can be used only after a replug.

Please note that *CmEmbedded* secures only the channel between systems and *CmDongle* but not the complete environment. Measures avoiding direct access to shared memory keys have to be designed by the software vendor.

CmActLicense

Module for *CmActLicense* software based licenses. The binding of the *CmActLicense* adapter needs to be implemented by the customer. See chapter below.

CmLAN

Module for accessing licenses from a *CodeMeter License Server* via network

Concurrent (License Cache)

Module for the concurrent access to one *CodeMeter* container. The function arbitrates between the instances of *CmEmbedded* instances. The coordination bases on shared memory.

© Copyright 2018 WIBU-SYSTEMS AG

Runtime Bridge

Module mostly for developers running a *CmEmbedded* protected executable on a computer also running the *CodeMeter* runtime. The container access is then tunneled to the runtime.

The following *Product Item Options* (PIO) are supported only by Runtime Bridge:

- License Quantity
- Customer Owned License Information
- Linger Time
- Named User
- Minimum Runtime Version
- Access Password
- Maximum Encryption Rate

Moreover, only the following license models can be employed using the Runtime Bridge module.

- Concurrent-/ Floating License in the Network
- Named User Licenses
- Machine-bound Licenses

Linux:

The *CmEmbedded* version of the AxEngine (linked to *CmEmbedded*) holds the Runtime Bridge module. If the *CodeMeter* runtime is loaded, this may lead to the fact that licenses stored in *CmEmbedded CmActLicense* are no longer found since *CmEmbedded* redirects to the runtime.

If you use *AxProtector* with *CmEmbedded* AxEngine, please make sure you stop the runtime on the system on which you want to simultaneously use *CmEmbedded* managed *CmActLicense*.

4.2 Operating System Specifics

The following sections cover operating system specifics you are to consider before employing *CmEmbedded*.

4.2.1 VxWorks

Currently,*CmEmbedded* can be employed only in Kernel mode, i.e. the code can be executed only in the context of a Kernel task. The execution within a RTP (Real Time Process) is currently not supported.

VxWorks knows two library variants:

- DKM (Downloadable Kernel Module): The DKM can be loaded into the Kernel at runtime. The Kernel solves missing symbols.
- Static library: The static library is an object file archive employed in the link step of a DKMs, VIP(VxWorks Image Project)s or RTPs. There exists no option to load these at runtime.

Wibu-Systems recommends to make sure that *CmEmbedded* exists/is installed only once. Please link DKMs only dynamically to *CmEmbedded*.

The License Cache / Concurrent Access is not required in VxWorks Kernel mode, if only a single *CmEmbedded* instance is used in the system and all other modules are dynamically linked.

4.2.2 QNX

The implementation of the random number generator on uses `/dev/urandom`.

Please start the Resource Manager before you use *CmEmbedded* the first time.

This ensures a high entropy to get a good quality of generated keys required for the *CmContainer* communication.

© Copyright 2018 WIBU-SYSTEMS AG

4.3 CmDongles - Platform requirements and operating systems specifics

In the following tables platform requirements and operating systems specifics for using CmDongle form factors and storage device types are listed.

4.3.1 Android

Information	Minimum Version 5 on ARM platform Tested on Raspberry Pi 3 with the following Android versions • 5.1.1 • 6.0.1 • 7.1.2 • 8.1.0 Deviant *CmEmbedded* functions: Android has a different design compared to platforms like Linux or Windows, so there are some functional exceptions: • Concurrent access is not implemented. Android has security mechanisms to prevent memory sharing between apps. The outcome of this is, that license cache and the encrypted communication between *CmEmbedded* and dongle – both based on the concurrent access mechanisms – are not implemented. This means, that only one instance of *CmEmbedded* protected or licensed apps can be run or more accurate – use one *CmContainer*.
Form Factor	**CmStick** USB Mass Storage The volume holding the `codemtr.io` file must be mounted with respective privileges (see Linux). Driver components: • `CONFIG_USB_STORAGE` • `CONFIG_BLK_DEV_SD` • `CONFIG_SCSI_MULTI_LUN` HID HID communication requires `USBDEVFS`. Note: The access to a *CmStick* in HID mode requires an extended privilege setup since user processes by default, cannot access HID raw devices.

4.3.2 Linux

Information	starting with Kernel 2.6.x
Form Factor	**CmStick** USB Mass Storage The volume holding the `codemtr.io` file must be mounted with respective privileges, e.g. `mount -tvfat -o uid=1000,gid=1000 /dev/sdx /mnt/cmdongle`. Driver components: • `CONFIG_USB_STORAGE` • `CONFIG_BLK_DEV_SD` • `CONFIG_SCSI_MULTI_LUN`

© Copyright 2018 WIBU-SYSTEMS AG

	HID starting with Kernel 2.6.39 Driver components: • `CONFIG_HID` • `CONFIG_HIDRAW` Note: The access to a *CmStick* in HID mode requires an extended privilege setup since user processes by default, cannot access HID raw devices.

The *CmEmbedded* License Cache bases on POSIX shared memory of the `libc` library. Please make sure, that in the Linux Kernel the component "`CONFIG_TMPFS`" is activated. Otherwise, the library with activated License Cache can not be used.

4.3.3 QNX

Form Factor	***CmStick*** USB Mass Storage *CmEmbedded* expects the `codemtr.io` volumes by default in `/fs` z.B. /fs/usb1. As in the case of Linux and Android, here also the respective privilege setup must exist for the `codemtr.io` file. Driver components: • `libusbdi.so` • `devu-ehci.so` • `devu-ohci.so` • `io-usb` • `devb-umass` • `io-blk.so` • `cam-disk.so` HID Driver components: • `libusbdi.so` • `devu-ehci.so` • `devu-ohci.so` • `io-usb` ***CmCard*** If *CmCards* are to be addressed by a "native" interface, the respective device driver of the board support package (BSP) must be integrated. • *CmCard SD/microSD* `devb-mmcsd-xxx` • *CmCard/CF* `ATA devb-eide` • *CmCard/CFast* `SATA devb-ahci`

4.3.4 VxWorks

Information	Versions 6.9 and 7.0
Form Factor	***CmStick*** USB Mass Storage

© Copyright 2018 WIBU-SYSTEMS AG

CmEmbedded expects *CmStick* volumes by form `/bd[n]` or `/bd[n]:[lun]`, e.g. `/bd0`

Driver components:

- `INCLUDE_USB_GEN2_STORAGE_INIT`

Host Controller Components

- `INCLUDE_EHCI_INIT` -> USB 2.0 *CmStick/M*
- `INCLUDE_EHCI_INIT` -> USB 1.1 *CmStick*

Note:

If the standard components cannot be used, please take a look in the `target.ref` document of the used board support package (BSP).

HID

Minimum VxWorks 6.9

Driver components:

- `INCLUDE_USB_GEN2_HELPER`
- `INCLUDE_USB_HOST_CLASS_INIT`

Note:

Binaries including up to 6.9.2 cannot be used with VxWorks 6.9.3 or higher since they are not binary compatible.

CmCard

CmCard SD/µSD

CmEmbedded expects *CmCard SD/microSD* volumes by form `/sd[n]:[partn]`, e.g. `/sd0:1`

Driver components:

- different depending on VxWorks Version and board support package (BSP).
- In VxWorks 6.8 the respective driver component of the BSP must be integrated, e.g. `DRV_STORAGE_FSL_SDHC`.
- In VxWorks 6.9 a standard component `DRV_STORAGE_SDHC` exists. VxWorks 7 provides specific components, see version 6.8.
  More detailed information locate in the `target.ref` document of the board support package (BSP).used.

CmCard/CF

CmEmbedded expects *CmCard/CF* volumes by form `/ata[portn]:[partn]` with ATA Card Reader and `/ahci[portn]:[partn]` with SATA Card Reader.

Driver components:

- different depending on VxWorks Version and board support package (BSP), e.g. `INCLUDE_DRV_STORAGE_AHCI` oder `INCLUDE_DRV_STORAGE_INTEL_AHCI`.

CmCard/CFast

CmEmbedded expects *CmCard/CFast* volumes by form `/ahci[portn]:[partn]`.

Note:

You are of course able to connect all cards using a USB Card Reader.

General Note:

The `codemtr.io` file should locate in a separate FAT16/FAT32 partition since the file cache is deactivated for this volume. This involves slower read and write processes.

© Copyright 2018 WIBU-SYSTEMS AG

4.3.5 Windows

Form Factor	**CmStick** USB Mass Storage Driver components: • SYSGEN_USB • SYSGEN_USB_STORAGE ***CmCard*** If *CmCards* are to be addressed by a "native" interface, the respective device driver of the board support package (BSP) must be integrated.

4.3.6 macOS

Information	Minimum macOS Version 10.12

4.4 Delivery packages

CmEmbedded can be delivered in two forms:

Form	Description
Static libraries	Default form of delivery. The lib corresponding to the customers system is delivered as a tested and released package.
Source code	Form of delivery for systems, environments and platforms which are not available in our tool chains and build systems or for customers needing to adapt the *CmEmbedded* to own operating systems or hardware platforms. Source code delivery always needs an existing NDA.

Operating System	Platform	Version	Test Hardware	AxProtector Support	ExProtector Support
Linux	ARM	v6, v7, v8	RasPi2, Pandaboard, Nanopi, Cubietruck, BeagleBoneBlack	x	x
Linux	Intel	x86, x86_64	zbox	x	
QNX	ARM	v7	Pandaboard		
QNX	Intel	x86	Intel Desktop Board D525MW		
VxWorks	ARM	v7	SabreLite IMX6Q		x
VxWorks	ppc		P2020		x
VxWorks	Intel	x86, x86_64	NITX315, Dell Optiplex		x
Windows version 7 and higher	Intel	x86, x86_64	zbox		
Android v5 and higher	ARM	v8	RasPi3	x (native)	
macOS	Intel	x86-64	MacMini	x	

© Copyright 2018 WIBU-SYSTEMS AG

4.5 How to Integrate CmEmbedded (libraries)

This section describes how to integrate the self-generated or shipped Wibu-Systems embedded libraries into your own applications.

Initializing the *CmEmbedded* library: Android, Linux and QNX

Starting with version Version 2.0 the CmEmbedded library uses so-called constructor and destructor functions.

Functions with this attribute are called before entry (constructor) and after leaving (destructor).

By default, this feature is activated (only CLANG and GCC).Then, if CmEmbedded is linked to an executable program, the library is automatically initialized and de-initialized without explicitly manually calling the CodeMeter Core API functions *CmStaticLibOnInit* and *CmStaticLibOnExit*.

This loads CmActLicense Universal Firm Code (UFC) and CmActLicense Firm Code licenses at library initialization, i. e. any resources, e. g. directories holding license container, must be available before program start.

This feature can be deactivated via a flag in the `cm_config.h` [141] file.

```
#define ENABLE_LIB_CONSTR_DESTR OFF
```

Then the *CmStaticLibOnInit* call has no effect since initialization had already been performed.

A *CmStaticLibOnExit* call explicitly de-initializes the library and the following destructor call has no effect.

Compatibility to previous programs using these both Core API functions is guaranteed.

4.5.1 Linux

Using the static library (`libwibucmembedded.a`)

- Add the following define `_CODEMETER_API_STATIC_LIB`

Static libraries delivered by Wibu-Systems use GLIBC starting with version 2.4 and higher. The following additional libraries must be linked:

- librt (`-lrt`)
- libpthread (`-lpthread`)

4.5.2 Windows

Microsoft Visual Studio

Using the static library (`wibucmembedded.lib`)

- Include the `wibucmembedded.lib` library using "**Configuration Properties** | **Linker** | **Input** | **Additional Dependencies**"
- Append the define `_CODEMETER_API_STATIC_LIB` in "**Configuration Properties** | **C/C++** | **Preprocessor**".
- If *CmEmbedded* will contain *CmLAN* and/or *Runtime Bridge* support, the file `ws2_32.lib` must be added to the library list.
- If *CmEmbedded* will contain *CmDongle* support, the file `setupapi.lib` must be added to the library list.

Static libraries use the static multi-threaded version of the run-time library. Depending on the build configuration, the runtime library must be changed to multi-threaded (/ MT) or multi-threaded debug (/ MTd).

© Copyright 2018 WIBU-SYSTEMS AG

4.6 How to identify the CmEmbedded version

In order to identify the implemented version of *CmEmbedded*, use the *CodeMeter* Core-API.

Calling function `CmGetInfo(CM_GEI_VERSION)` you obtain the version information in the related structure.

```
/***** CMVERSION *****/
typedef struct __CMVERSION {
  CMUSHORT musVersion;
  CMUSHORT musSubVersion;
  CMUSHORT musBuild;
  CMUSHORT musCount;
  CMUSHORT musYear;
  CMUSHORT musMonth;
  CMUSHORT musDay;
  /* CMUSHORT musReserve; */
} CMVERSION;
```



Wibu-Systems recommends to program a separate function for identifying the *CmEmbedded* version.

Moreover, using *CodeMeter Universal Support Tool* (*cmu*) on the developing system (see *CodeMeter* Developer Guide) you obtain the version from the binary library `libwibucmembedded.a` locating in the directory "`\build\result\lib`" of the respective delivered package.

For example:

```
cmu -f libwibucmembedded.a -v
cmu - CodeMeter Universal Support Tool.
Version 6.50c of 2017-Sep-12 (Build 2640) for Linux
Copyright (C) 2007-2017 by WIBU-SYSTEMS AG. All rights reserved.
Operating System:
 Name:         Ubuntu 16.04.3 LTS (Kernel 4.8.0-36-generic)
 Architecture: x86_64
Current system user:
 username:               bs
 domain:
RunTime version information:
 cmu:                    6.50.2640.503 2017-09-12
 CodeMeter-Service:      not running
 Security:               not running
File version information:
 File:         /mnt/hgfs/shared/libwibucmembedded.so.2.1
 Version:       2.01.286.200 2017-11-24
 Architecture: 64bit Linux/x86_64
```

© Copyright 2018 WIBU-SYSTEMS AG

5 CmDongle on embedded devices

This section covers three parts.

First, you get an overview of how to program licenses and license options 50 into the *CmContainer* using the tools which are part of the installed *CodeMeter* Software Development Kit (SDK) on your development system.

If you later want to use the programmed Context File (`*.WibuCmRaC`) / Update File (`*.WibuCmRaU`) files for licensed accesses to your application, you must use the *CodeMeter Embedded Core API* 84 to integrate desired function calls into the source code of your application running on the embedded device.

Second, if supported for your embedded environment, *AxProtector CmE* 72 is able to automatically encrypt already compiled executables without the need to alter your source code. This part includes sample command lines, a brief introduction, and available options.

Of course you can also protect your application without *AxProtector CmE*; then use the *CodeMeter Embedded Core API* 84 for integrating protection and security features (e.g. CmCrypt2 85).

Third, the shortly described *CodeMeter Embedded Core API* 84 (including code samples) allows direct access to licenses securely saved in a *CmContainer* from within your application. The complete API documentation is separately available for download from the Wibu-Systems website (https://www.wibu.com/manuals-guides.html).

For communication with the CmASIC - a dongle solution as chip - next to USB also CmDongle and SPI (Serial Peripheral Interface) 149 communication is available.

5.1 Licensing: Programming of CmContainer

The functionality of *CmContainers* for *CmEmbedded* is compatible to the desktop version of *CodeMeter*. For detailled information of programming and managing licenses please refer to the documentation of the *CodeMeter Software Development Kit*.

After you protected an application, you have several options to program *CmContainer* you want to deliver.

Basically, the programming of license information (*Firm Code*, *Product Code*, and *Product Item Options*) into *CmContainer* is accomplished by three methods:

- **local**: programming of locally connected *CmContainer* using a locally connected *Firm Security Box* (FSB).
- **file-based**: reprogramming of a Context File (license request) (`*.WibuCmRaC`) send by the licensee to the licensor into a Update File (license update) (`*.WibuCmRaU`) and the subsequent import by the licensee into his/her *CmContainer*.
- **protocol-based (SOAP)**: programming and managing of Context and Update Files (`*.WibuCmRaC` and `*.WibuCmRaU`) is done by the Internet supported network protocol SOAP (Simple Object Access Protocol) using *CodeMeter License Central*.

© Copyright 2018 WIBU-SYSTEMS AG

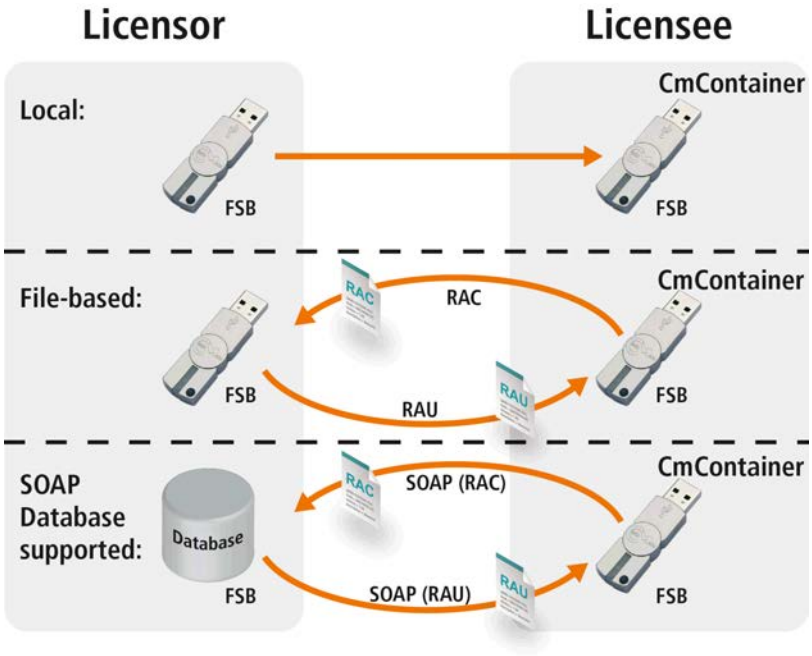


Figure 10: *CmContainer* Programing Options

For all three methods *CodeMeter* provides several tools:

- *CmBoxPgm* commandline tool for batch programming of *CmContainer* in production.
- *CodeMeter License Editor* graphic tool to program *CmDongles* or local tests of licensing strategies.
- *CodeMeter License Central* database-supported tool to create, manage, and deliver licenses using SOAP in a *Desktop* and *Internet* edition.

The tools *CmBoxPgm*, *CodeMeter License Editor*, and *CodeMeter License Central* you can use for file-based remote programming 69, i.e. *CodeMeter Field Activation Service (CmFAS)*.

Most of the tools base on the *CodeMeter Programming API* (HIP - *High Level Programming Interface*). This class-oriented interface allows you to access any object or process required to program or organize license entries in a *CmContainer* and features extended customizing.

The *Programming API* is available for many programming languages. Existing help programs have been generated for respective interfaces, for example, Delphi, Visual Basic, .NET, and Java. For more detailed information on the *Programming API* open the start menu item **"Start | All Programs | CodeMeter | Documentation | Programming-API"** (Press "Windows" key to open Start screen | Type "Programming API" | Press "Enter" key).

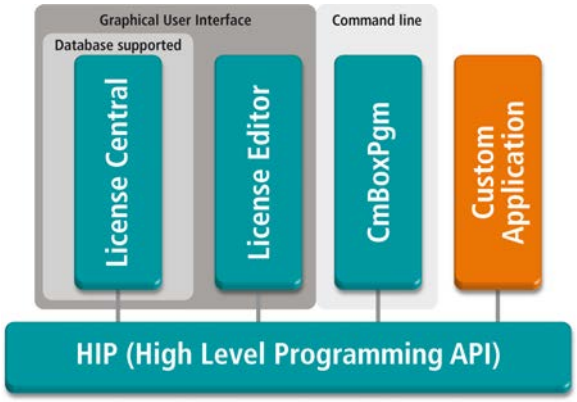


Figure 11: Tools for *CmContainer* Programming

© Copyright 2018 WIBU-SYSTEMS AG

5.1.1 CmBoxPgm

Besides programming of *CmDongle* using *CodeMeter License Central*[67], *CodeMeter* also provides the option for local programming of *CmContainer* using a commandline (console).

The local programming of *CmContainer* requires and uses up *CodeMeter* transactions. Please note that a *Unit Counter* in you FSB is decremented each time you locally program a *CmContainer*.

Advantages of the console

Commandline programming bears the special advantage to use scripts and batch files. Efficiently supported by a variety of parameters you are able to program processes, and apply them to several *CmContainer* in one go.

Application

Such advantages are essential, in particular, when you mass produce *CmContainer* or automate test processes.

Open CmBoxPgm

Open *CmBoxPgm* commandline via: "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**". *CmBoxPgm* opens in the user directory path.

Call start menu item **"Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt"**.

Press "Windows" key to open Start screen | Type "CodeMeter Command Prompt" | Press "Enter" key.

CmBoxPgm opens in the user directory path.

5.1.1.1 Using CmBoxPgm

By default, you find *CmBoxPgm* in form of the executable file `cmboxpgm.exe` in the directory "`%\Program Files%\CodeMeter\DevKit\bin`". For other operating systems find *CmBoxPgm* at the customary locations.

In the following description of options you can alternatively use the prefix '–' instead of '/'.

5.1.1.2 Basic Commands

This section describes the main commands of *CmBoxPgm*.

A basic option always concludes a command sequence which targets a *Firm Item* or *Product Item*.

If you receive error code `3912` or `9010` on programming (`No license profile available`), the complete `CmFirm.wbc` file is missing required for programming or it does not contain the proper, i.e. your `Firm Code`.
Thus import the `CmFirm.wbc` file you received from Wibu-Systems when you became customer. Most easy via drag & drop using *CodeMeter Control Center*
If the `CmFirm.wbc` is no longer available to you, please contact Wibu-Systems Support.

The following options are available.

`/ca` - Add

Add a new entry into the *CmContainer* (*Firm Item* or *Product Item*).

`/cau` - Add/Update

Update an existing entry in the *CmContainer* (*Firm Item* or *Product Item*) or adding a new entry if it does not exist yet.

`/cu` - Update

Update an existing entry in the *CmContainer* (*Firm Item* or *Product Item*).

When adding a *Firm Item* starting with CodeMeter version 4.50 the following is valid:
If on adding a `Firm Item` the text of the *Firm Item* is not explicitely defined, the text for the *Firm Code* defined in the file `CmFirm.wbc` is used. By default the text attribute for *CmDongle* is set to "`Text=Test Kit Firm Code`" and for *CmActLicense* to "`Text=CmAct Testkit`".
If you want to change this text you have two options: either by explicitely setting[55] the *Firm Item Text* or to edit the text attribute in the file `CmFirm.wbc`. You find the file `CmFirm.wbc` in the directory "`%ProgramData%\CodeMeter\DevKit`".

`/cd` - Delete

Deleting an existing entry from the *CmContainer* (*Firm Item*, *Product Item* or *Product Item Options*).

© Copyright 2018 WIBU-SYSTEMS AG

/cdx - Delete if
Delete an entry from the *CmContainer* if available (*Firm Item* or *Product Item*).
/l - List
List the contents of selected *CmContainer*, *Firm Items*, *Product Items* or *Product Item Options*.

Programming examples

CmBoxPgm /l
Lists the content of the first *CmContainer* which is not a *Firm Security Box*. Corresponds to `CmBoxPgm /qn1 /l`
CmBoxPgm /qb1 /l
List the content of a *CmContainer* with an index o. It does not make a difference whether the *CmContainer* holds an *Firm Security Box* entry or not. Corresponds to `CmBoxPgm /qn1:f /l`
CmBoxPgm.exe /qs1-1234 /l
List the content of a *CmContainer* with the serial number of `1-1234`.
CmBoxPgm.exe /qn2,4:f /l
List the content of the *CmContainer*in the index range between 2 and 4 including the *Firm Security Boxes*.

5.1.1.3 CmContainer Options

This section describes the available options referring to *CmContainer*.

You address *CmContainer*:

- individually: using either the serial number (`/qs`) or using the index (`/qb`),
- as a selection: using the index (`/qb`).

When addressing please note whether the *CmContainer* to be programmed is a *Firm Security Box* (FSB).

The following options are available:

Command	/qb - Box index
	Determines the *CmContainer* to be programmed. Expects a decimal value as argument which is interpreted as index. Must not be used together with options **/qn** or **/qs**.
Syntax	`/qb<Index>`
Command	**/qn[,y][:f] - Box Index Range**
	Determines the index range of *CmContainer* to be programmed. Two indices (1, 2, 3,...) representing the range's lower and upper limit, may be specified. The range's upper limit `y` is optional. If this value `y` is not specified, the range's lower limit is used. *Firm Security Boxes* (FSB) will be excluded automatically, if not explicitly requested by setting the FSB mode **[:f]**. The option must not be used together with option **/qb** or **/qs**.
Syntax	`/qn[<Index of first CmContainer>,]<Index of last CmContainer>[:f]`
Command	**/qs[m-]s - Serial Number**
	Use this option to specify the Mask and Serial Number of the *CmContainer* to be programmed. Expects a decimal value as argument. It is recommended to use both parameters to uniquely identify a *CmContainer*. Must not be used together with options **/qb** or **/qn**.
Syntax	`/qs<Serial Number>`
e.g.	`/qs1-12345` or `/qs2-12345` The mask parameter refers to the used *CodeMeter* chip version. Please note that the mask `m` and serial number `s` may change anytime depending on *CodeMeter* chip versions. Please consider these possible changes for future *CmContainer* generations on programming queries for *CmContainer*.
Command	**/pwd - Box password**
	Changes the *CmContainer* password.
Syntax	`/pwd "<old Password>"="<new password>"`
Command	**/r - Recursive Removal**
	Deletes each license entry (*Firm Item*, *Product Item*, *Module Item*) in a *CmContainer* for which required privileges exist in an available FSB (locally connected or on a network). If applied to the complete *CmContainer* all *Firm Items* are deleted.

© Copyright 2018 WIBU-SYSTEMS AG

Command	/r - Recursive Removal
	Please note that you require the respective delete privileges which may - depending on the license scheme - involve additional costs. If applied to a *Firm Item* all *Product Items* and *Module Items* are deleted at this *Firm Item* level. The *Firm Item* is kept however is empty.
Syntax	`/r`

Command	/rau - Remote Activation Update
	Executes the programming sequences stored within the specified *Remote Activation Update* file on the target *CmContainer* as far as applicable.
Syntax	`/rau:"<RaU file>"`

Command	/mrau - Merge Remote Activation Update Files
	Merges multiple Remote Activation Update RaU files (`*.WibuCmRaU`) into a single RaU file. The input files must be of the same type and should be specified in the order of their creation. The merging of RaU files applies only to *Universal Firm Code* licenses. *Universal Firm Code* RaU files must address the same target *CmContainer* and *Firm Item*.
Syntax	`/mrau:<input file 1>,<input file 2>[,...,<input file n>],<output file>`

Programming examples

`CmBoxPgm`	lists the content of the first *CmContainer* which is not a *Firm Security Box*. Corresponds to `CmBoxPgm /qn1 /l`
`CmBoxPgm /qb1 /l`	lists the content of the *CmContainer* with index 1. It does not matter whether the *CmContainer* holds a *Firm Security Box* entry or not. Corresponds to `CmBoxPgm /qn1:f /l`
`CmBoxPgm /qs1-1234 /l`	lists the content of the *CmContainer* with the serial number `1-1234`.
`CmBoxPgm /qn2,4:f /l`	lists the content of the *CmContainer* in the index array 2 - 4 including *Firm Security Boxes*.

5.1.1.4 Commandline Syntax

Option Blocks

The general syntax in *CmBoxPgm* follows the pattern of so-called option blocks. Option blocks summarize single programming sequences or lists of commands. At the same time, target specification and options are included.

The pattern of an option block is as follows:

```
<target declaration>  <target-specific options>  <operation>
```

Target Declaration

The initial part of an option block includes the required information on the target of an operation. Such target can include:

- single *CmContainer* or a selection of *CmContainer*,
- single *Firm Items*,
- *Product Items*

The syntax of the target declaration corresponds to the hierarchical structure of entries in a *CmContainer* and is ordered from the general to the specific.

Addressing a *Product Item* with the specification of the related *CmContainer* or an array of *CmContainer*, continues with the specification of the *Firm Codes*, the *Firm Item* which holds the *Product Item*, and ends with the specification of the *Product Item*. The typing effort is reduced because parts of the target declaration do not have to be repeated when already specified in a previous option block. If you add a series of *Product Items* to the same *Firm Item*, it is sufficient to one-time specify the *Firm Item* at the beginning of a programming sequence for the *Product Items*.

Target-specific Options

The middle part of an option block holds the target-specific options. Depending on the operation, that part can be or should be left blank.

Operation

The concluding part holds the specification of the operation to be executed.


Specifying the concluding part is mandatory

The most important operations correspond to the basic options and comprise the adding, updating and deletion of *Firm Items and Product Items*. Moreover, the contents of selected *Items* or complete *CmContainer* can be listed in the commandline.

© Copyright 2018 WIBU-SYSTEMS AG

For the time reference used while programming, the following time zones are valid:

Abbreviation	Description
CET	Central European Time
CST	Central Standard Time
EET	Eastern European Time
EST	Eastern Standard Time
MST	Mountain Standard Time
PST	Pacific Standard Time
UTC	Universal Time Coordinated

Table 5: Time Zones in *CmBoxPgm*

Month specifications follow the pattern: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

5.1.1.5 Firm Item Options

This section describes the various options related to *Firm Item*.

Firm Item commands are structured the following way:

```
f<Firm Code> [<Firm Item Options>] <Main Command>
```

The following options are available.

Command	`/f` - Firm Code
	Defines the *Firm Codes* to be used. Expects an unsigned decimal value as argument.
Syntax	`/f<value>`
Command	**`/fac`** - Firm Access Counter (FAC)
	Sets the *Firm Access Counters* to the specified value. Expects an unsigned decimal or a hexadecimal value, preceded by `0x` as argument. The default setting is `0xffff`.
Syntax	`/fac<value>`
	Sets the given *Firm Item* License Transfer Permissions. Expects as arguments either the token `none` to deny any permissions or the token `pull` to allow pull operations, i.e. the active collection of transfer licenses.
Syntax	`/fltperm:none|pull`
Command	**`/fpta`** - Firm Precise Time, absolute
	Sets the *Firm Precise Time* to the specified absolute value. Expects a date optionally followed by a time and the time zone as argument. If the time zone is omitted, the system's time zone is used instead.
Syntax	`/fpta<YYYY><Month><DD>[,<SS>:<MM>:<SS>[PST|MST|CST|EST|UTC|CET|EET]]`
	`/fpta2006Dec31,23:59:59UTC`
Command	**`/fptr`** - Firm Precise Time, relative
	Adds the specified number of days to the current value of the *Firm Precise Time*. If the *Firm Item* doesn't exist yet, the current system time plus the specified offset will be set as *Firm Precise Time*. Expects an integer value greater than or equal to zero as argument. If this *Firm Item* has not yet been created, the system time plus the specified number is used as *Firm Precise Time*. For example, `/fptr1` corresponds to 1 day immediately starting.
Syntax	`/fptr<number of days>`
Command	**`/ft`** - Firm Item Text
	Sets the *Firm Item Text*. Expects a character string (up to 256 characters) enclosed in double quote characters as argument.
Syntax	`/ft:"<Text>"`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/fuc** - Firm Update Counter
	Sets the *Firm Update Counter* to the specified value. Expects an unsigned decimal value as argument. This counter automatically increases while programming entries.
Syntax	`/fuc<wert>`

Programming examples

`CmBoxPgm /qn1,4 /f206 /ft:"My Company" /ca`	Adds a new *Firm Item* with the *Firm Code* 206 to the *CmContainer* within the index range from 1 to 4. *Firm Security Boxes* are excluded. The *Firm Precise Time* is set to the current system time. The *Firm Item Text* corresponds to the string specification "My Company". *Update Counter* and *Access Counter* are set to default values.
`CmBoxPgm /qb2 /f206 /fuc42 /fac0x1066 /cu`	Updates the *Firm Item* with the *Firm Code* 206 in the second *CmContainer*. The *Firm Item Update Counter* is set to a value of 42 and the *Firm Item Access Counter* set to a value of `0x1066`.
`CmBoxPgm /qs1-1234 /l /f206 /cu /l`	Lists the content of the *CmContainer*, updates the *Firm Item* with the *Firm Code* 206, and subsequently relists the content.
`CmBoxPgm /f206 /cd`	Deletes the *Firm Item* with the *Firm Code* 206.

5.1.1.6 FSB Entry Options

This section describes available options referring to the *Firm Security Box* (FSB).

Commands related to *Firm Security Boxen* (FSB) have the following setup:

```
/fsb<Firm Code> [<FSB Options>] <Main Command>
```

The following options are available:

Command	**/fsb** - FSB Entry
	This option initiates a FSB command sequence. Expects the *Firm Code* the FSB entry refers to as argument.
Syntax	`/fsb<Firm Code>`
Command	**/fk** - Firm Key
	Use this option to specify a new *Firm Key* (32 Bytes). Please act with extreme caution when using this option! When changing the existing *Firm Key* you deeply interfere in encryption and programming processes! Because then all future encryption operations and all *CmContainer* programming will refer to this "new" *Firm Key* ! "New" encrypted applications will not run with "old" programmed *CmContainer*! Vice versa, "old" encrypted applications will not run with "new" programmed *CmContainer*! For your own safety, the option "Changing the *Firm Key*" has to be activated by Wibu-Systems.
Syntax	`/fk:0x<hex data>`

5.1.1.7 Product Item Options

This section describes the various options related to *Product Items* or *Product Item Options* (PIO).

Necessary requirement for programming *Product Items* and PIO is an already existing *Firm Item*.

Product Item options commands are structured in the following way:

```
/f<Firm Code> [...] /p<Product Code>[...] [<PIO Options>] <Main Command>
```

TVB (**Trailing Validation Block**)

You have an option to perform an additional check before executing programming sequences. This holds for all *Product Item Options* with the exception of Text and User Data.
Using so-called *Trailing Validation Blocks* [TVB] you may define dependencies for single programming sequences.
Depending on set data (d), serial numbers (s) or update counter (u), commands are only executed when meeting the

© Copyright 2018 WIBU-SYSTEMS AG

specified criteria. For example, a programming is performed only with a specified serial number, or with a specified number of permitted updates.

By default, all TVBs are set, i.e. the programming sequences vary with a maximum, and the programming is possible only into the desired *CmContainer* including the specified status.

Command	**/p** - Product Code
	Defines the *Product Code* to be used. Expects an unsigned decimal value as argument. Optionally, the item reference or the *Feature Code* can be specified as further selection parameters. The *Item* reference value must be enclosed in square brackets. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/p<value>[=<new value>][:<Feature Code>|<reference>][,<TVB dep.>]`

e.g.		
	Selection of first *Product Items* with *Product Code* 13	`/p13`
	Selection of *Product Item* with *Product Code* 13, *Feature Map*=`0x00000001`	`/p13:0x00000001`
	Selection of *Product Item* with *Product Code* 13, *Product Item Reference* = 16	`/p13:[16]`

Command	**/papwd** - Access Password
	Adds, updates or deletes a Product Item's Access Password PIO. Accepts a text as password argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Please note, that using this PIO requires the *CmContainerType CmDongle* and a minimum Firmware version of 4.0.

Syntax	`/papwd:"<character string>"[,<TVB dep.>]`
	`/papwd[\"<character string>\"][,<TVB dep.>]`
	Removes the PIO (-) or sets the specified password.

e.g.		
	`/papwd:"Password"`	specifies the password `Password`.

Command	**/pat** - Activation Time
	Adds, updates or deletes the PIO *Activation Time* of a *Product Item.* Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pat-|[a<date>|r<offset>][,<TVB dep.>]` Removes the PIO (-) or sets an (a)bsolute or a (r)elative *Activation Time*.

Command	**/pata** - Activation Time, absolute
	Sets the *Activation Time* to the specified absolute value. Date inputs are accepted only before January 1st, 2100 00:00:00 UTC. Expects a date optionally followed by a time and the time zone as argument. If the time zone is omitted, the system's time zone will be used instead. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/pata<YYYY><Month><DD>[,<HH>:<MM>:<SS>[PST|MST|CST|EST|UTC|CET|EET][,<TVB dep.>]`
	or according to ISO-8601:
	`/pata<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm` or `±hhmm] [±hh][,<TVB dep.>]`

e.g.		
	Sets the *Activation Time* to December 31st, 2012, 1 second to midnight, UTC	`/pata2012Dec31,23:59:59UTC`

Command	**/patr** - Activation Time, relative
	Adds the specified number of days to the current *Activation Time*. Expects an integer value greater than or equal to zero as argument.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/patr** - Activation Time, relative
	If this *Firm Item* does not exist yet, the current system time plus the specified offset will be set as activation time. For example: `/patr1` corresponds to 1 day immediately starting. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/patr<days>[,<TVB dep.>]`

Command	**/pcoli** - Customer Owned License Information (COLI)
	Adds, updates or deletes the PIO *Customer Owned License Information* of a *Product Item*. Accepts a text (up to 256 characters) enclosed in double quote character escape sequences as argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/pcoli-`
	`/pcoli:"<text>"[,<TVB dep.>]`
	Deleting the PIO (-) or setting the specified text.

Command	**/ped** - Extended Protected Data
	Adds, updates of deletes the PIO *Extended Protected Data* of a *Product Item*. Input of the field index (type) [`0-127`] and a sequence of hexadecimal digits (up to 256 bytes) with preceded `0x`. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/ped<extended type>-|[:0x<hex data>][,<TVB dep.>]` Removes the PIO (-) or sets the specified data The specification of the hex number always has to be pair, i.e. `0x1` is invalid but not `0x01`.
e.g.	`/ped0:0x75BCD15` — Adds the decimal value 123456789 to the field (type) 0. `/ped2-` — Deletes the field (type) 2.

Command	**/pet** - Expiration Time
	Adds, updates or deletes the PIO *Expiration Time* of a *Product Item*. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pet-|[a<date>|r<offset>][,<TVB dep.>]` Removes the PIO (-).

Command	**/peta** - Expiration Time, absolute
	Sets the *Expiration Time* to the specified absolute value. Date inputs are accepted only before January 1st, 2100 00:00:00 UTC. Expects a date optionally followed by a time and the time zone as argument. If the time zone is omitted, the system's time zone will be used instead. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/peta<yyyy><month><dd>[,<hh>:<mm>:<ss>[PST|MST|CST|EST|UTC|CET|EET][,<TVB dep.>]`
	or according to ISO-8601:
	`/peta<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm` or `±hhmm] [±hh][,<TVB dep.>]`

e.g.	Sets the *Expiration Time* to December 31st 2009, 1 second to midnight, UTC — `/peta2009Dec31,23:59:59UTC` or `/peta2009-12-31T23:59:59Z`

Command	**/petr** - Expiration Time, relative
	Adds the specified number of days to the current *ExpirationTime*. Expects an integer value greater than or equal to zero as argument. If this *Firm Item* does not exist yet or in the case of an update an existing *Product Item* so far had no *Expiration Time*, the current system time plus the specified offset will be set as *Expiration Time*. For example: `/petr1` corresponds to 1 day immediately starting. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/petr** - Expiration Time, relative
Syntax	/petr<days>[,<TVB dep.>]
e.g.	Extends the *Expiration Time* by 30 days. — /petr30

Command	**/pfm** - Feature Map
	Adds, updates or deletes the PIO *Feature Map* of a *Product Item*. Expects an unsigned decimal or a hexadecimal value preceded by 0x. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	/pfm-|[<value>][,<TVB dep.>] Deletes the PIO (-).

Command	**/phd** - Hidden Data
	Adds, updates of deletes the PIO *Hidden Data* of a *Product Item*. Input of the field index (type) [0-127] and input of an ID for an extended PIO type. Either as access code or data section. The default and optimal data section entry length equals 242 bytes which is shorter than the maximum entry length of 256 bytes. Using this default length optimizes hardware resource performance in the *CmContainer*. Reading data is automatically done across entries, i.e. when an entry is completed by the maximum length automatically the next entry is read. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	Fills the *Hidden Data* PIO with user-defined data. The optimal length covers 242 Bytes. /phd<ext. type>,[<acc. code>][:0x<hex data>][,<TVB dep.>] Fills the *Hidden Data* PIO with <count> bytes of random data. /phd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>] Removes the PIO (-). /phd<ext. type>-
e.g.	/phd15:0x1122334455 — Fills the field (type)15 of the *Hidden Data* PIO with user-defined data. The optimal length covers 242 Bytes. The specification of the hex number always has to be pair, i.e. 0x1 is invalid but not 0x01
	/phd16,<acc. code>:r32 — Fills the field (type)16 of the *Hidden Data* PIO with 32 bytes of random data. The optimal length covers 242 Bytes. The *Access Code* <acc. code> can be a text input or an input of 16 bytes in hexadecimal format.

Command	**/plq** - License Quantity
	Adds, updates or deletes the PIO *License Quantity* of a *Product Item*. Accepts an unsigned decimal value as argument. Also permissions for network access can be specified. The following predefined configurations are supported.

Parameter	Description
default	sets default permissions
local	permits only local usage. If using a remote desktop connection (remote session) for accessing local only lienses, please note the following: • in the case of operating systems, such as, Windows7, 8, 10 etc. this license access works on connecting via remote session. • in the case of server operating systems, such as, Linux, Windows Server 2012, etc., this license access will not work and the event code 239 is returned.
red	sets permissions for Triple Mode Redundancy (TMR) usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of 0 and a preceding # character.
wan	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.

Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/plq** - License Quantity
Syntax	`/plq-|[<counter>][:<access permissions>][[:]#<license tag>][,<TVB dep.>]`



`/plq100`	Setup for 100 network licenses
`/plq100:wan`	Setup for 100 network licenses and CmWAN
`/plq100:red#42`	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42.

Command	**/plqa** - License Quantity

Sets the *Product Item*'s *License Quantity* to the given absolute value.
Accepts an unsigned decimal value as argument.
Also permissions for network access can be specified. The following predefined configurations are supported.

Parameter	Description
`default`	sets default permissions
`local`	permits only local usage (Universal Licenses only)
`red`	sets permissions for Triple Mode Redundancy usage. sets permissions for Triple Mode Redundancy usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of `0` and a preceding `#` character.
`wan`	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.

Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/plqa[:]<counter>[:<access permissions>][[:]#<license tag>][,<TVB dep.>]`



`/plqa100`	Setup for 100 network licenses
`/plqa100:wan`	Setup for 100 network licenses and CmWAN
`/plq100a:red#42`	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42

Command	**/plqr** - License Quantity, relative

Increments or decrements the *Product Item*'s *License Quantity* by the given value.
Accepts an unsigned decimal value as argument.
Also permissions for network access can be specified. The following predefined configurations are supported.

Parameter	Description
`default`	sets default permissions
`local`	permits only local usage (Universal Licenses only)
`red`	sets permissions for Triple Mode Redundancy usage. sets permissions for Triple Mode Redundancy usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of `0` and a preceding `#` character.
`wan`	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.

Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/plqr[:]<counter>[:<access permissions>][[:]#<license tag>][,<TVB dep.>]`



`/plqr100`	Setup for 100 network licenses
`/plqr-100`	Removes 100 network licenses
`/plq100:red#42`	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42

Command	**/plt** - Linger Time

Adds, updates or deletes the PIO *Linger Time* of a *Product Item*.
Accepts an unsigned decimal value as argument.
Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/plt-|[<seconds>][,<TVB dep.>]`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/plt** - Linger Time
	Removes the PIO (-).
e.g.	/plt15 — adds a *Linger Time* of 15 seconds to the PIO.

Command	**/pltdepth** - License Transfer Depth
	Adds or updates the *Product Item*'s license transfer depth. Accepts an unsigned decimal value between 0 and 65535 as argument.
Syntax	/pltdepth[:]<transfer depth>]
e.g.	/pltdepth:4 — adds or updates the*Product Item*'s transfer depth to a value of 4.

Command	**/pltltktarg** - License Transfer LTK Targets
	Adds or updates the *Product Item*'s list of LTK (Licenses Transfer Key) transfer targets. Accepts a list of 1-28 CmActLicense IDs (CmActId).
Syntax	/pltltktarg:<CmAct ID1>[,<CmAct ID 2>[,<CmAct ID3>]]...

Command	**/pltmbp** - License Transfer Maximum Borrow Period
	Adds or updates the *Product Item*'s license transfer maximum borrow period. Expects either a time span measured in days or the token none to specify an unlimited period as argument..
Syntax	/pltmbp:<time span>|none
e.g.	/pltmbp:3 — adds or updates the*Product Item*'s maximum borrow period to a value of 3 days.

Command	**/pltperm** - License Transfer Permission
	Adds or updates the *Product Item*'s license transfer permissions. Expects either the token none to deny any permissions or a comma separated list of permission tokens as argument. The following permissions are available:

permission	Description
fitransfer	the Firm Item may be transferred
return	the license may be returned

Syntax	/pltperm:<permissions>
e.g.	/pltperm:return — allows the transfer license to be returned

Command	**/plttarg** - License Transfer Targets
	Adds or updates the *Product Item*'s license transfer list of license transfer targets. Accepts a list of 1-28 CmActLicense IDs (CmActId).
Syntax	/plttarg:<CmAct ID1>[,<CmAct ID 2>[,<CmAct ID3>]]...

Command	**/plttype** - License Transfer Typ
	Adds or updates the *Product Item*'s license transfer list of license transfer type. Accepts one of the following transfer type tokens as argument:

Token	Transfer Type
none	no transfer allowed
movelicenses	move network licenses
borrowlocallicense	borrow license for local use

Syntax	/plttype:none|movecomplete|moveunits|movelicenses|borrowcomplete|borrowlocallicense

Command	**/pmer** - Maximum Encryption Rate
	Adds, updates or deletes a Product Item's Maximum Encyption Rate. Expects the encryption rate value given as the maximum number of encryptions per 30 seconds as argument.
Syntax	/pmer-|[<value>][,<TVB dep.>]
	Removes the PIO (-) or sets the maximum number of encryptions.
e.g.	/pmer20 — maximum encrypts 20 times per 30 seconds.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pmi** - Module Item
	Adds, updates or deletes a *Module Item*. Expects the *Module Item*'s *Product Code* given as an unsigned decimal value as argument. Optionally, the item reference or the *Feature Code* can be specified as further selection parameters. The item reference value must be enclosed in square brackets.
Syntax	`/pmi<Product Code>[:<Feature Code>|<reference>][,<TVB dep.>]`
e.g.	`/pmi13` — Select the first *Module Item* with *Product Code*=13 `/pmi13:0x00000001` — Select the *Module Item* with *Product Code*=13, Feature Map = 0x00000001 `/pmi13:[16]` — Select the *Module Item* with *Product Code*=13, item reference = 16
	On creating *Module Items* at the *Product Item* level the basic commands[52] are called separately for each *Module Item* and then for the complete *Product Item* level. `-p3000 -pt:"Parent Product Item with ModuleItems" -pup30 -pfm0x02 -pmi3120 -pt:"Module A" -puca100 -ca -pmi3130 -pt:"Module B" -puca20 -psd0:r32 -ca -ca`

Command	**/pmp** – Maintenance Period
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item*. Accepts start date and end date of the period as argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmp-|[d[<start date>,]<end date>|i[<start value>,]<end value>[,<TVB dep.>]` Removes the PIO or sets start and end of a *Maintenance Period*, either given as (d)ates or given as unsigned (i)nteger values. The start date is 2000-01-01 00:00:00.

Command	**/pmpd** – Maintenance Period (Date)
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item* using a date. Accepts start date and end date of the period as argument. The start date may be omitted. In this case it is set to 2000-01-01, 00:00:00 UTC Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmpd[<start time>,]<end time>[,<TVB dep.>]`
e.g.	`/pmpd2011Jul01,00:00:00,2012Jun30,23:59:59` or `/pmpd2011-07-01T00:00:00,2012-06-30T23:59:59` Sets a one-year *Maintenance Period* beginning with July 1st, 2011.

Command	**/pmpi** – Maintenance Period (Integer)
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item* using an integer. Accepts start and end of the period given as unsigned integer values as argument. The start date is 2000-01-01 00:00:00. The start date may be omitted. In this case it is set to a value of 0. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmpi[<start value>,]<end value>[,<TVB dep.>]`
e.g.	`/pmpi394416000` Sets a *Maintenance Period* until July 1st, 2012. The difference of 4565 days to 1.1.2000 is 394416000 seconds.

Command	**/pmrt** - Minimum Runtime Version
	Adds, updates or deletes the *Minimum CodeMeter Runtime Version*. Accepts a version number consisting of major version, minor version and optionally the build number as argument, e.g. '5.20' or '5.20.1300'. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pmrt-|[<major version>.<minor version>[.<build>]][,<TVB dep.>]`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pmrt** - Minimum Runtime Version	
	Deletes the PIO (-).	
e.g.	/pmrt5.20 /pmrt5.20.1300	adds a *Minimum Runtime Version* Major Version 5 and Minor Version 20. adds a *Minimum Runtime Version* Major Version 5, Minor Version 20 and Build Number 1300.

Command	**/pnmu** -Named User	
	Adds, updates or deletes a *Named User* PIO of a *Product Item*. Accepts a <check mode> specification followed by <configuration options> and, depending on the specified mode, (1) a user name, (2) domain and user name or (3) a user-defined text as <text> argument. The <configuration options> also allow to determine whether the specified <text> is to be stored as plain text in the *CmContainer*. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	/pnmu-|:<check mode>[,<configuration options>]:<text>[,<TVB dep.>]	
	Deletes the PIO (-).	
	Valid check mode specifications cover:	
	user	the user name is checked
	userdomain	domain and user name are checked
	userdefined	user-defined text is checked
	Valid configuration options cover:	
	noplain	plain text is not to be stored in the *CmContainer*, default
	plain	plain text is to be stored in the *CmContainer*
e.g.	/pnmu:user,plain:"Doe"	Sets user name 'Doe', to be stored as plain text.
	/pnmu:userdomain,noplain:"MyDomain\Doe"	Sets domain 'MyDomain' and user name 'Doe', no to be stored as plain text
	/pnmu:userdefined,plain:"MyText"	Sets userdefined text 'MyText', to be stored as plain text.

Command	**/pnwc** - Network License Counter
	Deprecated, please use option /plq instead (identical syntax).

Command	**/ppd** - Protected Data	
	Adds, updates of deletes the PIO *Protected Data* of a *Product Item*. Accepts a sequence of hexadecimal digits (up to 256 bytes) preceded by **0x**. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	/ppd-|[0x<hex data>][,<TVB dep.>]	
	Deletes the PIO (-).	

Command	**/psd** - Secret Data	
	Adds, updates of deletes the PIO *Secret Data* of a *Product Item*. Input of the field index (type) [0-127] and input of an ID for an extended PIO type. You specify a data range. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	Fills the *Secret Data* PIO with user-defined data. /psd<ext. type>,[<acc. code>][:0x<hex data>][,<TVB dep.>] Fills the *Secret Data* PIO with <count> bytes of random data. /psd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>] Removes the PIO (-) /psd<ext. type>-	
e.g.	/psd15:0x1122334455	Fills the field (type) *Secret Data* PIO with user-defined data. The specification of the hex number always has to be pair, i.e. 0x1 is invalid but not 0x01
	/psd16:r32	Fills the field (type) 16 of the *Secret Data* PIO with 32 bytes of random data.

Command	**/pt** - Text
	Adds, updates or deletes the PIO *Text* of a *Product Item*. Accepts a character string (up to 256 characters) enclosed in double quote characters as argument.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pt** - Text
Syntax	/pt - /pt:"<text>"
	Removes the PIO (-).
Command	**/puc** - Unit Counter
	Adds, updates or deletes the PIO *Unit Counter* of a *Product Item*. Accepts an unsigned decimal value as argument. Depending on the chosen mode the argument is interpreted as (a)bsolute or (r)elative value. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	/puc-|[a<value>|r<value>][,<TVB dep.>]
	Removes the PIO (-) or sets an (a)bsolute or (r)elative value of the *Unit Counter*.
Command	**/puca** - Unit Counter, absolute
	Sets the *Unit Counter* to the specified value. Expects an unsigned decimal value smaller than or equal to 4294967294 as argument for Firmware versions newer than 1.18.
	For a firmware version prior to 1.18 the maximum value is 16777215.
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	/puca<value>[,<TVB dep.>]
e.g.	/puca25 — Sets the value of the *Unit Counter* PIO to the absolute value of 25.
Command	**/pucr** - Unit Counter, relative
	Increments the *Unit Counter* of a *Product Item* by the specified amount. Expects an signed decimal value in the range [-2147483648, 2147483648] as argument for Firmware versions newer than 1.18.
	For a firmware version prior to 1.18 the range is limited to [-16777215, 16777215].
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	/pucr<signed value>[,<TVB dep.>]
e.g.	/pucr10 — Increases the value of the *Unit Counter* PIO by a value of 10.
Command	**/pud** - User Data
	Adds, updates or deletes the PIO *User Data* of a *Product Item*. Accepts a sequence of hexadecimal digits (up to 256 bytes) preceded by 0x.
Syntax	/pud-|[0x<hex data>]
	Removes the PIO (-).
e.g.	/pud0x1122334455 — Assigns the specified value to the *User Data* PIO. The specification of the hex number always has to be pair, i.e. 0x1 is invalid but not 0x01.
Command	**/pup** - Usage Period
	Adds, updates or deletes the PIO *Usage Period* of a *Product Item*. Input as integer value greater than or equal to a value of null.
	Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	/pup-|[:<days>][,<TVB dep.>]
	Removes the PIO (-).
e.g.	/pup:30 — adds or updates an *Usage Period* of 30 days.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pupa** - Usage Period, absolute
	Sets the length of the *Product Item*'s Usage Period to the given number of <days>. Input as integer value greater than or equal to a value of null. Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	/pupa[:]<days>[,<TVB dep.>]
e.g.	/pupr:30 sets the length of an *Usage Period* to 30 days.

Command	**/pupr** - Usage Period, relative
	Extends a Usage Period by the given number of <days>. If the PIO does not exist, yet, a Usage Period of the given length will be added. Input as integer value greater than or equal to a value of null. Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	/pupr[:]<days>[,<TVB dep.>]
e.g.	/pup:30 extends the length of an *Usage Period* by 30 days.

Command	**/puvd** - Universal Data
	Adds, updates or deletes a PIO *Universal Data* of a *Product Item*. Expects the *Universal Data*'s index as an unsigned decimal value as argument.
Syntax	/puvd[:]<index>
e.g.	/puvd:15 add or update a *Universal Data* field in index 15.

Command	**/puvdaes** - Universal Data AES key storage
	Option for the setup of a PIO *Universal Data* of a *Product Item* as storage for AES keys. Expects the key length as argument followed by the specification of the key. It is also possible to initiate a random generation of the key within the *CmContainer* by the specification of the token 'r'(andom).
Syntax	/puvdaes<key length>:<key> /puvdaes<key length>:r Currently the following key lengths are supported: • 256
e.g.	/puvdaes256:r generates a AES random key in the *CmContainer* with the key length 256 bits.

Command	**/puvdrsa** - Universal Data RSA key storage
	Option for the setup of a PIO *Universal Data* of a *Product Items* as storage for RSA keys. Expects the key length as argument followed by the specification of the keys. The keys can be set directly by the specification of exponent, prime p and prime q. It is also possible to initiate a random generation of the keys within the *CmContainer* by the specification of the token 'r'(andom), optionally followed by the exponent to use. Exponent, prime p and prime q have to be specified as hexadecimal data in Little Endian byte order. Each block of hexadecimal data has to be preceded by '0x'.
Syntax	/puvdrsa<key length>:<exponent>,<prime p>,<prime q> /puvdrsa<key length>:r[:<exponent>] Currently the following key lengths are supported: • 4096
e.g.	/puvdrsa4096:r generates a RSA random key in the *CmContainer* with the key length 4096.

Command	**/pwupidata** - WUPI Data
	Adds or removes a sequence of *Hidden Data* PIOs that are used as WUPI data storage. New WUPI data storage either can be filled with the contents of a specified file or preset with a user-defined fill byte. In this case, the size of the WUPI data has to be specified. ext. type: number of the first *HiddenData* field to be programmed (default value of 0).

© Copyright 2018 WIBU-SYSTEMS AG

	`block size`: number of bytes to be stored for each field (`0..256`, default and optimum value is `242`). The default and optimal data section entry length equals 242 bytes which is shorter than the maximum entry length of 256 bytes. Using this default length optimizes hardware resource performance in the *CmContainer*. Reading data is automatically done across entries, i.e. when an entry is completed by the maximum length automatically the next entry is read. `acc. code`: *HiddenData AccessCode* (default is calculated). `size`: number of bytes to be created as WupiData. `fillbyte`: fill bytes used to fill the reserved storage by WupiData.
Syntax	`/pwupidata:[e<ext. type>][,b<block size>][,a<acc. code>]:<file>` — Allocates WUPI Data storage, filled with the contents of file `<file>`. `/pwupidata:s<size>[,e<ext. type>][,b<block size>][,a<acc. code>][,f<fill byte>]` — Allocates `<size>` bytes of WUPI Data storage, initialized with the value `<fill byte>`. `/pwupidata[:s<size>][,e<ext. type>][,b<block size>]-` — Removes the WUPI Data storage.

5.1.1.8 Special Commands

This section describes special options. The following options are available.

Command	**`/bkp`** - Backup File
	Enables the backup file viewer mode. expects a *CodeMeter* backup file. Only allowed in combination with list option `/l`.
Syntax	`/bkp:\"<Backup Datei>\"`
Command	**`/crac`** - Create Remote Activation Context File (*.WibuCmRaC)
	Enables the creation of a Remote Activation Context File (`*.WibuCmRaC`). Optionally the target file or the target directory can be specified. If a Remote Activation Context File (`*.WibuCmRaC`) is specified, the contents of every target *CmContainer* will be stored there. Otherwise, for every target a Remote Activation Context File "`<serial number>.WibuCmRaC`" will be created in the specified target directory or in the current directory if the argument is omitted. This option cannot be used in the Remote Activation mode.
Syntax	`/crac[:<*.WibuCmRaC file>|<*.WibuCmRaC target directory>]`
Command	**`/rcl`** - Cleanup Registry (*CmDongle* only)
	Deletes the *CodeMeter* related Windows registry entries of the chosen categories. The supported *CmDongle* form factors comprise: *CmStick* (c), *CmStick/M* (m), Removable Media (r), Unknown USB devices (u). If omitted all registry entries belonging the categories (c) and (m) will be deleted by default. This command requires administrator privileges. Use this feature with care. The cleanup may have unknown side effects. Cleaning the categories Removable Media or Unknown USB devices may have also the effect that entries not related to *CodeMeter* will be deleted.
Syntax	`/rcl[:cmru]`
Command	**`/log`** - Logging
	Enables logging. Expects the path to a logfile as argument. The optional mode specifier '+' has the effect that the log output will be appended to the log file, otherwise the file's contents will be overwritten.
Syntax	`/log:"<logfile>"` `/log[:]\"<logfile>\"[+]`
Command	**`/?`** - Help
	Issues further help on desired topics.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/?** - Help
	If no topics are specified, the complete help list is issued.
Syntax	/?[<topic>|<option>]
Command	**/v** - Verbose Mode
	Activates the detailed display mode.
Syntax	/v
Command	**/val** - Validation Mode
	Activates the validation mode. In this mode a *CmContainer* returns a confirmation sequence after each successful programming operation. The received data is validated with the *Firm Security Box*. By default, this mode is deactivated increasing the performance. Using this option reactivates the validation routine.
Syntax	/val
Command	**/vslf** - Validation Of Signed Log Files
	Validates either the contents of a given signed *CodeMeter* log file or a sequence of log files located in a given directory. Expects the path to the file that contains the public keys to use for validation and the path to the log file respectively log directory as arguments.
Syntax	/vslf:<public key file>,<log file>|<log directory> The <public key file> musst be created as comma separated file (CSV). For this file the following notation is valid: <major version>,<minor version>, <firm code>, 0x<public key> You can specify several different values and *Public Keys* for separate runtime versions. The required values and *Public Key* can be obtained from the generated *CodeMeter* log files. Please note, that the *Public Key* differs for each *Firm Code* and each *CodeMeter* Runtime version, i.e. you must have access to the the same version as your customers to obtain this information.

e.g.	From the *CodeMeter* log file:
	`2014-06-24T06:06:19 SignedLogfile FirmCode:10, PublicKey:a809304778d517c44a22d65e1fcedd51a4e2a956fa89e93bb1a24e2100000000a2ad17e685306d6e15eb6b7ebc8cc72ebc97c0f52721b584836696de00000000, Runtime-Version:5.20.1432.500, LogfileId:1`
	the following <public key file> is derived:
	`5,20,10,0xa809304778d517c44a22d65e1fcedd51a4e2a956fa89e93bb1a24e2100000000a2ad17e685306d6e15eb6b7ebc8cc72ebc97c0f52721b584836696de00000000`

Programming examples

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /petr30 /puca1492 /pfm0x8000 /ca
```

Adding a *Product Item* with the *Product Code* 2001, a 30 days *Expiration Time*, an *Unit Counter* value of 1492, and the *Feature Map* 0x8000 to the *Firm Item* with the *Firm Code* 6000010 in *CmContainer* 1-1234.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /petr335 /pucr426 /pt: "Text" /cu
```

Updating the *Product Item* with the *Product Code* 2001. The *Expiration Time* is extended by 335 days, and the *Unit Counter* increased by 426 units. In addition, a text is added to the *Product Item*.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /pet /cd
```

Deleting the *Expiration Time* of the *Product Item* 2001.

```
CmBoxPgm /f6000010 /p2008 /ca /laf:1-1234.WibuCmRaC,1-1234.WibuCmRaU,1-1234.WibuCmRaM
```

Adding the *Product Item* 2008 per remote programming. Besides the remote Update File, also a Modified Context File is created allowing for later reprogramming.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /plq5 /ca
```

Setting the *License Quantity* to 5 for *Product Item* 2001 on *Firm Item* 6000010. Sets in *Product Item* 2001 in *Firm Item* 206 the License Quantity to 5.

5.1.2 CodeMeter License Central

Ticket System

Integrating software protection into the software is one but fundamental aspect which strongly affects system security. At the same time, the integration of software protection into sales, production and support processes also determines whether

© Copyright 2018 WIBU-SYSTEMS AG

a system is easy to operate, and thus is accepted by both customers and employees. The latter processes we summarize as Back Office Integration (BOI).

CodeMeter License Central is a ticket system with a standardized graphical user interface to create, manage, and deliver both *CmDongles* and *CmActLicenses*.

A detailed description of *CodeMeter License Central* please find in the manual to be downloaded in the developer area at www.wibu.com.

Editions of *CodeMeter License Central*

CodeMeter License Central is available in two editions:

- *CodeMeter License Central Desktop Edition*
- *CodeMeter License Central Internet Edition*

Both Desktop and Internet Edition are functionally identical, differing only in licensed use, integration, and support services.

The *Desktop Edition* can be used on a single server in your company. The operating system is Linux Ubuntu. The database runs on MySQL only. Access is via a browser-based front end. You get a VM image which meets the requirements and requires only the VMware Workstation or ESX/EXSi server to run.

The *Internet* Edition is designed for distributed installation on multiple servers. You can use an existing database server (MySQL or Microsoft SQL Server; for support for other database platforms please inquire directly). The core of *CodeMeter License Central*, based on an Apache Web Server and Tomcat Server, can be installed on other Linux distributions or Windows if you desire.

Sales Interface

When you program a *CmContainer* for a specific license, you send a related request with an item number to *CodeMeter License Central* and receive back a unique ticket. Since this scenario in most cases involves the selling of this item, we refer to this interface as the *Sales Interface*. The ticket contains the authorization to add the license to a *CmContainer*.

Depot Interface

You decide whether you instantly program the license yourself, deal with it later or transfer the ticket to your customer. If you decide to transfer, then your customer is able to collect the licenses bought at any time, for any *CmContainer*. We call the interface for collecting licenses the *Depot Interface*.

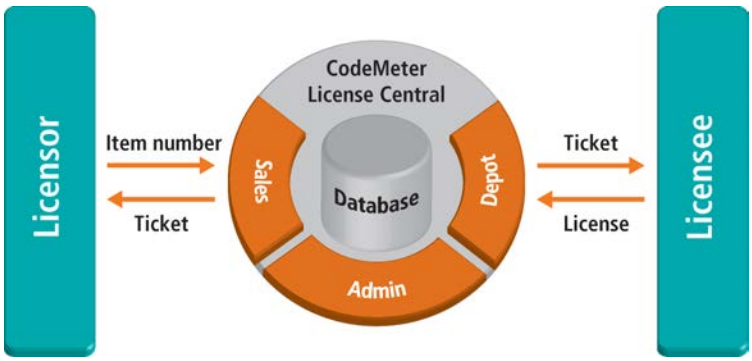


Figure 12: License Collection by Licensee

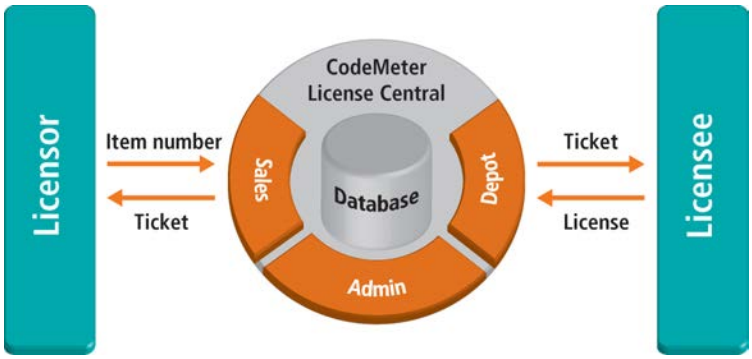


Figure 13: License Collection by Licensor

Admin Interface

Next to the *Depot Interface* and *Sales Interface*, *CodeMeter License Central* features the *Admin Interface*. The *Admin Interface* comprises functions for defining license properties (e.g. *Expiration Time*, *License Quantity*, etc.), for managing access rights, for generating statistics and reports, and for carrying out support activities.

© Copyright 2018 WIBU-SYSTEMS AG

The following figure shows an overview of interfaces and related functions in *CodeMeter License Central*.

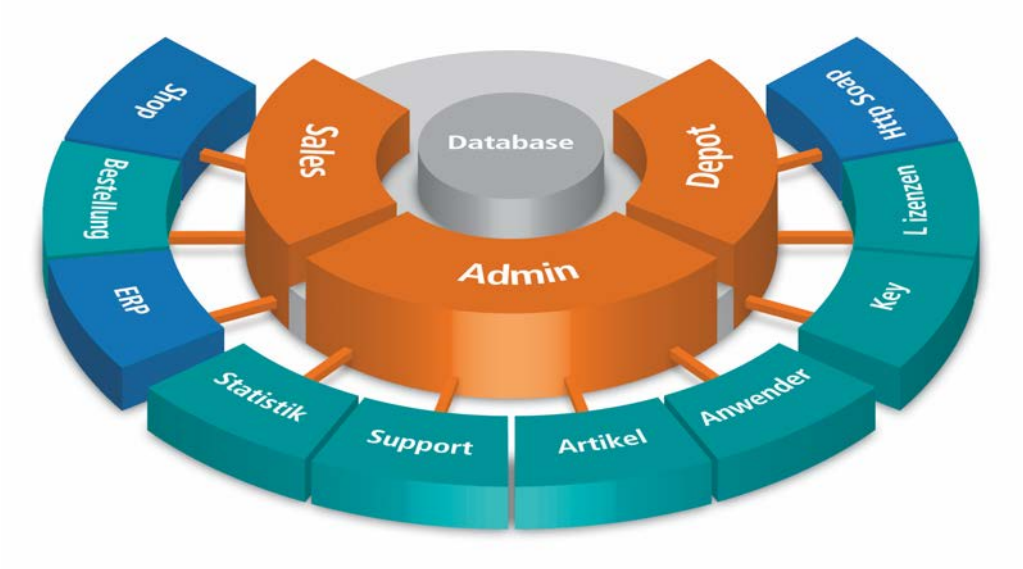


Figure 14: *CodeMeter License Central* - Interfaces and Functions

5.1.3 Programming by File Transfer

***CmDongle*-Licenses**

Remote updating a *CmDongle* requires some information on the *CmDongle* to be reprogrammed. This information is safely stored and transferred in a Context File (license request, `*.WibuCmRaC`).

Context File (`*.WibuCmRaC`) - License Request

The creation of a Context File (`*.WibuCmRaC`) is bound to the physical ownership of a *CmDongle*. On creation the *Firm Code* to be included is specified. Usually, the own *Firm Code* is specified because only the container holding it can be altered. In addition, the file holds the serial number of the *CmDongle*. When you as licensor receive the Context File (`*.WibuCmRaC`) from your licensee, you can see in detail which of your licenses and license options are stored in the *CmDongle*. The licensee generates this file in *CodeMeter Control Center* by the process of the license update.

Update File (`*.WibuCmRaU`) - License Update

On the basis of this Context File (`*.WibuCmRaC`), you as licensor, are able to generate a so-called Update File (`*.WibuCmRaU`), in order to modify existing licenses using the tools *CodeMeter License Editor*, *CmBoxPgm* or *CodeMeter License Central*. The provided options are the same as with physically existing *CmDongle* you can add new or alter existing licenses, e.g. extending *Expiration Time*, or delete licenses. The Update File (`*.WibuCmRaU`) holds the update sequences and is valid only for a specific *CmContainer*. A licensee is able to only one-time import the file into the specified *CmDongle*.

***Firm Update Counter* (FUC)**

After the successful import of the update file by the licensee in *CodeMeter Control Center* a specific counter, i.e. the *Firm Update Counter (FUC)*, at the *Firm Item* level is increased. By increasing the counter a repeated import of the Update File (`*.WibuCmRaU`) is invalid.

This is of special importance, for example, when the Update File (`*.WibuCmRaU`) holds programming commands which add a new license entry, increase a *Unit Counter* by a number of units, or extend an *Expiration Time* for a number of days.

Modified Context File (`*.WibuCmRaM`)

When creating an Update File (`*.WibuCmRaU`) automatically a so-called Modified Context File (`*.WibuCmRaM`) is created providing you with an image of the content your licensee owns when s/he imported the Update File (`*.WibuCmRaU`). In the case of a new update, e.g. license extension, you can either use a new Context File (`*.WibuCmRaC`) sent by the licensee, or you use the current Modified Context File (`*.WibuCmRaM`) as programming basis. Many licensors already in-house-create the Context File (`*.WibuCmRaC`) directly after programming, and can manage the update process without licensee interference.


If, in the meantime, the *CmDongle* has been reprogrammed by another licensor, all files keep valid.

The following figure illustrates this process.

© Copyright 2018 WIBU-SYSTEMS AG

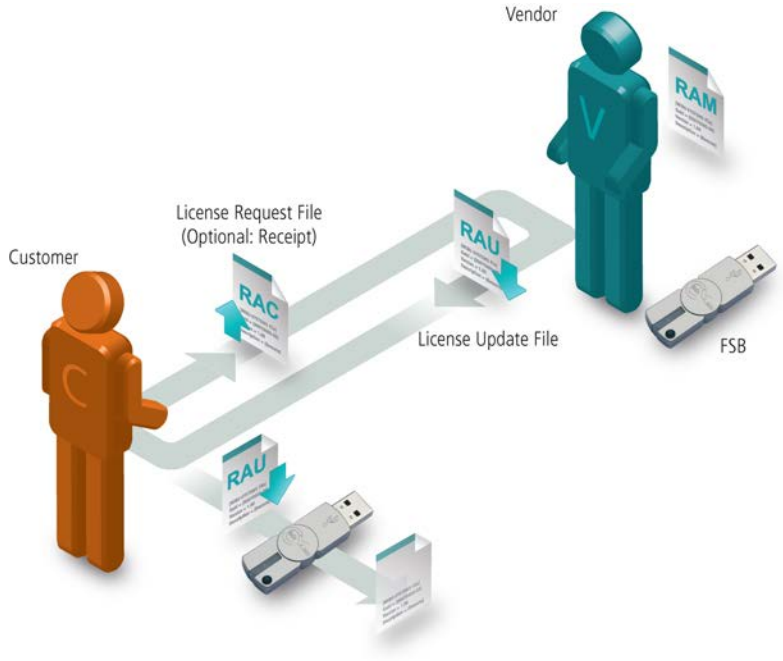


Figure 15: *CmFAS* - File-based Remote Update *CmDongle*

CmActLicense-Licenses

With two exceptions, remote programming of *CmActLicense* licenses largely follows the process for *CmDongles* as described above.

Firstly, before creating the initial Context File (license request, `*.WibuCmRaC`) the customer has to import an empty license container he receives by the vendor. This LIF file (License Information File) in the `*.WibuCmLIF` format holds information on the binding scheme 16 and additional activation options 16 of the *CmActLicense* license which are used to be able to uniquely bind the license to a computer or a device. Required hardware features of a computer or a device are detected and additional activation information transferred. Only on this bases the initial license request file is created. Following, based on this license request file the vendor reprograms this license request file into an Update File (license update, `*.WibuCmRaU`) the customer imports. Starting from this point the file exchange process between customer and vendor is the same for *CmDongle* and *CmActLicense* licenses.

Secondly, currently on reprogramming the context file into a update file a Modified Context File (`*.WibuCmRaM`) is not created.

The following figure illustrates this process.

© Copyright 2018 WIBU-SYSTEMS AG

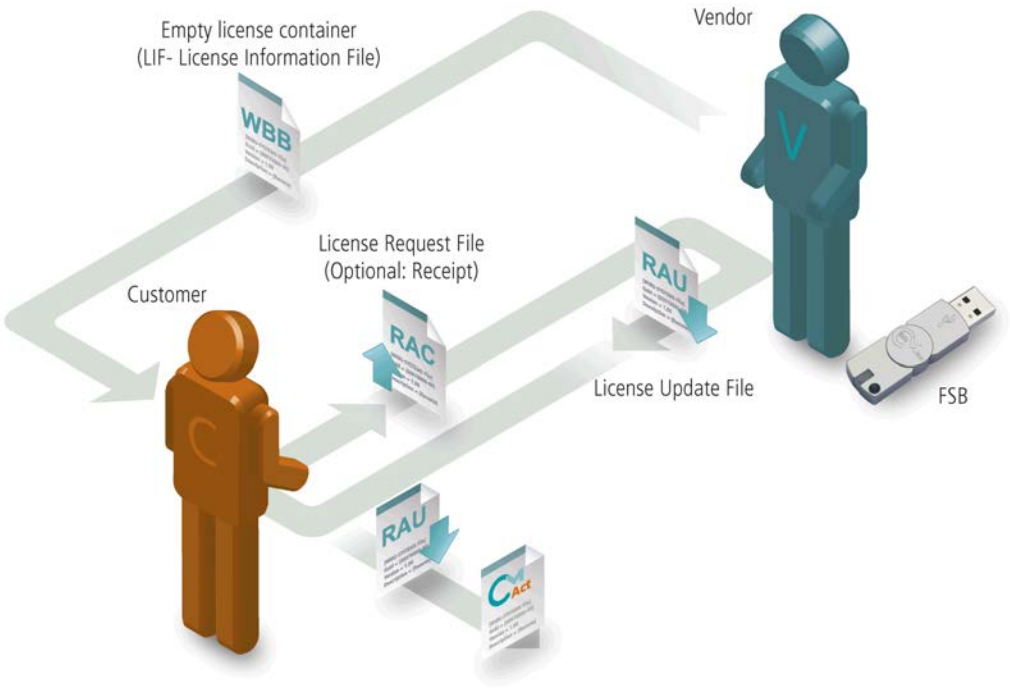


Figure 16: *CmFAS* - File-based Remote Update *CmActLicense*

© Copyright 2018 WIBU-SYSTEMS AG

5.2 Protection: How to encrypt using AxProtector CmE

Please note, that if you encrypt using *AxProtector* with *CmEmbedded* AxEngine for *CmActLicense Universal Firm Code*, you must for security reasons specify a Firmware version 1.14 or higher.
For *CmActLicense Firm Code* you must specify a Firmware version lower than 1.14.

5.2.1 Automatic Software Protection using AxProtector

No Programming Skills required

With *AxProtector* you have a tool at hand that can automatically encrypt already compiled executables. *AxProtector* allows you to integrate *CodeMeter*® into your application - quickly and smoothly - without the need to alter your source code. It is so easy to use, that integration can take place without any programming skills.

In just a few minutes, *AxProtector* encrypts and protects your application for a variety of project types

AxProtector is also available as a commandline variant 73 for Windows 32-bit / 64-bit, .NET, Linux, macOS, and Java applications. Using the *AxProtector* GUI is a simple way to generate a commandline that can be extended and used further to accomplish automatic protection.

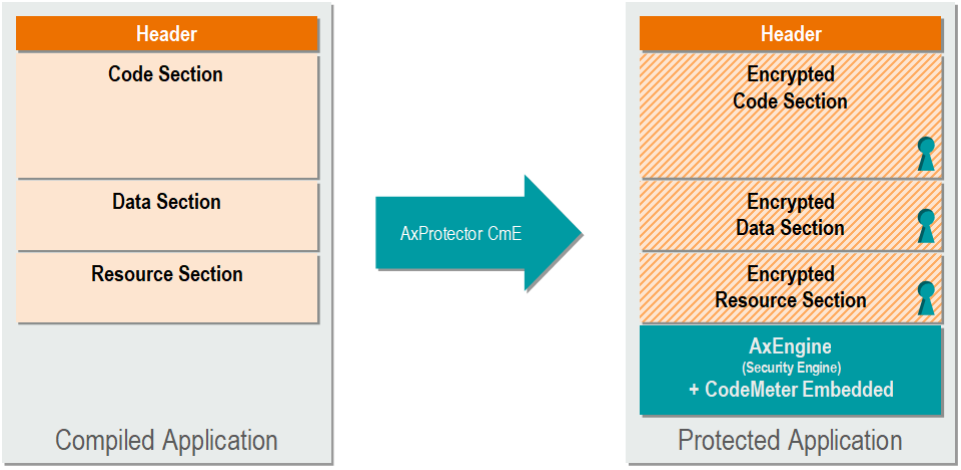


Figure 17: *AxProtector CmE*

The following table summarizes what kind of software applications can be encrpyted using various project types and tools for different operating systems:

Please note that applications protected by using *AxProtector* and then shipped are not identical to the original applications.
This may affect other already applied security measures, e.g. the use of signatures.
In such cases, you must perform other own security measures, e.g. the use of signatures after *AxProtector* has protected the application.

Application to be protected	Project type	GUI Windows	Commandline
Linux, Android	AxProtector Embedded		Windows commandline

AxProtector:

- supports the encryption of all existing *CodeMeter* license options (*Product Item Options*). Thus all necessary license information is integrated into the encryption, for example, network licenses, or license checks at runtime.
- features functions to identify debugger use: in the case a debugger is detected, a *CmContainer* can be locked.
- provides the feature of "on-demand-decryption", i.e. parts of the protected application (source code and resources) are decrypted only when accessed. This "on demand decryption" effectively protects against memory dumping and the extraction of unprotected versions.
- offers the use of freely customizable user message dialogs including the creation of individual texts for purchasing options or errors and also the embedding of company logos.

© Copyright 2018 WIBU-SYSTEMS AG

5.2.2 AxProtector Samples

The following *AxProtector* command line sample show how to use the basic parameter of licensing system (-kcm 74), *Firm Code* / *Product Code* (-f 74/-p 74), CmEmbedded (-xc 73), framework (-cct 79) which have to be considered. For the other options see the *AxProtector* command line options 73.

Android Dongle

```
AxProtectorLin -o:P010 -kcm -f6000010 -p1210 -eurc1 -wu2 -cct:android-so -u:UserMsgDroid
-sl -nl -cav -ccs -xc -car3,1 -v -# helloCounterAndroid
```

Linux32/64, armv7EL Dongle

```
AxProtectorLin -o:P010 -kcm -f6000010 -p1210 -eurc1 -wu2 -xc -u:UserMsgLin -sl -nl -cav
-ccs -car3,1 -v -# helloCounter
```

Linux armv7HF, armv8 CmDongle

```
AxProtectorLin -o:P010 -kcm -f6000010 -p1210 -eurc1 -wu2 -cct:linux-armhf -u:UserMsgLin
-sl -nl -cav -ccs -xc -car3,1 -v -# helloCounter
```

5.2.3 Commandline Options for AxProtector

The commandline application comes in several versions you find in the directory "%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin":

5.2.3.1 Basic Options

Option	-X
valid for	

Links the static library of the licensing system to the application to be protected.

This option is set by default.

Setting this option increases the security compared to linking the dynamic library.

Option	-XC
valid for	

Allows using *AxProtector* Linux x86/x86_64 with *CmEmbedded* instead of *CodeMeter License Server*.
This allows starting with Version 9.11 to protect Linux binaries (architectures ARM / Intel (32-64 bit)).

Please note, that this is supported for *CmDongle* only.

Option	-XCA
valid for	

Allows using *AxProtector* Linux x86/x86_64 with *CmEmbedded* including the programmed CmEmbedded Adapter instead of *CodeMeter License Server*. This allows starting with Version 9.11 to protect Linux binaries (architectures ARM / Intel (32-64 bit)).
A dynamic library 'libCmActAdapter.so' is sought (must locate in the same directory as the binary to protect) and added to the encrypted binary. If the library is not found, an error message informing on the cancelled encryption displays. Then the license access for CmActLicense uses this library.

Please note, that this is supported for *CmActLicense* only.

Option	-A[AES]
valid for	

Specifies the encryption algorithm (*CodeMeter* only).

By default, the AES encryption algorithm in CBC mode is used (Default).

© Copyright 2018 WIBU-SYSTEMS AG

5.2.3.2 Options for the Licensing System

Option	-K([CA][CM][WK])

valid for 

Specifies the licensing system.

Parameter	-KCA

uses *CmActLicense*.

Parameter	-KCM

uses *CmDongle* (Default).

Parameter	-KWK[-x]

uses *WibuKey*. x stands for:

1	uses encryption algorithm 1 (*WibuKey* only).
2	uses encryption algorithm 2 (*WibuKey* only).
3	uses encryption algorithm 3 (*WibuKey* only).
4	uses encryption algorithm 4 (*WibuKey* only) (Default).
5	uses encryption algorithm 5 (*WibuKey* only).

The following options should directly locate after the -K option since they refer to the actual defined licensing system.
When you use both licensing systems for an executable file, the options set are valid for the defined licensing system.

Option	-Fx

valid for 

Specifies the *Firm Code*(x) to be used.
Expects the input of an unsigned integer value <n>.

Option	-Px

valid for 

Specifies the *Product Code* (x) to be used.
Expects the input of an unsigned integer value <n>.

Option	-CFx

valid for 

Specifies the *Feature Code* (x) of the *Feature Map* to be used.

Only valid when using *CodeMeter*.

Expects the input of an unsigned integer value <n>.

Option	-RD([YYYYMonDD[,HH:MM:SS[:<Timezone>]]][:now]) or according ISO-8601 with T(ime) and Z(one) parameter: -RD([<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm or ±hhmm] [±hh]]][:now])

valid for 

defines the *Release Date* for encrypting and decrypting (only *CmDongle* and *CmActLicense* only).
Specification is in format year, month, and optional hours, minutes, seconds, and the timezone
The input of [:now] applies the current date.

Option	-D:v

valid for 

Specifies the minimum driver version (v).
Input of v using (x.y).
Default setting: *CodeMeter* 4.20.

Option	-FW:v

valid for 

Defines the minimum firmware version(v).
Input of v using (x.y).
Default setting: *CodeMeter* 1.0

Option	-S([L][N|W]|C)

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Option	-S([L][N|W]|C)

Specifies the search order of the subsystem when searching for licenses.
The options `N` and `W` may be used alternatively only.

Parameter	-SL

Uses the local subsystem (local).

Parameter	-SN

Uses the network subsystem (network).

Parameter	-SLN

Uses first the local subsystem (local), then the network subsystem (network).

Parameter	-SNL

Uses first the network subsystem (network), then the local subsystem (local).

Parameter	-SW

Uses the Wide Area Network subsystem (WAN).

Parameter	-SLW

Uses first the local subsystem (local), then the Wide Area Network subsystem (WAN).

Parameter	-SWL

Uses first the Wide Area Network subsystem (WAN), then the local subsystem (local).

Parameter	-SC

First searches the local and then the network subsystem, and if a network was found the network drive.

Option	-N[C[A]|L[A]|N|X|X[A]]

valid for 

Specifies the network access.

Parameter	-NC[A]

convenient mode (compatibility mode): here each started instance on the network allocates a normal user limit, and the local access is unlimited (no user limit).

Parameter	-NL[A]

normal user limit: here each started instance allocates a license regardless whether the *CmContainer* is found locally or on the network.

Parameter	-NN

no user limit: here any number of instances can be started locally or on the network. No licenses are allocated.

Parameter	-NNI

aditionally uses the flag "Ignore Linger Time".
This license option allows to define an allocation time of the license after a protected application has been released or finished (more Information in the *CodeMeter* Developer Guide).

Parameter	-NS

station share: here several started applications on a client allocate only a single license.



You use this option when allowing the end-user to start the protected application several times. On terminal server each session allocates a single license. In virtual machines each machine allocates a single license.

Parameter	-NX[A]

exclusive mode: here only a single started instance per PC is allowed. Each access allocates a single license.
If the exclusive access is defined, the automatic runtime check is no longer internally activated. Using option '-car' 78 for the runtime check has now to be set explicitly.



Please note that *AxProtector* Java does not support the option `[A]`.

Option	-SIG(Fx):(Px):(CFx):(SD|HD)DN[:CP]

valid for 

Selects a private key to calculate the signature.

Parameter	Fx

specifies the *Firm Code* (x) used for signature.

© Copyright 2018 WIBU-SYSTEMS AG

Option	-SIG(Fx):(Px):(CFx):(SD|HD)DN[:CP]

Parameter	Px

specifies the *Product Code* (x) used for signature.

Parameter	CFx

specifies the *Feature Code* (x) used for signature.

Parameter	SD

specifies whether a *Secret Data* datafield contains the signature key.

Parameter	HD

specifies whether a *Hidden Data* data field contains the signature key.

Parameter	DN

specifies the number / index of the *Hidden* or *Secret Data field.*

Parameter	CP

specifies the path to the certificate used to sign the executable.

Option	-SIG:PK[,CP]

valid for 

Defining the path to the private key file and the certificate used to sign and check the executable for code integrity.

Parameter	PK

specifies the path to the private key file

Parameter	CP

specifies the path to the certificate used to sign the executable

5.2.3.3 Options for Encrypting and Decrypting

Option	-CA[[A[!]],[Ct[,u]],[D[m]],[E],[G[!,[,1]]],[L],[M],[R[t][,m]],[S[p]],[T[t][,u]],[V],[Z]]

Encrypts the executable file using automatic encryption.

Parameter	-CAA <1>

valid for 

Activates the security options (Advanced Protection Schemes, APS).
<1> covers the options [0, 15]



When applying more than one security option (APS), you can combine the options 1, 2, 4 and 8 with "or".
are mutually exclusive. If both options are set, automatically -CAA8 is applied.

Option	Description
0	No resource encryption is performed.
1	Resource encryption applies (APS 1)
2	Static modification applies (APS 2)
4	Dynamic modification applies (APS 3)
8	Extended static modification applies (APS 4)
	CAA6 applies APS 2 and 3 CAA13 applies APS 1, 4 and 8

Parameter	-CACt<,u>

valid for 

Checks the *CmContainer* system time related to the PC time. A protected application runs only when the PC time in a time window is t minutes younger and, optionally, u minutes older than the *CmContainer* system time.

Parameter	-CAD<m>

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CAD<m>

Specifies the file access mode for the automatic encryption of files which have been encrypted using the option -CD.

0	Decrypts the file's content block by block (4Kb) on demand (on demand).
1	Decrypts the file's complete content at once on first access (at once). Depending on the file size an access may result in a delay.
2	Prevents that on file encryption the protected application is generally able to write data to the hard drive (vie only). Here all writing to a file is prevented.
4	Decrypts the content of very huge files (e.g. large MPG3 files with a size of 500 MB) used with file encryption (read and decrypt on demand). In this mode, by default, writing back is deactivated.

Parameter	-CAE

valid for 

Activates instantly the detection that a has been removed from the PC ('plug-out') (*CmDongle* only).


If the connection to a *CmDongle* should fail or the license cannot be accessed, you can assign a reasonable number of "ignores" allowing the end-user to continue working without a license access.

Parameter	-CAG<1>

valid for 

Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC).
<I> covers the options [0,367]


When applying more than one Anti-Debugging mechanism (ADC), you can combine options with "or".

Option	Description
1	Checks whether a debugger is attached to your application. In the case a debugger is detected, the application does not start (ADC1).
2	Checks additionally for Kernel debugger programs, e.g. "SoftICE". In the case a debugger is detected, the application does not start (ADC2).
4	Checks in an extended search for debugger programs running parallel to your applications. Cracker tools, such as IMPREC are detected. In the case a debugger is detected, the application does not start (ADC3).
8	Checks for all debugger programs. Then no debugger programs are allowed, i.e. also within developer environments (IDE), e.g. VISUAL STUDIO, DELPHI. In the case a debugger is detected, the application does not start (ADC4).
16	Locking of the license entry and thus of the hardware when a debugger program has been detected (ADC5).  Using this option requires that the developer has programmed the *CmContainer* with a *Firm Access Counter* (FAC). The *Firm Access Counter* (FAC) is located at the *Firm Item* level of a *CmContainer*. This counter allows you to control whether a *Firm Item* can be used for encryption and decryption operations. By default, the FAC is deactivated and has a value of 65535 (0xFFFF). It can be programmed to any other value between 1 and 65534. The *CmContainer* is locked when the FAC has a value of 0. Depending on the settings all licenses of a vendor can be locked if a hacker attack occurs. The owner / end-user of the locked *CmContainer* must contact the software vendor for unlocking codes. How and how often unlocking is granted depends on the vendor's policy.
32	Adds a mechanism to the application preventing the attachment of a debugger program to the application at runtime (generic debugger detection) (ADC6).  Please note, that this mechanism of a generic debugger detection is not supported neither for the AxProtector project type 'File Encryption", nor *SmartShelter SDL*.
64	Detects whether the application is to be started in a virtual machine and prevents this (ADC7).
128	Hardware locking is performed only with a valid *Firm Access Counter* (only in combination with ADC5 and *CmContainer*).
256	*Firm Access Counter* decrementing by 1 is performed (only in combination with ADC5 and *CmContainer*).

Parameter	-CAL

valid for 

Limits the automatic encryption to specified areas.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CAM

valid for 

Adds the menu items 'Control' and 'About' to the application's system menu.

Parameter	-CAR<t>,<m>

valid for 

Adds a runtime check to the automatic encryption.
The check occurs every <t> seconds. he default setting is 300 seconds (5 minutes).
<m> specifies how often the end-user is able to ignore a failed check (threshold).

Parameter	-CAS<p>

valid for 

Specifies the size of the protected application to be encrypted. You enter the length, in percent, anywhere from 0 to 100%.
The default setting is 75 percent.

Parameter	-CAT(t)(,u)

valid for 

On each start of the application, a *Certified Time* update is triggered. Then the application starts regardless of whether the update was successful or not, and writes it into the *CmContainer*.
Then the application starts when the time difference between the *Certified Time* and the system's PC-Time is not greater than <t> specified in hours.
<u> specifies the valid time span in hours within which the difference between the *Certified Time* and the system time is allowed to range without a new *Certified Time* update (*CodeMeter* only).

 <t> has to be equal or greater than <u>.

Parameter	-CAV

valid for 

Adds a code integrity check to the automatically encrypted application.
The code integrity check may also cover several executable files / libraries. Then each file is able to check the integrity of all other files.

For this the files must be written into a separate section [CheckCodeIntegrity DLLs] of the *.wbc file. Please export the *.wbc file by using the *AxProtector* GUI menu item "**File | Export**" and change the file meeting your requirements, for Windows e.g.:

```
[CheckCodeIntegrity DLLs]
Image1=draw.exe
Image2=rectangle.dll
Image3=circle.dll
Image4=triangle.dll
```

Moreover, on encrypting a library also a list of applications can be transferred allowed to load these libraries. On loading the library then it is checked whether the process name includes one of the specified names.

If not, an error message displays and subsequently the application closes.

This list of process / application names is also specified in the section [CheckCodeIntegrity DLLs]. Please use the parameter AllowedExe1, AllowedExe2 which are available, e.g.:

```
AllowedExe1=PlainApp.exe
AllowedExe2=ExtendedApp.dll
```

 If the same application name is also specified in the image 1, image2, etc. list above, a code integrity check is also performed for this application.

Option	-CC[[A[a:s],[B],[D],[E],[F],[H],[I],[K][M],[O],[Q],[R],[S],[T],[X]]

Sets compatibility parameters.

Parameter	-CCA

valid for

Defines the target system / subsystem in combination with the option -CCX for .NET executables including debugging of encrypted applications.

<a> contains the target platform [1,2].
1: x86 / Intel 32 Bit
2: ArmV4i

<s> contains the subsystem [9]
9: Windows CE System

Parameter	-CCB

valid for

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CCB
	Disables usage of proprietary (B)ase relocation representation.

Parameter	-CCE
valid for	
	Specifies that the PE is not enlarged.

Parameter	-CCF<number>
valid for	
	Specifies extended protection / special handling using a numeric value. <number> holds numerical value 1: AutoCAD 2011 - ARX files (plug-ins)

Parameter	-CCH
valid for	
	Prevents all global hooking in a protected application.

Parameter	-CCI
valid for	
	Allows to use the protected application in a way that the added protection does not change the eventual existing loading sequence for a protected application file. The import directory of the protected application is not changed. This replaces the option "-ccm" which is no longer required.

Parameter	-CCK
valid for	
	does not explicitly unload DLL, Windows XP only.

Parameter	-CCO
valid for	
	Activates special handling for ActiveX / OCX images.

Parameter	-CCQ
valid for	
	Clears license use for the protected applications not when WM_QUIT is called but with the call of ExitProcess().

Parameter	-CCR
valid for	
	Deactivates the renaming of sections.

Parameter	-CCS
valid for	
	Specifies that all licenses must locate in the same *CmContainer* connected to the same PC as it was the case with the first license found.

Parameter	-CCT:<selector>
valid for	
	allows specifying the AxEngine by using <selector> as selected architecture. The entries cover: linux-armhf, embedded-micro, android-so.  -ctt:linux-armhf: specifies the AxEngine for the linux-armhf architecture (e.g. Raspberry).

Parameter	-CCX
valid for	
	States that also so-called mixed-mode assemblies are protected. This allows to encrypt .NET assemblies which cannot be encrypted using *AxProtector .NET*. Next to Win32 also Win64/x64 mixed-mode assemblies can be protected using the native *AxProtector*. The library wbcor32/64.dll is required to allow running the protected assembly.

Option	-CC[D[flags]],[[K],[S]]
valid for	

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CCD

valid for 

Defines flags for dlopen when loading shared objects. The flags may be `or`'d using the Linux constants:

- `RTLD_LAZY` `0x00001`
- `RTLD_NOW` `0x00002`
- `RTLD_NOLOAD` `0x00004`
- `RTLD_DEEPBIND` `0x00008`
- `RTLD_GLOBAL` `0x00100` (not set means `RTLD_LOCAL`)

Option	-CD[C][H](K([CA][CM][WK]Fx[Py])

valid for

Encrypts a file 1:1 file and adds a header holding encryption information. These files can automatically decrypted by an automatically encrypted application.

Parameter	-CDC

valid for

Applies the file name extension from the `*.wbc` file.

Parameter	-CDH

valid for

Specifies that the license access is kept open when the player closes the file, and a handle is kept open. This option is valid for single, separate files.

Parameter	-CDK([CA]|[CM]|[WK])

valid for

Specifies the used licensing system:
`CA` uses *CmActLicense*
`CM` uses *CmDongle*

Parameter	-CDK([CA]|[CM]|[WK])F

valid for

Specifies the *Firm Code* (`x`) required for the encrypted application to be able to open the encrypted file.

Parameter	-CDK([CA]|[CM]|[WK])P

valid for

Specifies the *Product Code* (`y`) required to open the protected application. The *Firm Code* must be previously set. More than one *Firm Code* - *Product Code* pair can be set.

Option	-CI[H][N][D]

valid for

Encrypts explicitly defined source code fragments within the executable file to be used with *IxProtector*.

Parameter	-CIH

valid for 

Defines that Wupi`XXX` functions functions are NOT dynamically hooked.

Parameter	-CIN

valid for 

Defines that no error messages are displayed when an error occurs.

Parameter	-CIP

valid for

Defines that no pointer based IxProtector table is searched.

Option	-CP<l>

gilt für

Installs a cleanup mechanism deleting all created files and registry entries on exiting an application if it has been started on a *CmContainer*.
`<l>` , causes that all deleted entries are written to a log file into the directory where the application locates.

Option	-E[A(C||R)],[E(C||R)],[F],[M],[T],[U(S(C|R)[n]|R(C|R)[n]|I)])

Defines additional checks while encryption and decryption operations are performed.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-EA

valid for

Activates an *Activation Time* check (*CodeMeter* only) .

- `C` Checks if the Product Item Option *Activation Time* exists.
- `I` Ignores the Product Item Option *Activation Time* (*CodeMeter* only).
- `R` Requires the Product Item Option *Activation Time*.

Parameter	-EE

valid for

Activates an *Expiration Time* check.

- `C` Checks if the Product Item Option *Expiration Time* exists.
- `I` Ignores the Product Item Option *Expiration Time* (*CodeMeter* only).
- `R` Requires the Product Item Option *Expiration Time*.

Parameter	-EF

valid for

Activates the decrement of the *Firm Access Counter* (*CodeMeter* only).

Parameter	-EM

valid for

Activates an *Maintenance Period* check.

- `C` Checks, if the Product Item Option *Maintenance Period* exists.
- `R` Requires the Product Item Option *Maintenance Period*.

Parameter	-ET

valid for

Enforces an *Certified Time* after the *CmContainer* is activated.


This option requires an activated *Expiration Time*.

Parameter	-EU

valid for

Activates an *Unit Counter* check and the counter decrementing by the specified value `<n>`.

- `S` Checks and decrements at the start of the protected application only.
- `R` Checks and decrements on each runtime check.
  The option `R` includes the option `S`.

  For options S and R the following options exist:

  - `C` Checks whether the Product Item Option *Unit Counter* exists (default setting).
  - `R(R)` Requires the Product Item Option *Unit Counter*.
  - `<n>` specifies the decrement. The default setting is `0`.
- `I` Ignores the Product Item Option *Unit Counter* (*CodeMeter* only).


-`eurr2` activates a required *Unit Counter* on each runtime check and decrements it each time by the value of `2`.

Option	-RIDx[,y]

valid for

Specifies the number of RID variants (`x`) and traps (`y`).
If RID=0 is set automatically, then the default value of 256 is used.

Option	-RIDIXx[,y]

valid for

Specifies the number of RID variants (`x`) and traps (`y`) when using *IxProtector* (WUPI).
If RID is set to a value of `0`, automatically the default value (64/8) is applied.

Option	-G[o,l]][:"Marker",]]

valid for

© Copyright 2018 WIBU-SYSTEMS AG

Option	-G[o,l][:"Marker",]]

Excludes the specified range from the encryption.

`<o>` Defines the exclusion at the beginning of the range.

`<l>` Defines the length of the range to be excluded
`"Marker"` identifies the text marker within the source code identifying the beginning of the range to be excluded from the encryption.

Option	-FW

valid for 

Sets the minimum Firmware version in the encryption.

Option	-W[C[t]E[t]][P][U[c]]

valid for 

Specifies the threshold of issued warnings.

Parameter	-WC[t]

valid for 

Specifies the threshold `<t>` in hours for the *Certified Time*.

Parameter	-WE[t]

valid for 

Specifies the threshold <t> in days for the *Expiration Time*.

Parameter	-WP

valid for 

Activates a warning if the *Usage Period* has not yet been activated.

Parameter	-WU[c]

valid for 

Specifies the threshold `<c>` in units for the *Unit Counter*.

5.2.3.4 Runtime Options

Option	-I

valid for

Specifies that the exception handling for plug-in DLL files is used.
This option exclusively works with DLL files. When the plug-in is loaded and no licensing system linked, the plug-in does not closes the complete application, and does not issue error messages.

Option	-L:xx

valid for 

Specifies the language of the user-defined message texts.
`cn`: sets Chinese
`de`: sets German
`fr`: sets French
`jp`: sets Japanese
`us`: sets English (default setting)

Option	-M[A][C][E][I][S][T][U][W[C|P|T|U][L][R][V]]:"msg"

valid for 

Specifies the output text of user messages of the protected application.
`"msg"` holds the string for the desired event.
Line breaks, inverted comma, tabs, etc. can be specified in the output text. Type `"\n"`, `"\r"`, `"\t"`, `"\"` in the string at the desired position.

Parameter	-MA

valid for 

Specifies the application name which is transferred to the server and displayed in *CodeMeter WebAdmin*. No standard name is set. If this option is not set, the internal name of the executable file is used.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-MC

valid for 

Holds the text displayed in the system menu item 'About'.

Parameter	-ME

valid for 

Holds the text, if an error has occurred.

Parameter	-MI

valid for 

Holds the text displayed, if the required driver of the licensing systems is not installed.
The return value to the User Message DLL corresponds to the already existing WUPI error code `wibu::UpiErrorLicenseModuleNotLoaded`.

Parameter	-MS

valid for 

Holds the text displayed while the protected application is started.

Parameter	-MT

valid for 

Holds the text displayed, if the date of the *Expiration Time* has been reached

Parameter	-MU

valid for 

Holds the text displayed, if the *Unit Counter* has reached a value of 0.

Parameter	-MWC

valid for 

Holds the text displayed, if the time difference between *Certified Time* and System Time is too big. This option requires the activated option `-wc<t>`.

Parameter	-MWP

valid for 

Holds the text displayed on application start, if an existing *Usage Period* has not yet been activated.

Parameter	-MWT

valid for 

Holds the text displayed, if the end of the *Expiration Time* or the *Usage Period* has been reached.

Parameter	-MWU

valid for 

Holds the text displayed, if the *Unit Counter* is about to reach a value of 0.

Option	-U[:FileName]

valid for 

Calls the file `UserMsgXX.dll` where XX stands for a country placeholder, e.g. `De`, `Us`, etc..
`[:FileName]`
Calls the specified class for the message handling.
The class must be a secondary class to the `com.wibu.xpm.MessageHandler`. For example, `com.wibu.xpmSwingMessageHandler` as default standard message handler of the runtime package.

Option	-UM[:FileName]

valid for 

Calls the user-defined message assembly UserMsg if this assembly exists.
If `[:FileName]` is specified the implemented message assembly holds the name `<FileName>.dll`.

> Specify the name without a file extension `*.dll`.

Option	-UI

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Option	-UI

Implements the message assembly inline configured by an `*.ini` file.

Option	-UDT[C|A|:"<text>"]

valid for 

Sets user defined text in AxEngine CmAccess calls. If set, overrides user defined text set by a message DLL.

C	Use computer name
A	Use application name
"text"	Use specified text

Option	-O[:FileName]

valid for 

Specifies the path and the name of the encrypted destination file.

5.2.3.5 Operational Options

Option	-!

valid for 

Creates a command file (*.wbc).

Option	-V

valid for 

Activates the verbose mode.

 In the case of  use `-vn`.

Option	-#[File]

valid for 

Prints the logging to the specified `[File]`. This option exists next to automatic output to the AxProtector.log. `[//Documents and Settings\user]`.

 In the case of  on specifying a file name, please note that the relative path information always bases on the directory of the input assembly.

Option	-? or -h

valid for 

Shows the options in commandline mode.

Option	@cmds.wbc

valid for 

Adds a specified `*.wbc` file holding parameters for the automatic encryption of an executable file.

5.3 CodeMeter Core API: Licensed access to applications on the embedded device

CmEmbedded is the runtime counterpart of *CodeMeter License Server* in the embedded realm and allows direct access to licenses securely saved in a *CmContainer* from within your application.

The Application Programming Interface (API) required to perform license-related operations is fully compatible with the *CodeMeter Core API* of the *CodeMeter Software Development Kit*. However, depending on modules implemented only a subset of the functions can be used.

Please refer to the separate documentation of the *CodeMeter Core API* adapted *to CodeMeter Embedded.* Please download the webhelp-based documentation from the Wibu-Systems website.

© Copyright 2018 WIBU-SYSTEMS AG

5.3.1 Code Samples

Here you find some code samples to show how licensed access to applications on the embedded device for selected *CodeMeter Core* API functions.

5.3.1.1 List all connected CmDongles

```
#include <stdio.h>
#include <CodeMeter.h>

int main (int argc, const char * argv[])
{
  HCMSysEntry hEntry;
  CMACCESS2 cmacc;
  CmStaticLibOnInit();
  memset(&cmacc, 0x0, sizeof(CMACCESS2));
  cmacc.mflCtrl = CM_ACCESS_SUBSYSTEM;
  hEntry = CmAccess2(CM_ACCESS_LOCAL, &cmacc);
  if ( hEntry ) {
    CMBOXINFO aBoxInfo[4];
    int nRet, k;
    nRet = CmGetBoxes(hEntry, CM_GB_ALLPORTS, &aBoxInfo[0], 4 );
    nRet = min(nRet, 4);
    for (k=0;k<nRet;k++) {
      printf("- Box %u-%u Ver %i.%02i\n",
             aBoxInfo[k].musBoxMask, aBoxInfo[k].mulSerialNumber,
             aBoxInfo[k].mbMajorVersion, aBoxInfo[k].mbMinorVersion );
    }
    CmRelease(hEntry);
  }
  CmStaticLibOnExit();
  return 0;
}
```

5.3.1.2 Encrypting using CmCrypt2

```
void do_direct_crypt(void)
{
  CMACCESS2 cmAcc;
  HCMSysEntry hcmse;
  int res;
  unsigned int cbDest = 16;
  unsigned char pbDest[16];
  CMCRYPT2 cmCrypt;

  memset(&cmCrypt, 0, sizeof(cmCrypt));
  memset(&cmAcc, 0, sizeof(cmAcc));

  cmAcc.mflCtrl |= CM_ACCESS_NOUSERLIMIT;
  cmAcc.mulFirmCode = 100003;
  cmAcc.mulProductCode = 1;
  hcmse = CmAccess2(CM_ACCESS_LOCAL, &cmAcc);
  if(0 == hcmse) {
    /* No matching license entry or CmContainer found. */
    return;
  }

  cmCrypt.mcmBaseCrypt.mflCtrl |= CM_CRYPT_FIRMKEY;
  cmCrypt.mcmBaseCrypt.mflCtrl |= CM_CRYPT_AES;
  cmCrypt.mcmBaseCrypt.mulEncryptionCodeOptions |= CM_CRYPT_UCIGNORE;
  /* set a random number as Encryption Code (part of the key) */
  cmCrypt.mcmBaseCrypt.mulEncryptionCode = 12345;

  res = CmCrypt2(hcmse, CM_CRYPT_DIRECT_ENC, &cmCrypt, pbDest, cbDest);
  if(0 != res) {
```

© Copyright 2018 WIBU-SYSTEMS AG

```
    /* The number of en-/decrypted bytes was returned. */
  } else {
    /* Some errors occurred. */
  }
  CmRelease(hcmse);
}
```

5.3.1.3 Remote Programming

5.3.1.31 Create a Context File

```
/* Store the content of ctx_buf in an ASCII file with .WibuCmRaC ending. */
size_t create_context(CMUSHORT mask, CMUINT serial, CMUINT firm_code,
                      unsigned char *ctx_buf, size_t max_size)
{
  CMACCESS2 cm_access;
  HCMSysEntry hbox;
  int ret = 0;
  size_t written = 0;
  memset( &cm_access, 0, sizeof(cm_access) );
  cm_access.mflCtrl = CM_ACCESS_Force;
  cm_access.musBoxMask = mask;
  cm_access.mulSerialNumber = serial;
  hbox = CmAccess2( CM_ACCESS_LOCAL, &cm_access );
  if (NULL != hbox) {
    if (CmGetRemoteContextBuffer(hbox, 0, &firm_code, 1,
                                 ctx_buf, max_size, &written) != 1) {
       printf("Create context failed, err=%d\n", CmGetLastErrorCode());
    }
    CmRelease(hbox);
  }
  return written;
}
```

5.3.1.32 Execute a CmContainer Update

```
/* You have to open the WibuCmRaU file and store the content in update_buf before */
int execute_update(unsigned char *update_buf, size_t size)
{
  CMACCESS2 cm_access;
  CMFAS cmfas_res;
  HCMSysEntry hsubsys;
  int ret = 0;
  size_t nresult = 0;
  memset( &cm_access, 0, sizeof(cm_access) );
  cm_access.mflCtrl = CM_ACCESS_SUBSYSTEM;
  hsubsys = CmAccess2( CM_ACCESS_LOCAL, &cm_access );
  if (NULL != hsubsys) {
    if (!CmExecuteRemoteUpdate(hsubsys, CMFAS_CTL_CMD_UPDATE, update_buf, size,
                               0, 0, & cmfas_res, 1, &nresult)) {
      printf("Execute update failed, result=%d\n", cmfas_res.mulResult);
      ret = -1;
    }
    CmRelease(hsubsys);
  }
  return ret;
}
```


© Copyright 2018 WIBU-SYSTEMS AG

5.3.1.4 Access a network license

```
void lan_access(char *ip)
{
  CMACCESS2 cm_access;
  HCMSysEntry hlan;
  int ret = 0;
  memset(&cm_access, 0, sizeof(cm_access));
  cm_access.mulFirmCode = 10;
  cm_access.mulProductCode = 13;
  strncpy(cm_access.mszServerName, ip, sizeof(cm_access.mszServerName)-1);
  hlan = CmAccess2(CM_ACCESS_LAN, &cm_access);
  if (NULL != hlan) {
    /* Connection to server has been established */
    CmRelease(cm_handle);
  }
}
```

© Copyright 2018 WIBU-SYSTEMS AG

6 CmActLicense on embedded devices

This section covers four parts.

First, it is described how to create the Adapter[89] as required for an individual *CmActLicense* binding followed by how to create a CmActLicense license file[98].

Second, you get an overview of how to program licenses and license options[50] into the *CmContainer* using the tools which are part of the installed *CodeMeter* Software Development Kit (SDK) on your development system.

If you later want to use the programmed context / update files for licensed accesses to your application, you must use the *CodeMeter Embedded Core API*[84] to integrate desired function calls into the source code of your application running on the embedded device.

Third, if supported for your embedded environment, *AxProtector CmE*[72] is able to automatically encrypt already compiled executables without the need to alter your source code. This part includes sample command lines, a brief introduction, and available options.

Of course you can also protect your application without *AxProtector CmE*; then use the *CodeMeter Embedded Core API*[137] for integrating protection and security features (e.g. CmCrypt2[138]).

Fourth, the shortly described *CodeMeter Embedded Core API*[137] (including code samples) allows direct access to licenses securely saved in a *CmContainer* from within your application. The complete API documentation is separately available for download from the Wibu-Systems website (https://www.wibu.com/manuals-guides.html).

CmActLicense ***Universal Firm Code*** (UFC) licenses - as compared to *CmActLicense* ***Firm Code*** - do not have a generation counter. This means resetting a license status covers the complete history as long as the license update file (`*.WibuCmRaU`) holds a valid ***Firm Item*** and ***Product Item*** update.

If the ***Firm Item*** update is missing, only the license information file (`*.lif`) is imported. The license itself must be 'restored' by the independent software vendor (ISV).



Please note the following for broken or invalid *CmActLicense* ***Universal Firm Code*** (UFC) licenses:

Broken:

As soon as the binding value changes (on purpose or inadvertently), the license is lost. Since the dynamic part can no longer be decrypted, units, time spans, or saved key material are lost and cannot beeing restored.

Invalid:

If the actual dynamic data and its backup has become invalid (on purpose or inadvertently), here the independent software vendor (ISV) is able to specify the license with the reimport option to allow the end user to import the latest license status.

CmActLicense Adapter

CmActLicense represents the software-based variant of the protection and licensing technology *CodeMeter*.

Here licenses and the keys responsible for encrypting and decrypting are saved to a *CmActLicense* license file which is cryptographically safeguarded and signed.

This virtual *CmContainer* is unique and bound only to a specific device. The unique binding is guaranteed by a digital "finger print" calculated from specific features of a device. This ensures that *CmActLicense* licenses are valid only for the identified device and are not transferable.

On a desktop system, the hardware characteristics are well-known. In contrast, on embedded systems characteristics can be quite different. Thus your help is required to implement some adaptations.

These adapted code objects are called *CmActAdapter* and provide a platform- and/or vendor-specific communication layer for *CmEmbedded*.

CmEmbedded knows four different kinds of adapters, each of which serves a designated purpose.

Binding Extension

The Binding Extension adapter is required to determine the binding value for a *CmActLicense*. The implementation for this task highly depends on the actual product and must always be implemented as an adapter.

Dynamic Data

The Dynamic Data adapter holds a collection of binary data and comprises the dynamic license specific data which change at runtime of the *CodeMeter* protected embedded application. This data must be protected against tampering which is achieved using a special Security Counter. The location where the data is stored may vary drastically depending on the chosen platform, for example, the target system may or may not have any file system.

License Reader

© Copyright 2018 WIBU-SYSTEMS AG

The License Reader adapter is used to load and read the encrypted license files during initialization.

License Writer

The License Writer adapter is required to import new license files or update existing licenses.

6.1 The Adapter - how to create an individual CmActLicense binding



The individual *CmActLicense* binding can either be integrated as library (lib) via linking or must be build from the source code Wibu-Systems deliver according to a NDA (non disclosure agreement).

The *CmActLicense* module manages software-based license container. When the end-user uses the license via *CodeMeter Core API* no difference exist between *CmDongle* and *CmActLicense* licenses.

In contrast to a hardware CmDongle, the *CmActLicense* license container is bound to specific characteristics/features of a target system. This binding is implemented by the so-called *Binding Extension*.

Using the Adapter module *CmEmbedded* provides the option to customize the license container management to the requirements of the target system.

Comparison to the *CodeMeter* Runtime Service

Using the binding schema "Binding Extension" the *CodeMeter* Runtime service provides a mechanism for software vendors to implement a binding of the *CmActLicense* license container by themselves. The *CodeMeter* Runtime service loads a signed dynamic library of the vendor (`.dll`,`.so`,`.dylib`, etc) which holds code to extract the features desired for the binding.

In contrast to the adapter implementation in *CmEmbedded*, the dynamic library only holds a main function for value detection and some helper functions.

The schemes *SmartBind* and *None Bind* are not available.



Starting with Version 2.10 *CmActLicense* now also supports *Universal Firm Codes* (*UFC*).

To be able to handle *UFC CmContainer,* the Adapter implementation requires implementing two additional functions in the *License Writer* 92 section. Then a parallel use of *CmActLicense* UFC and *CmActLicense Firm Codes* is possible.

6.1.1 Compatibility to previous CmEmbedded Versions

Licenses

Generally, all licenses generated with *CmEmbedded* Version 1.5 and higher can still be used with *CmEmbedded* 2.30

Adapter

CmEmbedded Version 2.0 covers changes in the Adapter interface. The `Load` functions of the Binding *Extension* and *Dynamic Data* adapter now holds the *Binding Extension* name and *Firm Code* not as string but as object of the structure `ACT_ADAPTER_LOAD_DESCR`.

The adapter object no longer is a `pointer typedef` but directly the structure `ACT_ADAPTER` is used.

In addition, the *License Writer* adapter has been extended by two functions:
`act_adapter_create_license_by_serial` and `act_adapter_delete_license_by_serial`.



Due to these changes previous adapater implementations are neither source code not binary compatible.

6.1.2 Adapter

The Adapter:

- handles the read and write operations of data from and to *CmActLicense* license container. This data is categorized in two types: static and dynamic.
- allows the binding to specific device features.

© Copyright 2018 WIBU-SYSTEMS AG

The Adapter consists of four main components:

- *License Reader*
  The *License Reader* adapter is exclusively responsible for reading the *CmActLicense* license container.
- *License Writer*
  The *License Writer* adapter is responsible for writing and deleting of license container. A license container is written on importing an update or on de-initializing the *CmEmbedded* library.
- *Binding Extension*
  The *Binding Extension* adapter handles the reading of system features and the generation of the binding value.
- *Dynamic Data*
  The *Dynamic Data* adapter manages the dynamic data of all license container. Dynamic data cover encrypted binary data which must be persistently saved, e.g. Flash, EEPROM, file. It holds, among other things, *Unit Counter*, *Usage Period* and an current time stamp.

The *CmEmbedded* library contains a function-less adapter stub able to be replaced by the own implementation.

6.1.3 Configuration

The configuration flag `ENABLE_CMACT_ADAPTER_CUSTOM` in the file `cm_config.h` must be set to `ON`.

6.1.4 General

The single adapter *License Reader*, *License Writer*, *Binding Extension* and *Dynamic Data* should be implemented in separate `*.c` files. Each adapter implementation has to provide the following general interface.

```
WibuBool act_adapter_init(struct ACT_ADAPTER *adapter);
WibuBool act_adapter_deinit(struct ACT_ADAPTER * adapter);
WibuBool act_adapter_unload(struct ACT_ADAPTER **adapter_ref);
```

The functions must not be defined again. They already exist in the general adapter module (`act_adapter.c`). The actual functionality is internally provided by a function pointer. This requires setting in the adapter objects the following function pointer:

- `f_init`
- `f_deinit`
- `f_unload`

`f_init`

The `init` function provides the option to initialize internal resources the adapter object has to hold. If no initialization is required, the function pointer can be set to `act_adapter_generic_init`.

`f_deinit`

The `deinit` function provides the option to release internal resources held by the adapter object. If no de-initialization is required, the function pointer can be set to `act_adapter_generic_deinit`.

`f_unload`

The `unload` function can also be used as the `deinit` function to provide the option to release resources of the adapter object. If no resources are to be released, a stub function of the following form can be used:

```
static WibuBool unload(struct ACT_ADAPTER * adapter)
{
  adapter->last_result_code = ACT_ADAPTER_SUCCESS;
  return TRUE;
}
```

Load functions

Each adapter requires a specific load function. The load functions are the only linker dependencies to be fulfilled, The remaining functionality is - as already mentioned in the general part - implemented using function pointer.

```
struct ACT_ADAPTER * act_adapter_license_reader_load_impl(void);
struct ACT_ADAPTER * act_adapter_license_writer_load_impl(void);
struct ACT_ADAPTER * act_adapter_binding_ext_load_impl(struct ACT_ADAPTER_LOAD_DESCR const *load_descr);
struct ACT_ADAPTER * act_adapter_dynamic_data_load_impl(struct ACT_ADAPTER_LOAD_DESCR const *load_descr);
```


© Copyright 2018 WIBU-SYSTEMS AG

6.1.5 Error Codes

The adapter object saves the error which occurred last to the attribute `last_result_code`. A list of all possible error codes locates in the `act_adapter.h` header file. In the function to be implemented on success the error code `ACT_ADAPTER_SUCCESS` should be set, otherwise a corresponding error code from the list.

6.1.6 License Reader

The *License Reader* adapter is exclusively responsible for reading the *CmActLicense* license container.

```
size_t act_adapter_get_number_of_licenses(struct ACT_ADAPTER *adapter);
size_t act_adapter_get_number_of_licenses(struct ACT_ADAPTER *adapter);

const void * act_adapter_get_license_at(struct ACT_ADAPTER *adapter,
                                        size_t idx,
                                        char* opt_name_ref,
                                        size_t* license_size_in_bytes_ref);

WibuBool act_adapter_close_read_license(struct ACT_ADAPTER *adapter,
                                        const void *license_ptr);
```

The adapter object provides the option to save context-related information of the *License Reader* in the pointer `u.license_reader.license_reader_data`.

The listed functions represent the interface to *CmEmbedded*. Since internally function pointer are allowed to be used, the functions in the single adapter cannot hold the same names. The definitions in the example functions in the following sections do not contain the prefix `act_adapter_`.

Number of license container

The function `act_adapter_get_number_of_licenses` is used by *CmEmbedded* to obtain the number of license container managed by the *License Reader* adapter. Since *CmEmbedded* accesses to license container content via index, the function must create a relation index to content. This may, for example, be done by using an array of file names, flash or RAM addresses.

```
static size_t get_number_of_licenses(struct ACT_ADAPTER *adapter)
{
  size_t num_of_licenses = 0;

  /* look for license container in the file system or at specific flash, RAM addresses */

  return num_of_licenses;
}
```

Read license container data

Using the function `act_adapter_get_license_at` *CmEmbedded* gets a pointer to the content of the license container and its size. If memory has been allocated to the license container content, a consecutive implementation of `act_adapter_close_read_license` must re-allocate (release) the memory.

```
static const void *get_license_at(struct ACT_ADAPTER *adapter, size_t idx,
                                  char *opt_name_ref, size_t *license_size)
{
  void *buffer = NULL;

  /* get and store the address of the license container data in the buffer pointer */

  if (license_size != NULL) {
    *license_size = /* set size of the license container data */;
  }

  if (opt_name_ref != NULL) {
    /* copy license container name to opt_name_ref */
  }

  return buffer;
}
```

The example above provides a template for the access to the license data. Next to the pointer and the size also the name of the license must be copied to `opt_name_ref`. In the case of a file-based implementation, the license name equals the file name.

© Copyright 2018 WIBU-SYSTEMS AG

```
static WibuBool close_read_license(struct ACT_ADAPTER *adapter, const void *license)
{
  WibuBool freed = FALSE;

  adapter->last_result_code = ACT_ADAPTER_SUCCESS;

  /* lookup license pointer and free it if necessary */

  return freed;
}
```

Load

```
struct ACT_ADAPTER *act_adapter_license_reader_load_impl(void)
{
  struct ACT_ADAPTER *adapter = NULL;

  adapter = /* allocate dynamic or static memory for adapter object */;

  if (adapter != NULL) {
    adapter->type = ACT_ADAPTER_READ_LICENSES;
    adapter->is_dyn_allocated = /* depending on how the adapter object was allocated */;
    adapter->f_init = act_adapter_generic_init;
    adapter->f_deinit = act_adapter_generic_deinit;
    adapter->f_unload = /* put pointer to a custom unload function */;

    adapter->u.license_reader.f_get_number_of_licenses = get_number_of_licenses;
    adapter->u.license_reader.f_get_license_at = get_license_at;
    adapter->u.license_reader.f_close_read_license = close_read_license;

    adapter->u.license_reader.license_reader_data = /* store license reader specific data */;
  }

  return adapter;
}
```

6.1.7 License Writer

The *License Writer* adapter is responsible for writing and deleting of license container. A license container is written on importing an update or on de-initializing the *CmEmbedded* library.

```
WibuBool act_adapter_create_license(struct ACT_ADAPTER *adapter, const char *name);
WibuBool act_adapter_create_license_by_serial(struct ACT_ADAPTER *adapter,
                                              uint16_t mask, uint32_t serial,
                                              char const* ext);
WibuBool act_adapter_delete_license(struct ACT_ADAPTER *adapter, const char *name);
WibuBool act_adapter_delete_license_by_serial(struct ACT_ADAPTER *adapter,
                                              uint16_t mask, uint32_t serial);
WibuBool act_adapter_write_license(struct ACT_ADAPTER *adapter,
                                   const void *content, size_t content_size);
WibuBool act_adapter_close_write_license(struct ACT_ADAPTER *adapter);
```

Creating a license container

Depending on the used licensing system, i.e. *CmActLicense Firm Code* (range 5,000.000-5,999,999) and/or *CmActLicense Universal Firm Code* (UFC), a single or both `create` functions must be implemented.

The function `act_adapter_create_license` is called up by the *CmActLicense Firm Code* module. The transferred name is a SHA1 hash over the data of the *CmActLicense* license container with the file extension `.wbb`.

The function `act_adapter_create_license_by_serial` is called up by the *CmActLicense Universal Firm Code* (UFC) module.

Depending on the saving mode, either a file is created, a flash sector or single pages are written or an address array is written.

In order to later guarantee using the correct container when later writing the license container data, e.g. import of an update, internally a handle or descriptor to this created license container must be stored in the adapter object, For this reason, the adapter structure contains the following attribute `u.license_writer.license_writer_data`.

© Copyright 2018 WIBU-SYSTEMS AG

```
static WibuBool create_license(struct ACT_ADAPTER *adapter, const char *name)
{
  WibuBool success = FALSE;

  /* create a license for example as file or prepare flash pages/sectors */

  return success;
}
```

The `name` parameter points to an unique license name, e.g. `3ABA0E5FB3B74A3227BD475B8A96600C58FA3E5D.wbb`. The function must be implemented on using *CmActLicense Firm Code* licenses.

```
static WibuBool create_by_serial(struct ACT_ADAPTER *adapter, uint16_t mask,
                                 uint32_t serial, char const *ext)
{
  WibuBool success = FALSE;

  /* create a license for example as file or prepare flash pages/sector */

  return success;
}
```

Here the license is identified by mask and serial number. Via `ext` parameter *CmEmbedded* transfers a file extension - a feature currently not yet used. This function must be implemented only for *CmActLicense Universal Firm Code* (UFC) licenses.

Delete license container

Also on deleting the difference between *CmActLicense Firm Code* and *CmActLicense Universal Firm Code* (UFC) exists. The *CmActLicense* Firm Code module uses the function `act_adapter_delete_license`, the *CmActLicense Universal Firm Code* (UFC) module uses the function `act_adapter_delete_license_by_serial`.

Depending on the saving mode, either a file, a flash sector or single pages or an address array are deleted.

```
static WibuBool delete_license(struct ACT_ADAPTER *adapter, const char *name)
{
  WibuBool success = FALSE;

  /* delete a license for example from file system or erase flash pages/sectors */

  return success;
}
```

The `name` parameter points to an unique license name, e.g. `3ABA0E5FB3B74A3227BD475B8A96600C58FA3E5D.wbb`. The function must be implemented on using *CmActLicense Firm Code* licenses.

```
static WibuBool delete_by_serial(struct ACT_ADAPTER *adapter, uint16_t mask,
                                 uint32_t serial)
{
  WibuBool success = FALSE;

  /* delete a license for example from file system or erase flash pages/sectors */

  return FALSE;
}
```

Here the license is identified by mask and serial number. Via `ext` parameter *CmEmbedded* transfers a file extension - a feature currently not yet used. This function must be implemented only for *CmActLicense Universal Firm Code* (UFC) licenses.

Writing license container content

CmEmbedded using the function `act_adapter_write_license` writes the content of a license container in a file, flash page or memory addresses. The following example holds the data to be written and their size. In order to detect the correct license container the attribute `.license_writer.license_writer_data` of the adapter object is used.

```
static WibuBool write_license(struct ACT_ADAPTER *adapter, const void *content,
                              size_t content_size)
{
  WibuBool success = FALSE;

  /* write license container content to a file, flash pages or to an address range */

  return success;
}
```

Load

© Copyright 2018 WIBU-SYSTEMS AG

```
struct ACT_ADAPTER *act_adapter_license_writer_load_impl(void)
{
  struct ACT_ADAPTER *adapter = NULL;

  adapter =  /* allocate dynamic or static memory for adapter object */;

  if (adapter != NULL) {
    adapter->type = ACT_ADAPTER_WRITE_LICENSES;
    adapter->is_dyn_allocated =  /* depending on how the adapter object was allocated */;
    adapter->f_init = act_adapter_generic_init;
    adapter->f_deinit = act_adapter_generic_deinit;
    adapter->f_unload = unload;

    adapter->u.license_writer.f_delete_license = delete_license;
    adapter->u.license_writer.f_delete_license_by_serial = delete_by_serial;
    adapter->u.license_writer.f_create_license = create_license;
    adapter->u.license_writer.f_create_license_by_serial = create_by_serial;
    adapter->u.license_writer.f_write_license = write_license;
    adapter->u.license_writer.f_close_write_license = close_license;

    adapter->u.license_writer.license_writer_data = /* */;
  }

  return adapter;
}
```

6.1.8 Binding Extension

The *Binding Extension* adapter handles the reading of system features and the generation of the binding value. The adapter must implement the following interface functions:

```
WibuBool act_adapter_get_binding_value(struct ACT_ADAPTER *adapter,
                                       uint8_t *binding_value_ptr);
```

'None' binding schema

It was mentioned that the none-bind schema is not supported. However, the option exists to bind to a single fix characteristic obtaining a device-independent license.

Reading the binding value

Please note to use characteristics supporting sufficient entropy. For example, consecutive serial numbers as a characteristics are particularly unsuitable. The characteristics are not to be directly copied into the maximum 32 byte array, but should be hashed using SHA256. Then the result can be transferred to *CmEmbedded*.

The following example shows a simple implementation of the characteristics detection.

```
static WibuBool get_binding(struct ACT_ADAPTER * adapter, uint8_t* binding_value_ptr)
{
  uint8_t digest[32];
  struct C_SHA256 sha_ctx;

  c_sha256_init(&sha_ctx);

  c_sha256_update(&sha_ctx, feature, num_of_bytes);

  /* add some more values */

  c_sha256_final(digest, &sha_ctx);

  memcpy(binding_value_ptr, digest, sizeof(digest));

  return TRUE;
}
```

The load function must check whether the combination of *Firm Code* and *Binding Extension* name is supported by the adapter and set the respective function pointer.

© Copyright 2018 WIBU-SYSTEMS AG

```
struct ACT_ADAPTER *act_adapter_binding_ext_load_impl(struct ACT_ADAPTER_LOAD_DESCR const *load_descr)
{
  struct ACT_ADAPTER *adapter = NULL;

  /* check if there is an adapter available for firm code and plugin name combination */
  if (binding_ext_data_adapter_available(load_descr)) {
    adapter = /* allocate dynamic or static memory for adapter object */;

    if (adapter != NULL) {
      adapter->type = ACT_ADAPTER_GET_BINDING_VALUE;
      adapter->is_dyn_allocated = /* depending on how the adapter object was allocated */;
      adapter->f_init = act_adapter_generic_init;
      adapter->f_deinit = act_adapter_generic_deinit;
      adapter->f_unload = /* put pointer to a custom unload function */;
      adapter->u.binding_ext.f_get_binding_value = get_binding;
    }
  }

  return adapter;
}
```

6.1.9 Dynamic Data

The *Dynamic Data* adapter manages the dynamic data of all license container. Dynamic data cover encrypted binary data which must be persistently saved, e.g. Flash, EEPROM file. It holds, among other things, *Unit Counter*, *Usage Period*, and a current time stamp.

```
size_t act_adapter_get_security_counter(struct ACT_ADAPTER *adapter,
                                        uint16_t mask, uint32_t license_serial);
WibuBool act_adapter_increment_security_counter(struct ACT_ADAPTER * adapter,
                                                uint16_t mask, uint32_t license_serial);


WibuBool act_adapter_write_dynamic_data(struct ACT_ADAPTER * adapter,
                                        uint16_t mask,
                                        uint32_t license_serial,
                                        int slot, const void* data_ptr, size_t data_size);

size_t act_adapter_get_dynamic_data_size(struct ACT_ADAPTER * adapter,
                                         uint16_t mask,
                                         uint32_t license_serial,
                                         int slot);

size_t act_adapter_read_dynamic_data(struct ACT_ADAPTER * adapter,
                                     uint16_t mask,
                                     uint32_t license_serial,
                                     int slot,
                                     void* dyn_data,
                                     size_t dyn_data_size);

WibuBool act_adapter_is_dynamic_data_available(struct ACT_ADAPTER * adapter,
                                               uint16_t mask, uint32_t serial, int slot);
```

Each license container requires two locations to which the data is written (see Toggle Mechanism). If the dynamic data is written to files, Wibu-Systems recommends to manage two files for each license container.

Toggle Mechanism

The toggle mechanism alternately writes the dynamic date to different locations. This makes always available the current and the previous state of the license. If, for example, the actual dynamic data is damaged (on purpose or inadvertently), a fall-back on the backup is possible. Although the current state is lost, the license can be further used and must not be re-activated. The toggle mechanism bases on s special counter, the *Security Counter* 97 that has to be protected against unauthorized write accesses.

Loading

The *CmActLicense* implementation requires only the load function of the adapter as fix dependency. The other functionalities are used via function pointer.

© Copyright 2018 WIBU-SYSTEMS AG

```
struct ACT_ADAPTER* act_adapter_dynamic_data_load_impl(struct ACT_ADAPTER_LOAD_DESCR const *load_descr)
{
  struct ACT_ADAPTER *adapter = NULL;

  /* check if there is an adapter available for Firm Code and plugin name combination */
  if (dynamic_data_adapter_available(load_descr)) {
    adapter = /* allocate dynamic or static memory for adapter object */;

    if (adapter != NULL) {
      adapter->type = ACT_ADAPTER_DYNAMIC_DATA;
      adapter->is_dyn_allocated = /* depending on how the adapter object was allocated */;
      adapter->f_init = act_adapter_generic_init;
      adapter->f_deinit = act_adapter_generic_deinit;
      adapter->f_unload = unload;
      adapter->u.dynamic_data.f_dynamic_data_available = dynamic_data_available;
      adapter->u.dynamic_data.f_get_security_counter = get_security_counter;
      adapter->u.dynamic_data.f_increment_security_counter = increment_security_counter;
      adapter->u.dynamic_data.f_write_dynamic_data = write_dynamic_data;
      adapter->u.dynamic_data.f_get_dynamic_data_size = get_dynamic_data_size;
      adapter->u.dynamic_data.f_read_dynamic_data = read_dynamic_data;
    }
  }

  return adapter;
}
```

Check, if dynamic data is available

CmEmbedded checks using the function `act_adapter_is_dynamic_data_available`, if the dynamic data of a license exist.

```
static WibuBool dynamic_data_available(struct ACT_ADAPTER *adapter,
                                       uint16_t mask, uint32_t serial, int slot)
{
  WibuBool ret = FALSE;

  /* */

  return ret;
}
```

The respective function pointer adapter `u.dynamic_data.dynamic_data_available` of the adapter object must point to the address of the `dynamic_data_available` implementation.

Writing the dynamic data

For writing the dynamic data *CmEmbedded* uses the function `act_adapter_write_dynamic_data`.

The respective function pointer adapter `u.dynamic_data.f_write_dynamic_data` of the adapter object must point to the address of the `write_dynamic_data` implementation. The function pointer must be set in the function `act_adapter_dynamic_data_load_impl`.

```
static WibuBool write_dynamic_data(struct ACT_ADAPTER *adapter, uint16_t mask,
                                   uint32_t license_serial, int slot,
                                   const void *data_ptr, size_t data_size)
{
  WibuBool ret = FALSE;

  /* store dynamic data for example to flash, NVRAM or file system */

  return ret;
}
```

The function gets a matching dynamic data adapter object, the serial number and mask of the license, the slot number (0 or 1), and a pointer to the data including the buffer size.

Reading the dynamic data

For reading the dynamic data a function with the following signature is executed.

© Copyright 2018 WIBU-SYSTEMS AG

```
static size_t read_dynamic_data(struct ACT_ADAPTER *adapter, uint16_t mask,
                                uint32_t license_serial, int slot, void *buffer,
                                size_t buffer_size)
{
  size_t size = 0;

  /* read dynamic data for example from flash, NVRAM or file system */

  return size;
}
```

The parameter are identical to the writing function. However, here the dynamic data must be written to the transferred buffer.

In order to avoid unnecessary write accesses to the dynamic file, on size calculation of the dynamic data *CmEmbedded* does not use the reading function but requires the implementation of the following function.

```
static size_t get_dynamic_data_size(struct ACT_ADAPTER *adapter, uint16_t mask,
                                    uint32_t license_serial, int slot)
{
  size_t size = 0;

  /* determine dynamic data size for a specific license in slot */

  return return size;
}
```

The respective function pointer `u.dynamic_data.read_dynamic_data` and `u.dynamic_data.get_dynamic_data_size` of the adapter object must be set to the address of the `read_dynamic_data` and `get_dynamic_data_size` implementations.

Writing and reading the Security Counter

Using the *Security Counter* value defines in which slot the current dynamic data locates (toggle mechanism 95). *CmEmbedded* used the function `act_adapter_get_security_counter` to read the current state and the function `act_adapter_increment_security_counter` to increase the *Security Counter* value by a value of 1.

The *Security Counter* value must be read before writing the dynamic data in order to detect the correct slot. Subsequent to an update of the dynamic data the *Security Counter* value is incremented by a value of 1.

```
static size_t get_security_counter(struct ACT_ADAPTER * adapter, uint16_t mask,
                                   uint32_t license_serial)
{
  size_t cnt = 0;

  /* lookup security counter for specified license and return it */

  return cnt;
}
static WibuBool increment_security_counter(struct ACT_ADAPTER * adapter,
                                           uint16_t mask, uint32_t license_serial)
{
  WibuBool ret = FALSE;

  /* lookup and increment the security counter of one specific license */

  return ret;
}
```

The respective function pointer `u.dynamic_data.f_get_security_counter` and `u.dynamic_data.f_increment_security_counter` of the adapter object must be set to the addresses of both above implementations.

6.1.10 Build

The four adapter are implemented in separate `*.c` files. These are translated by the used compiler to object files. In the subsequent linking step they are, additionally to the linker, transferred to the *CmEmbedded* library (`libwibucmembedded.a` or `wibucmembedded.lib`). Since the linker overwrites the adapter stub of the library, now the self-created adapter stubs are used.

© Copyright 2018 WIBU-SYSTEMS AG

6.2 Create CmActLicense license file

In order to create a *CmActLicense* license file, please proceed as follows:

1. Make sure you installed the *CodeMeter* SDK on the development system.
2. Make sure you connected the *Firm Security Box* (FSB) for *CmActLicense* the development system.
3. Import the `CmFirm.wbc` file holding the *Firm Code* information, e.g. *Firm Code* `6000010`, and the file `CmActFI.wbc` using the following *cmu* commandline:

   `cmu32.exe -i -f CmFirm.wbc` and `cmu32.exe -i -f CmAct[firm code].wbc`.
   `[firm code]` stands for your *Firm Code*.
4. Create an empty license container (LIF) on the development system using the following *CmBoxPgm* commandline:

   > If you already know the 'binding' value of your device, you alternatively have the option to use the *CmBoxPgm* command `"/lbind"`. Then you can skip steps **4** through **7**.
   >
   > `CmBoxPgm.exe -f<firm code> -p<product code> -ca -laf:My CmActLicense.WibuCmRaU -lpn:"My CmActLicense" -lpid:0001 -lbind:<Binding Value) -lfs:cus:<Plugin Name>PlugIn -los:win,emb`
   >
   > Make sure you use the same `<Plugin Name>` as specified in the binding extension[94] and for the dynamic data[95].
   >
   > This command creates a `*.WibuCmRaU` file without the need to create a LIF and a `*.WibuCmRaC` file before.

   `CmBoxPgm.exe -f<firm code> -p<product code> -ca -lif:MyLicense.wbb -lpn:"License LIF" -lpid:0001 -lfs:cus:<Plugin Name>PlugIn`.
   Make sure you use the same `<Plugin Name>` as specified in the binding extension[94] and for the dynamic data[95].
   The `<pluginname>` can hold up to 31 valid characters spanning 'A'..'Z', 'a'..'z', '0'..'9', '_'.

   As result you get the file `MyLicense.wbb`.
5. Import the license file `License.wbb` directly into the embedded system by using the *CodeMeter Core API* call ***CmExecuteRemoteUpdate***.
6. On the embedded system generate a Context File (license request, `*.WibuCmRaC`) from the empty license container (LIF) using the *CodeMeter Core API* call ***CmGetRemoteContextBuffer***.

   As result you hold the file `MyLicense.WibuCmRaC`.
7. On the development system create the Update File `MyLicense.WibuCmRaU` on basis of the `MyLicense.WibuCmRaC` Context File.

   Now create the license and add all required license options using the following *CmBoxPgm* commandline:
   `CmBoxPgm.exe -f<firm code> -ca -p<product code> -ca -laf:MyLicense.WibuCmRaC, MyLicense.WibuCmRaU -lpn:"My CmActLicense" -lpid:0001 -lfs:cus:<Plugin Name>PlugIn -los:win,emb`.
   Make sure you use the same `<Plugin Name>` as specified in the binding extension[94] and for the dynamic data[95].

   As a result now you hold the usable Update File `MyLicense.WibuCmRaU`.
8. On your embedded system import the Update File `MyLicense.WibuCmRaU` directly by using the *Core API* call ***CmExecuteRemoteUpdate***.

6.3 Licensing: Programming of CmContainer

The functionality of *CmContainers* for *CmEmbedded* is compatible to the desktop version of *CodeMeter*. For detailled information of programming and managing licenses please refer to the documentation of the *CodeMeter Software Development Kit*.

After you protected an application, you have several options to program *CmContainer* you want to deliver.

Basically, the programming of license information (*Firm Code*, *Product Code*, and *Product Item Options*) into *CmContainer* is accomplished by three methods:

- **local**: programming of locally connected *CmContainer* using a locally connected *Firm Security Box* (FSB).
- **file-based**: reprogramming of a Context File (license request) (`*.WibuCmRaC`) send by the licensee to the licensor into a Update File (license update) (`*.WibuCmRaU`) and the subsequent import by the licensee into his/her *CmContainer*.
- **protocol-based (SOAP)**: programming and managing of Context and Update Files (`*.WibuCmRaC` and `*.WibuCmRaU`) is done by the Internet supported network protocol SOAP (Simple Object Access Protocol) using *CodeMeter License Central*.

© Copyright 2018 WIBU-SYSTEMS AG

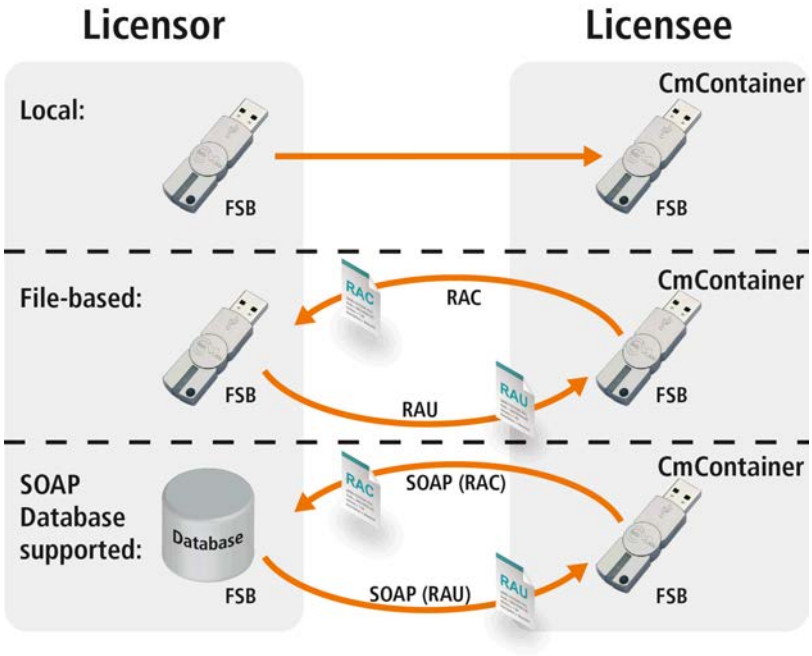


Figure 18: *CmContainer* Programing Options

For all three methods *CodeMeter* provides several tools:

- *CmBoxPgm* commandline tool for batch programming of *CmContainer* in production.
- *CodeMeter License Editor* graphic tool to program *CmDongles* or local tests of licensing strategies.
- *CodeMeter License Central* database-supported tool to create, manage, and deliver licenses using SOAP in a *Desktop* and *Internet* edition.

The tools *CmBoxPgm*, *CodeMeter License Editor*, and *CodeMeter License Central* you can use for file-based remote programming[69], i.e. *CodeMeter Field Activation Service (CmFAS)*.

Most of the tools base on the *CodeMeter Programming API* (HIP - *High Level Programming Interface*). This class-oriented interface allows you to access any object or process required to program or organize license entries in a *CmContainer* and features extended customizing.

The *Programming API* is available for many programming languages. Existing help programs have been generated for respective interfaces, for example, Delphi, Visual Basic, .NET, and Java. For more detailed information on the *Programming API* open the start menu item **"Start | All Programs | CodeMeter | Documentation | Programming-API"** (Press "Windows" key to open Start screen | Type "Programming API" | Press "Enter" key).

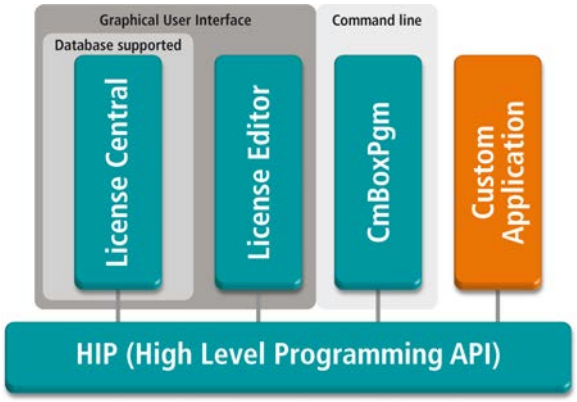


Figure 19: Tools for *CmContainer* Programming

© Copyright 2018 WIBU-SYSTEMS AG

6.3.1 CmBoxPgm

Besides programming of *CmDongle* using *CodeMeter License Central*[67], *CodeMeter* also provides the option for local programming of *CmContainer* using a commandline (console).

The local programming of *CmContainer* requires and uses up *CodeMeter* transactions. Please note that a *Unit Counter* in you FSB is decremented each time you locally program a *CmContainer*.

Advantages of the console

Commandline programming bears the special advantage to use scripts and batch files. Efficiently supported by a variety of parameters you are able to program processes, and apply them to several *CmContainer* in one go.

Application

Such advantages are essential, in particular, when you mass produce *CmContainer* or automate test processes.

Open CmBoxPgm

Open *CmBoxPgm* commandline via: "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**". *CmBoxPgm* opens in the user directory path.

Call start menu item **"Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt"**.

Press "Windows" key to open Start screen | Type "CodeMeter Command Prompt" | Press "Enter" key.

CmBoxPgm opens in the user directory path.

6.3.1.1 Using CmBoxPgm

By default, you find *CmBoxPgm* in form of the executable file `cmboxpgm.exe` in the directory "`%\Program Files%\CodeMeter\DevKit\bin`". For other operating systems find *CmBoxPgm* at the customary locations.

In the following description of options you can alternatively use the prefix '–' instead of '/'.

6.3.1.2 Basic Commands

This section describes the main commands of *CmBoxPgm*.

A basic option always concludes a command sequence which targets a *Firm Item* or *Product Item*.

If you receive error code `3912` or `9010` on programming (`No license profile available`), the complete `CmFirm.wbc` file is missing required for programming or it does not contain the proper, i.e. your `Firm Code`.
Thus import the `CmFirm.wbc` file you received from Wibu-Systems when you became customer. Most easy via drag & drop using *CodeMeter Control Center*
If the `CmFirm.wbc` is no longer available to you, please contact Wibu-Systems Support.

The following options are available.

`/ca` - Add

Add a new entry into the *CmContainer* (*Firm Item* or *Product Item*).

`/cau` - Add/Update

Update an existing entry in the *CmContainer* (*Firm Item* or *Product Item*) or adding a new entry if it does not exist yet.

`/cu` - Update

Update an existing entry in the *CmContainer* (*Firm Item* or *Product Item*).

When adding a *Firm Item* starting with CodeMeter version 4.50 the following is valid:
If on adding a `Firm Item` the text of the *Firm Item* is not explicitely defined, the text for the *Firm Code* defined in the file `CmFirm.wbc` is used. By default the text attribute for *CmDongle* is set to "`Text=Test Kit Firm Code`" and for *CmActLicense* to "`Text=CmAct Testkit`".
If you want to change this text you have two options: either by explicitely setting[55] the *Firm Item Text* or to edit the text attribute in the file `CmFirm.wbc`. You find the file `CmFirm.wbc` in the directory "`%ProgramData%\CodeMeter\DevKit`".

`/cd` - Delete

Deleting an existing entry from the *CmContainer* (*Firm Item*, *Product Item* or *Product Item Options*).

© Copyright 2018 WIBU-SYSTEMS AG

/cdx - Delete if
Delete an entry from the *CmContainer* if available (*Firm Item* or *Product Item*).
/l - List
List the contents of selected *CmContainer*, *Firm Items*, *Product Items* or *Product Item Options*.

Programming examples

CmBoxPgm /l
Lists the content of the first *CmContainer* which is not a *Firm Security Box*. Corresponds to `CmBoxPgm /qn1 /l`
CmBoxPgm /qb1 /l
List the content of a *CmContainer* with an index o. It does not make a difference whether the *CmContainer* holds an *Firm Security Box* entry or not. Corresponds to `CmBoxPgm /qn1:f /l`
CmBoxPgm.exe /qs1-1234 /l
List the content of a *CmContainer* with the serial number of `1-1234`.
CmBoxPgm.exe /qn2,4:f /l
List the content of the *CmContainer*in the index range between 2 and 4 including the *Firm Security Boxes*.

6.3.1.3 CmActLicense Options

This section describes operations related to the licensing system *CmActLicense* and/or *Universal Firm Code* licenses.

The following options are available:

Command	**/lac** – *CmActLicense* License Activation Code
	Use this option to calculate a *CmActLicense* activation code for activation by phone. Expects an installation identifier as argument.
Syntax	`/lac:<installation ID>`

Command	**/lacids** – Allowed CmAct Ids
	Use this option to specify which `CmAct ID` a *CmActLicense* license activation request file may have. Expects a comma-separated list of `CmAct ID`s as argument. *CmActLicense* license activation request files with a `CmAct ID` that does not match one of these `CmAct ID`s will be rejected. This option only can be used in combination with /laf[101] and a Universal *CmActLicense* as target.
Syntax	`/lacids:<CmAct ID1>[,<CmAct ID 2>[,<CmAct ID3>]]...`

Command	**/laf** – *Universal Firm Code* License Update File
	Use this option to set the path to the input file to use and the path to the update file to create. In the case of a *Universal Firm Code* license, the path to a modified license context file may be specified as third argument. In combination with option /lfs[102] `:none`, the input file argument may be omitted. If option /lbind[101] is used, the input file argument must be omitted. The modified license context file argument must be omitted, if the option /lpo:noramfile[103] is set. Valid input files for *Universal Firm Code* licenses are: • *Universal Firm Code* license Context Files (`*.WibuCmRaC`) • *Universal Firm Code* license Modified Cntext Files (`*.WibuCmRaM`)

Syntax	`/laf:\"<input file>\"[,\"<update file>\"[,\"<modified context file>\"]]`
	`/laf:\"<update file>\"[,\"<modified context file>\"]`
	`/laf:["<update file>"]`

Command	**/lbind** – *CmActLicense* Binding Value
	Use this option to set a binding value if the binding value of the end-user PC is known. Expects a sequence of 32 bytes in hexadecimal notation preceded by `0x` as argument. In the case of a *Universal Firm Code* license this argument may be followed by device ID specification, given as a sequence of 64 bytes in the same notation.This option is only supported in combination with the binding mode /lfs:cus[102] and /laf[101] . The request file argument of option /laf must be omitted.
Syntax	`/lbind:0x<hex data>[,0x<hex data>]`

Command	**/ldf** – Display of *CmActLicense* License File
	Use this option to display the contents of a *CmActLicense* license file.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/ldf** – Display of *CmActLicense* License File
Syntax	`/ldf:<file>`
Command	**/ldi** – Display of *CmActLicense*-Installation ID
	Use this option to display information about a *CmActLicense* installation identifier.
Syntax	`/ldi:<installation ID>`
Command	**/lfs** – *CmActLicense* Binding Scheme (License Feature Set)
	Use this option to set the *CmActLicense* Binding Schemes (License Feature Set). ***CodeMeter SmartBind*** *CodeMeter SmartBind* optimizes assuring the validity of licenses, in the case of changing hardware properties of the PC to which the licenses are bound. Wibu-Systems recommends to use this option. In justified instances using *CodeMeter SmartBind* also allows to set a tolerance level. It defines the allowed variation between the initial hardware configuration of the PC when the license was activated the first time and the current configuration. You are able to select one of the following tolerance levels: `1` (=tight), `2` (=medium), or `3` (=loose). By default, *CodeMeter SmartBind* uses the tolerance level 2. If you like to change this setting please contact Wibu-Systems Support before you do so. also supports Binding Schemes which relate to fix or configurable hardware properties of the PC. Wibu-Systems recommends to contact Wibu-Systems Support before you do so. **Syntax**: `/lfs:smart[:<tolerance level>]` ***CmActLicense* with *SmartBind* for licenses in a VM (Virtual Machine)** The behavior of *CmActLicense* with the binding scheme *SmartBind* for licenses in a VM is defined as follows: • If the VM is copied. i.e. the "I copied it" option has been selected, the license breaks. • If the VM is moved, i.e. the "I moved it" option has been selected, then the license remains intact in case of the same CPU types. However, if the CPU types differs, the license also breaks except the tolerance level has been set to a value of `"3"` (loose). **Fix Hardware Properties** Use the following four basic fix hardware properties which can be combined to create the Binding Scheme. Use the optional parameter `<count>`o define how restrictive the scheme is to be, i.e. how many properties need to remain unchanged.

Hardware Property	Description
`'b'`	(B)IOS
`'c'`	(C)PU
`'d'`	(d)isk
`'n'`	(n)etwork adapter

Syntax:
`/lfs:[b][c][d][n][:<count>]`
Configurable Hardware Properties
Use one of the following other configurable hardware properties which cannot be combined.

Binding Scheme	Description
`'non'`	(non)e – no hardware binding
`'cus'`	(cus)tom plugin; expects the plugin's name (up to 31 characters) as argument, valid characters are 'A'

Syntax:
`/lfs:non|ser|cus:<plugin name>`

Command	**/lif** – *CmActLicense* License Information File (`*.WibuCmLIF`) (file-based activation)
	Use this option to set the path to the *CmActLicense* license information file.
Syntax	`/lif:"<license information file>"`
Command	**/lip** – *CmActLicense* License Information File (activation by phone)
	Use this option to set the path to the *CmActLicense* license information file.
Syntax	`/lip:"<license information file>"`
Command	**/lmrt** – Minimum required *CodeMeter* Runtime version

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/lip** – *CmActLicense* License Information File (activation by phone)
	Use this option to specify the minimum *CodeMeter* Runtime version that is required for using *CmActLicense*. As argument a major and minor version number is expected, e.g. '4.50'. The most recent version supported by *CmActLicense* is 4.50. For universal licenses (license transfer) additionally build and revision number can be specified.
Syntax	`/lmrt:<major version>.<minor version>` `/lmrt:<major version>.<minor version>[.build version>[.<revision>]]`

Command	**/lopt** – *CmActLicense* License Options
	Use this option to specify *CmActLicense* license options.
	Valid license option identifiers are:

Flag	Description
`'vm'`	*CmActLicense* license can be used on a (V)irtual (M)achine)
`'reimport'`	*CmActLicense* activation file can be reimported any time.

Syntax	`/lopt:<license option>[,<license option>]`

Command	**/los** – *CmActLicense License* Target Operating System
	Use this option to specify on which operating system(s) the *CmActLicense* license can be used. This option is mandatory. The following operating systems are supported:

Operating System	Description
`'Win'`	(Win)dows, all supported Windows versions, Windows 2000 or higher
`'Mac'`	(Mac) OS X
`'Lin'`	(Lin)ux
`'Emb'`	(Emb)edded devices

Syntax:	`/los:<OS version>`

Command	**/lpid** – *CmActLicense* ID (CmAct ID)
	Use this option to set the license's `CmAct ID` and, if using activation by phone, the `Telephone ID`. As `CmAct ID` argument a combination of four ASCII characters is expected. Please note that in case of *Universal Firm Code* licenses additional rules apply: • The `CmAct ID` must begin with the character `'2'` as the identifier of this specific license type. • The other characters either must be letters or numbers. As `Telephone ID` argument an unsigned number is expected.
Syntax	`/lpid:<CmAct ID>[-<Telephone ID>]`

	`/lpid:<CmAct ID>[-<Telephone ID>]`
	Examples:
	`- CmAct ID = '2ABC'; Telephone ID = 123`
	`/lpid:2ABC-123`
	`- CmAct ID = '2ABC'; Telephone ID omitted (= 0)`
	`/lpid:2ABC`

Command	**/lpo** – *Universal Firm Code* License Programming Options
	Use this option to set *Universal Firm Code* license programming options. - `ctupdate`: allow update of the *Firm Security Box*'s (FSB) *Certified Time* - `noramfile`: disable creation of a modified context file
Syntax	`/lpo:<prog. option>[,<prog. option>]`

Programming examples

How do I program and update a *CmActLicense* license?

Creating and activating a PC-bound *CmActLicense* covers the following single steps:

- Creating a binding scheme
  A binding scheme defines the hardware characteristics of a PC used for the binding. Here *CodeMeter* provided *SmartBind* as an easy and at the same time secure way to uniquely bind licenses to a PC.
- Importing an empty 'virtual' *CmContainer* by the end user.

© Copyright 2018 WIBU-SYSTEMS AG

- Detecting the actual hardware characteristics of the PC using a digital "fingerprint" and transferring to the ISV using a license request file.
- Programming of licenses for this *CmContainer* by the ISV and sending a license update file to the end user.
- Transferring the binding and activation information via import of the license update file by the end user.
- Sending of a receipt of the activation process from the end user to the ISV.


These single steps are automized in *CodeMeter License Central*. The following description refers to using *cmu32 CmBoxPgm*.


Please note that you require at least a *CodeMeter License Server* runtime environment of Version 6.10. No firmware version requirements exist for the *Firm Security Box* (FSB).

1. Create a `*.WibuCmLiF`-Datei (License Information File).
   This file corresponds to an empty license container however holds specifications on binding schemes and additional activation options to be used for unique binding of a license to the computer or the device. By importing the empty license container the customer two things happen. Firstly, the necessary information on the computer or the device are detected and, secondly, the basis for binding the license using a unique, digital "finger print" is prepared.
   Type the following example commandline:

```
cmboxpgm -f6000010 -lif:"UFCLIF.WibuCmLIF" -lfs:smart -lpn:"Universal Firm Code" -
lpid:2000 -v
```

Parameter	Description
`f`	Sets the *Firm Code*: here the evaluation *Universal Firm Code*
`lif`	Sets the license information file
`lfs`	Sets the binding scheme *SmartBind*
`lpn`	Sets CmActLicense name
`lpid`	Sets the CmActLicenseID (here 2000 for the *CmContainer Type CmActLicense;* `2xxxx` stands for *CmContainer Type CmActLicense*, e.g. `1xxxx` for *CmDongle*, see here*)*
`v`	Activates the verbose mode

2. Import the `*.WibuCmLiF` file onto the desired system.
   Type the following example commandline:

```
cmu32 --import --file UFCLIF.WibuCmLIF
```

   The return holds the assigned serial number of the *CmActLicense CmContainer*.

```
cmu32 - CodeMeter Universal Support Tool.
Version 6.30Beta (Level 1) of 2016-May-04 (Build 2215) for Win32
Copyright (C) 2007-2016 by WIBU-SYSTEMS AG. All rights reserved.
The file contains 1 Update:
  CmActLicense update file (Universal): FirmCode 0.
Execute Update ...
  CmActLicense update file: Serial number 130-1781635890, FirmCode 6000010.
   --> successful
1 successful update done
```

3. Create the Context File (license request, `*.WibuCmRaC`).
   Type the following example commandline:

```
cmu32 --create-lt-context --lt-request-file "Context.WibuCmRaC" -s130-1781635890 --
firmcode 6000010
```


Use here the serial number you obtained in step **2**.

4. Program a license and create the Update File (license update, `*.WibuCmRaU`) based on the Context File (license request, `*.WibuCmRaC`).
   Type the following example commandline:


Using *CmBoxPgm* next to the features License Transfer 61 and Module Items 62 introduced with version 6.00 you may also use all other Product Item Options 56 to map your desired license strategy.

© Copyright 2018 WIBU-SYSTEMS AG

```
CmBoxPgm -qs130-1781635890 -f6000010 -ft:"Universal Firm Code - Programming" -ca -
p1234 -pt:"Product Code - Universal Firm Code" -ca -
laf:"Context.WibuCmRaC","Universal_FC_Programming.WibuCmRaU"
```

Parameter	Description
qs	Addresses the *CmContainer*
f	Sets the *Firm Code*: here the evaluation *Universal Firm Code*
ft	Sets the *Firm Code* text
ca	Adds the programmed entry
p	Sets the *Product Code*
pt	Sets the *Product Code* text
ca	Adds the programmed entry
laf	Creates the specified License Activation File on basis of the specified license request Context File.

5. Import the Update File (license update, `*.WibuCmRaU`) into the *CmActLicense CmContainer*.
Type the following example commandline:

```
cmu32 --import-lt-update --lt-fsb --lt-update-file Universal_FC_Programming.WibuCmRaU
-s130-1781635890
```

Now you have - as *CodeMeter WebAdmin* shows - successfully programmed and imported a *Universal Firm Code* license.



How do I program a Trial License?

In order to create a *CmActLicense* Trial License[16], please proceed as follows:

1. Open *CmBoxPgm* commandline via: "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**".
CmBoxPgm opens in the user directory path.
2. Insert the following commandline. Please note not to transfer hyphens or line breaks into the commandline!

```
cmboxpgm /f6000010 /ft:"MyCompany" /cau /p2002 /pup90 /ca /laf:"UpdateTrialLicense.Wi
buCmRaU" /lpn:"Trial CmActLicense" /lfs:none /lpid:2000
```

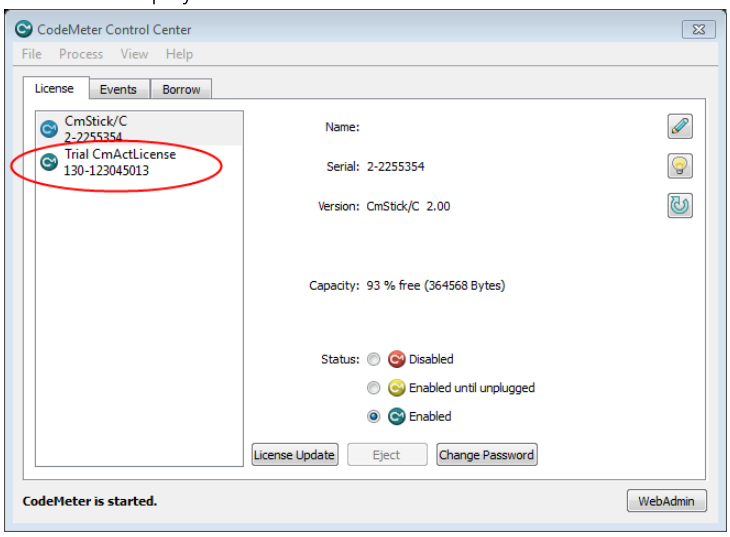
Description
A *CmActLicense* license container with a `F`(irm Code) `6000010` and a `f`(irm Item)`t`(ext) `"MyFirma"` is updated (`/cau`) and a `p`(roduct Code) `2002` added with a usage period (`pup`) of 90 days (`/ca`). The additional *CmActLicense* options comprise the license activation file (`/laf`) `"UpdateTrialLicense.WibuCmRaU"` with the *CmActLicense* name (`/lpn`) `"Trial CmActLicense"` covering a binding scheme (`/lfs`) `"None"` and a CmActLicenseID (`/lpid`) of `2000`. Optionally, you may also allow the use on virtual machines (`/lopt:vm`) or alternatively to the usage period set an absolute expiration time (`/peta`) less than 90 days. A Trial License cannot be updated and imported only once, i.e. the option `/reimport` is not allowed to be set.
Output

© Copyright 2018 WIBU-SYSTEMS AG

A message like the one below is returned indicating the successful creation and the license activation file `UpdateTrialLicense.WibuCmRaU` is created in the user directory (`%Users%\`).

```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 130-1781635890, FC=6000010
*** Add Product Item, CmContainer 130-1781635890, FC=6000010, PC=2002
```

3. Open *CodeMeter Control Center*.
4. Import license activation file.
   Either by drag&drop onto *CodeMeter Control Center* or via menu item "**File | Import License**".
   The license displays in *CodeMeter Control Center*.



How do I program a Protection Only License?

In order to create a *CmActLicense* Protection Only[16] License, please proceed as follows:

1. Open *CmBoxPgm* commandline via: "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**".
   CmBoxPgm opens in the user directory path.
2. Insert the following commandline. Please note not to transfer hyphens or line breaks into the commandline! !

```
cmboxpgm /f6000010 /ft:"MyCompany" /cau /p2002 /ca /laf:"Update-ProtectionOnly-
License.WibuCmRaU" /lpn:"Protection Only CmActLicense" /lfs:none /lpid:2000
```

Description

A *CmActLicense* license container with a `F`(irm Code) `f6000010` and a `f`(irm Item)`t`(ext) "`MyFirma`" is updated (`/cau`) and a `p`(roduct Code) `2002` added (`/ca`).
The additional *CmActLicense* options comprise the license activation file (`/laf`) "`UpdateProtectionOnlyLicense.WibuCmRaU`" with the *CmActLicense* name (`/lpn`) "`Protection Only CmActLicense`" covering a binding scheme (`/lfs`) "`None`" and a CmActLicenseID (`/lpid`) of `2000`.
Optionally, you may also allow the use on virtual machines (`/lopt:vm`) and any number of file reimport (`/reimport`).

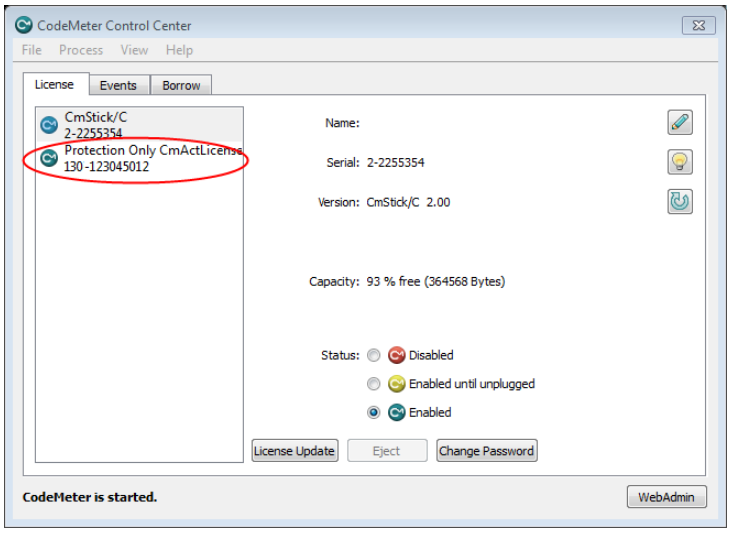
Output

A message like the one below is returned indicating the successful creation and the license activation file `Update-ProtectionOnly-License.WibuCmRaU` is created in the user directory (`%Users%\`).

```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 130-1781635890, FC=6000010
*** Add Product Item, CmContainer 130-1781635890, FC=6000010, PC=2002
```

3. Open *CodeMeter Control Center*.
4. Import license activation file.
   Either by drag&drop onto *CodeMeter Control Center* or via menu item "**File | Import License**".
   The license displays in *CodeMeter Control Center*.

© Copyright 2018 WIBU-SYSTEMS AG



6.3.1.4 Commandline Syntax

Option Blocks

The general syntax in *CmBoxPgm* follows the pattern of so-called option blocks. Option blocks summarize single programming sequences or lists of commands. At the same time, target specification and options are included.

The pattern of an option block is as follows:

```
<target declaration>  <target-specific options>  <operation>
```

Target Declaration

The initial part of an option block includes the required information on the target of an operation. Such target can include:

- single *CmContainer* or a selection of *CmContainer*,
- single *Firm Items*,
- *Product Items*

The syntax of the target declaration corresponds to the hierarchical structure of entries in a *CmContainer* and is ordered from the general to the specific.

Addressing a *Product Item* with the specification of the related *CmContainer* or an array of *CmContainer*, continues with the specification of the *Firm Codes*, the *Firm Item* which holds the *Product Item*, and ends with the specification of the *Product Item*. The typing effort is reduced because parts of the target declaration do not have to be repeated when already specified in a previous option block. If you add a series of *Product Items* to the same *Firm Item*, it is sufficient to one-time specify the *Firm Item* at the beginning of a programming sequence for the *Product Items*.

Target-specific Options

The middle part of an option block holds the target-specific options. Depending on the operation, that part can be or should be left blank.

Operation

The concluding part holds the specification of the operation to be executed.

Specifying the concluding part is mandatory

The most important operations correspond to the basic options and comprise the adding, updating and deletion of *Firm Items and Product Items*. Moreover, the contents of selected *Items* or complete *CmContainer* can be listed in the commandline.

For the time reference used while programming, the following time zones are valid:

Abbreviation	Description
CET	Central European Time
CST	Central Standard Time
EET	Eastern European Time
EST	Eastern Standard Time
MST	Mountain Standard Time
PST	Pacific Standard Time
UTC	Universal Time Coordinated

© Copyright 2018 WIBU-SYSTEMS AG

Table 6: Time Zones in *CmBoxPgm*


Month specifications follow the pattern: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

6.3.1.5 Firm Item Options

This section describes the various options related to *Firm Item*.

Firm Item commands are structured the following way:

```
f<Firm Code> [<Firm Item Options>] <Main Command>
```

The following options are available.

Command	`/f` - Firm Code
	Defines the *Firm Codes* to be used. Expects an unsigned decimal value as argument.
Syntax	`/f<value>`
Command	**`/fac`** - Firm Access Counter (FAC)
	Sets the *Firm Access Counters* to the specified value. Expects an unsigned decimal or a hexadecimal value, preceded by `0x` as argument.  The default setting is `0xffff`.
Syntax	`/fac<value>`
	Sets the given *Firm Item* License Transfer Permissions. Expects as arguments either the token `none` to deny any permissions or the token `pull` to allow pull operations, i.e. the active collection of transfer licenses.
Syntax	`/fltperm:none|pull`
Command	**`/fpta`** - Firm Precise Time, absolute
	Sets the *Firm Precise Time* to the specified absolute value. Expects a date optionally followed by a time and the time zone as argument.  If the time zone is omitted, the system's time zone is used instead.
Syntax	`/fpta<YYYY><Month><DD>[,<SS>:<MM>:<SS>[PST|MST|CST|EST|UTC|CET|EET]]`
	`/fpta2006Dec31,23:59:59UTC`
Command	**`/fptr`** - Firm Precise Time, relative
	Adds the specified number of days to the current value of the *Firm Precise Time*. If the *Firm Item* doesn't exist yet, the current system time plus the specified offset will be set as *Firm Precise Time*. Expects an integer value greater than or equal to zero as argument.  If this *Firm Item* has not yet been created, the system time plus the specified number is used as *Firm Precise Time*. For example, `/fptr1` corresponds to 1 day immediately starting.
Syntax	`/fptr<number of days>`
Command	**`/ft`** - Firm Item Text
	Sets the *Firm Item Text*. Expects a character string (up to 256 characters) enclosed in double quote characters as argument.
Syntax	`/ft:"<Text>"`
Command	**/fuc** - Firm Update Counter
	Sets the *Firm Update Counter* to the specified value. Expects an unsigned decimal value as argument.  This counter automatically increases while programming entries.
Syntax	`/fuc<wert>`

Programming examples

`CmBoxPgm /qn1,4 /f206 /ft:"My Company" /ca`	Adds a new *Firm Item* with the *Firm Code* 206 to the *CmContainer* within the index range from 1 to 4. *Firm Security Boxes* are excluded. The *Firm Precise Time* is set to the current system time. The *Firm Item Text* corresponds to the

© Copyright 2018 WIBU-SYSTEMS AG

	string specification "My Company". *Update Counter* and *Access Counter* are set to default values.
`CmBoxPgm /qb2 /f206 /fuc42 /fac0x1066 /cu`	Updates the *Firm Item* with the *Firm Code* 206 in the second *CmContainer*. The *Firm Item Update Counter* is set to a value of 42 and the *Firm Item Access Counter* set to a value of `0x1066`.
`CmBoxPgm /qs1-1234 /l /f206 /cu /l`	Lists the content of the *CmContainer*, updates the *Firm Item* with the *Firm Code* 206, and subsequently relists the content.
`CmBoxPgm /f206 /cd`	Deletes the *Firm Item* with the *Firm Code* 206.

6.3.1.6 FSB Entry Options

This section describes available options referring to the *Firm Security Box* (FSB).

Commands related to *Firm Security Boxen* (FSB) have the following setup:

```
/fsb<Firm Code> [<FSB Options>] <Main Command>
```

The following options are available:

Command	**`/fsb`** - FSB Entry
	This option initiates a FSB command sequence. Expects the *Firm Code* the FSB entry refers to as argument.
Syntax	`/fsb<Firm Code>`
Command	**`/fk`** - Firm Key
	Use this option to specify a new *Firm Key* (32 Bytes). ⚠ Please act with extreme caution when using this option! When changing the existing *Firm Key* you deeply interfere in encryption and programming processes! Because then all future encryption operations and all *CmContainer* programming will refer to this "new" *Firm Key* ! "New" encrypted applications will not run with "old" programmed *CmContainer*! Vice versa, "old" encrypted applications will not run with "new" programmed *CmContainer*! For your own safety, the option "Changing the *Firm Key*" has to be activated by Wibu-Systems.
Syntax	`/fk:0x<hex data>`

6.3.1.7 Product Item Options

This section describes the various options related to *Product Items* or *Product Item Options* (PIO).

ℹ Necessary requirement for programming *Product Items* and PIO is an already existing *Firm Item*.

Product Item options commands are structured in the following way:

```
/f<Firm Code> [...] /p<Product Code>[...] [<PIO Options>] <Main Command>
```

TVB (**Trailing Validation Block**)

ℹ You have an option to perform an additional check before executing programming sequences. This holds for all *Product Item Options* with the exception of Text and User Data.
Using so-called *Trailing Validation Blocks* [TVB] you may define dependencies for single programming sequences. Depending on set data (d), serial numbers (s) or update counter (u), commands are only executed when meeting the specified criteria. For example, a programming is performed only with a specified serial number, or with a specified number of permitted updates.

By default, all TVBs are set, i.e. the programming sequences vary with a maximum, and the programming is possible only into the desired *CmContainer* including the specified status.

Command	**`/p`** - Product Code
	Defines the *Product Code* to be used. Expects an unsigned decimal value as argument. ℹ Optionally, the item reference or the *Feature Code* can be specified as further selection parameters. The *Item* reference value must be enclosed in square brackets.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/p** - Product Code
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/p<value>[=<new value>][:<Feature Code>|<reference>][,<TVB dep.>]`

Selection of first *Product Items* with *Product Code* 13	`/p13`
Selection of *Product Item* with *Product Code* 13, *Feature Map*=`0x00000001`	`/p13:0x00000001`
Selection of *Product Item* with *Product Code* 13, *Product Item Reference* = 16	`/p13:[16]`

Command	**/papwd** - Access Password
	Adds, updates or deletes a Product Item's Access Password PIO. Accepts a text as password argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Please note, that using this PIO requires the *CmContainerType CmDongle* and a minimum Firmware version of 4.0.

Syntax	`/papwd:"<character string>"[,<TVB dep.>]`
	`/papwd[\"<character string>\"][,<TVB dep.>]`
	Removes the PIO (-) or sets the specified password.

`/papwd:"Password"`	specifies the password `Password`.

Command	**/pat** - Activation Time
	Adds, updates or deletes the PIO *Activation Time* of a *Product Item*. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pat-|[a<date>|r<offset>][,<TVB dep.>]` Removes the PIO (-) or sets an (a)bsolute or a (r)elative *Activation Time*.

Command	**/pata** - Activation Time, absolute
	Sets the *Activation Time* to the specified absolute value. Date inputs are accepted only before January 1st, 2100 00:00:00 UTC. Expects a date optionally followed by a time and the time zone as argument. If the time zone is omitted, the system's time zone will be used instead. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/pata<YYYY><Month><DD>[,<HH>:<MM>:<SS>[PST|MST|CST|EST|UTC|CET|EET][,<TVB dep.>]`
	or according to ISO-8601:
	`/pata<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm` or `±hhmm] [±hh][,<TVB dep.>]`

Sets the *Activation Time* to December 31st, 2012, 1 second to midnight, UTC	`/pata2012Dec31,23:59:59UTC`

Command	**/patr** - Activation Time, relative
	Adds the specified number of days to the current *Activation Time*. Expects an integer value greater than or equal to zero as argument. If this *Firm Item* does not exist yet, the current system time plus the specified offset will be set as activation time. For example: `/patr1` corresponds to 1 day immediately starting. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/patr<days>[,<TVB dep.>]`

Command	**/pcoli** - Customer Owned License Information (COLI)
	Adds, updates or deletes the PIO *Customer Owned License Information* of a *Product Item*. Accepts a text (up to 256 characters) enclosed in double quote character escape sequences as argument.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pcoli** - Customer Owned License Information (COLI)
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/pcoli-`
	`/pcoli:"<text>"[,<TVB dep.>]`
	Deleting the PIO (-) or setting the specified text.

Command	**/ped** - Extended Protected Data
	Adds, updates of deletes the PIO *Extended Protected Data* of a *Product Item*. Input of the field index (type) [`0-127`] and a sequence of hexadecimal digits (up to 256 bytes) with preceded `0x`. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/ped<extended type>-|[:0x<hex data>][,<TVB dep.>]` Removes the PIO (-) or sets the specified data The specification of the hex number always has to be pair, i.e. `0x1` is invalid but not `0x01`.
e.g.	`/ped0:0x75BCD15` Adds the decimal value 123456789 to the field (type) 0. `/ped2-` Deletes the field (type) 2.

Command	**/pet** - Expiration Time
	Adds, updates or deletes the PIO *Expiration Time* of a *Product Item*. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pet-|[a<date>|r<offset>][,<TVB dep.>]` Removes the PIO (-).

Command	**/peta** - Expiration Time, absolute
	Sets the *Expiration Time* to the specified absolute value. Date inputs are accepted only before January 1st, 2100 00:00:00 UTC. Expects a date optionally followed by a time and the time zone as argument. If the time zone is omitted, the system's time zone will be used instead. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	`/peta<yyyy><month><dd>[,<hh>:<mm>:<ss>[PST|MST|CST|EST|UTC|CET|EET][,<TVB dep.>]`
	or according to ISO-8601:
	`/peta<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm or ±hhmm] [±hh][,<TVB dep.>]`

e.g.	Sets the *Expiration Time* to December 31st 2009, 1 second to midnight, UTC: `/peta2009Dec31,23:59:59UTC` or `/peta2009-12-31T23:59:59Z`

Command	**/petr** - Expiration Time, relative
	Adds the specified number of days to the current *ExpirationTime*. Expects an integer value greater than or equal to zero as argument. If this *Firm Item* does not exist yet or in the case of an update an existing *Product Item* so far had no *Expiration Time*, the current system time plus the specified offset will be set as *Expiration Time*. For example: `/petr1` corresponds to 1 day immediately starting. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/petr<days>[,<TVB dep.>]`
e.g.	Extends the *Expiration Time* by 30 days. `/petr30`

Command	**/pfm** - Feature Map
	Adds, updates or deletes the PIO *Feature Map* of a *Product Item*. Expects an unsigned decimal or a hexadecimal value preceded by `0x`. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pfm-|[<value>][,<TVB dep.>]`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pfm** - Feature Map
	Deletes the PIO (-).

Command	**/phd** - Hidden Data
	Adds, updates of deletes the PIO *Hidden Data* of a *Product Item*. Input of the field index (type) [0-127] and input of an ID for an extended PIO type. Either as access code or data section. The default and optimal data section entry length equals 242 bytes which is shorter than the maximum entry length of 256 bytes. Using this default length optimizes hardware resource performance in the *CmContainer*. Reading data is automatically done across entries, i.e. when an entry is completed by the maximum length automatically the next entry is read. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	Fills the *Hidden Data* PIO with user-defined data.
	The optimal length covers 242 Bytes.
	`/phd<ext. type>,[<acc. code>][:0x<hex data>][,<TVB dep.>]`
	Fills the *Hidden Data* PIO with <count> bytes of random data.
	`/phd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>]`
	Removes the PIO (-).
	`/phd<ext. type>-`

e.g.		
	`/phd15:0x1122334455`	Fills the field (type)15 of the *Hidden Data* PIO with user-defined data. The optimal length covers 242 Bytes. The specification of the hex number always has to be pair, i.e. `0x1` is invalid but not `0x01`
	`/phd16,<acc. code>:r32`	Fills the field (type)16 of the *Hidden Data* PIO with 32 bytes of random data. The optimal length covers 242 Bytes. The *Access Code* `<acc. code>` can be a text input or an input of 16 bytes in hexadecimal format.

Command	**/plq** - License Quantity
	Adds, updates or deletes the PIO *License Quantity* of a *Product Item*. Accepts an unsigned decimal value as argument. Also permissions for network access can be specified. The following predefined configurations are supported.

Parameter	Description
`default`	sets default permissions
`local`	permits only local usage. If using a remote desktop connection (remote session) for accessing local only lienses, please note the following: • in the case of operating systems, such as, Windows7, 8, 10 etc. this license access works on connecting via remote session. • in the case of server operating systems, such as, Linux, Windows Server 2012, etc., this license access will not work and the event code 239 is returned.
`red`	sets permissions for Triple Mode Redundancy (TMR) usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of `0` and a preceding `#` character.
`wan`	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.

	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/plq-|[<counter>][:<access permissions>][[:]#<license tag>][,<TVB dep.>]`

e.g.		
	`/plq100`	Setup for 100 network licenses
	`/plq100:wan`	Setup for 100 network licenses and CmWAN
	`/plq100:red#42`	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42.

Command	**/plqa** - License Quantity
	Sets the *Product Item*'s *License Quantity* to the given absolute value. Accepts an unsigned decimal value as argument. Also permissions for network access can be specified. The following predefined configurations are supported.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/plqa** - License Quantity	
	Parameter	Description
	default	sets default permissions
	local	permits only local usage (Universal Licenses only)
	red	sets permissions for Triple Mode Redundancy usage. sets permissions for Triple Mode Redundancy usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of 0 and a preceding # character.
	wan	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	/plqa[:]<counter>[:<access permissions>][[:]#<license tag>][,<TVB dep.>]	
e.g.	/plqa100	Setup for 100 network licenses
	/plqa100:wan	Setup for 100 network licenses and CmWAN
	/plq100a:red#42	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42

Command	**/plqr** - License Quantity, relative	
	Increments or decrements the *Product Item*'s *License Quantity* by the given value. Accepts an unsigned decimal value as argument. Also permissions for network access can be specified. The following predefined configurations are supported.	
	Parameter	Description
	default	sets default permissions
	local	permits only local usage (Universal Licenses only)
	red	sets permissions for Triple Mode Redundancy usage. sets permissions for Triple Mode Redundancy usage. In the scenario of an increased availability a license access using the License Quantity considers other server providing matching entries. If a license quantity is set up for the Triple Mode Redundancy usage, an additional license tag must be specified. Required is an integer greater than a value of 0 and a preceding # character.
	wan	sets permissions for CmWAN usage (Wide Area Network, WAN). Note that you need a separate *Firm Security Box* (FSB) license entry is required you are able to receive by Wibu-Systems.
	Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	/plqr[:]<counter>[:<access permissions>][[:]#<license tag>][,<TVB dep.>]	
e.g.	/plqr100	Setup for 100 network licenses
	/plqr-100	Removes 100 network licenses
	/plq100:red#42	Setup for 100 network licenses and Triple Mode Redundancy, license tag 42

Command	**/plt** - Linger Time	
	Adds, updates or deletes the PIO *Linger Time* of a *Product Item*. Accepts an unsigned decimal value as argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).	
Syntax	/plt-|[<seconds>][,<TVB dep.>]	
	Removes the PIO (-).	
e.g.	/plt15	adds a *Linger Time* of 15 seconds to the PIO.

Command	**/pltdepth** - License Transfer Depth	
	Adds or updates the *Product Item*'s license transfer depth. Accepts an unsigned decimal value between 0 and 65535 as argument.	
Syntax	/pltdepth[:]<transfer depth>]	
e.g.	/pltdepth:4	adds or updates the*Product Item*'s transfer depth to a value of 4.

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pltltktarg** - License Transfer LTK Targets
	Adds or updates the *Product Item*'s list of LTK (Licenses Transfer Key) transfer targets. Accepts a list of 1-28 CmActLicense IDs (CmActId).
Syntax	`/pltltktarg:<CmAct ID1>[,<CmAct ID 2>[,<CmAct ID3>]]...`

Command	**/pltmbp** - License Transfer Maximum Borrow Period	
	Adds or updates the *Product Item*'s license transfer maximum borrow period. Expects either a time span measured in days or the token `none` to specify an unlimited period as argument..	
Syntax	`/pltmbp:<time span>|none`	
e.g.	`/pltmbp:3`	adds or updates the*Product Item*'s maximum borrow period to a value of 3 days.

Command	**/pltperm** - License Transfer Permission	
	Adds or updates the *Product Item*'s license transfer permissions. Expects either the token `none` to deny any permissions or a comma separated list of permission tokens as argument. The following permissions are available:	
	permission	Description
	`fitransfer`	the Firm Item may be transferred
	`return`	the license may be returned
Syntax	`/pltperm:<permissions>`	
e.g.	`/pltperm:return`	allows the transfer license to be returned

Command	**/plttarg** - License Transfer Targets
	Adds or updates the *Product Item*'s license transfer list of license transfer targets. Accepts a list of 1-28 CmActLicense IDs (CmActId).
Syntax	`/plttarg:<CmAct ID1>[,<CmAct ID 2>[,<CmAct ID3>]]...`

Command	**/plttype** - License Transfer Typ	
	Adds or updates the *Product Item*'s license transfer list of license transfer type. Accepts one of the following transfer type tokens as argument:	
	Token	Transfer Type
	`none`	no transfer allowed
	`movelicenses`	move network licenses
	`borrowlocallicense`	borrow license for local use
Syntax	`/plttype:none|movecomplete|moveunits|movelicenses|borrowcomplete|borrowlocallicense`	

Command	**/pmer** - Maximum Encryption Rate	
	Adds, updates or deletes a Product Item's Maximum Encyption Rate. Expects the encryption rate value given as the maximum number of encryptions per 30 seconds as argument.	
Syntax	`/pmer-|[<value>][,<TVB dep.>]`	
	Removes the PIO (-) or sets the maximum number of encryptions.	
e.g.	`/pmer20`	maximum encrypts 20 times per 30 seconds.

Command	**/pmi** - Module Item	
	Adds, updates or deletes a *Module Item*. Expects the *Module Item*'s *Product Code* given as an unsigned decimal value as argument. Optionally, the item reference or the *Feature Code* can be specified as further selection parameters. The item reference value must be enclosed in square brackets.	
Syntax	`/pmi<Product Code>[:<Feature Code>|<reference>][,<TVB dep.>]`	
e.g.	`/pmi13`	Select the first *Module Item* with *Product Code*=13
	`/pmi13:0x00000001`	Select the *Module Item* with *Product Code*=13, Feature Map = 0x00000001
	`/pmi13:[16]`	Select the *Module Item* with *Product Code*=13, item reference = 16
	On creating *Module Items* at the *Product Item* level the basic commands[52] are called separately for each *Module Item* and then for the complete *Product Item* level. `-p3000 -pt:"Parent Product Item with ModuleItems" -pup30 -pfm0x02 -pmi3120 -pt:"Module A" -puca100 -ca -pmi3130 -pt:"Module B" -puca20 -psd0:r32 -ca -ca`	

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pmp** – Maintenance Period
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item*. Accepts start date and end date of the period as argument. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmp-|[d[<start date>,]<end date>|i[<start value>,]<end value>[,<TVB dep.>]` Removes the PIO or sets start and end of a *Maintenance Period*, either given as (d)ates or given as unsigned (i)nteger values. The start date is 2000-01-01 00:00:00.

Command	**/pmpd** – Maintenance Period (Date)
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item* using a date. Accepts start date and end date of the period as argument. The start date may be omitted. In this case it is set to 2000-01-01, 00:00:00 UTC Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmpd[<start time>,]<end time>[,<TVB dep.>]`
e.g.	`/pmpd2011Jul01,00:00:00,2012Jun30,23:59:59` or `/pmpd2011-07-01T00:00:00,2012-06-30T23:59:59` Sets a one-year *Maintenance Period* beginning with July 1st, 2011.

Command	**/pmpi** – Maintenance Period (Integer)
	Adds, updates or deletes the PIO *Maintenance Period* of a *Product Item* using an integer. Accepts start and end of the period given as unsigned integer values as argument. The start date is 2000-01-01 00:00:00. The start date may be omitted. In this case it is set to a value of 0. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none). Requires *CodeMeter* Firmware 1.18 or higher.
Syntax	`/pmpi[<start value>,]<end value>[,<TVB dep.>]`
e.g.	`/pmpi394416000` Sets a *Maintenance Period* until July 1st, 2012. The difference of 4565 days to 1.1.2000 is 394416000 seconds.

Command	**/pmrt** - Minimum Runtime Version
	Adds, updates or deletes the *Minimum CodeMeter Runtime Version*. Accepts a version number consisting of major version, minor version and optionally the build number as argument, e.g. '5.20' or '5.20.1300'. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pmrt-|[<major version>.<minor version>[.<build>]][,<TVB dep.>]` Deletes the PIO (-).
e.g.	`/pmrt5.20` adds a *Minimum Runtime Version* Major Version 5 and Minor Version 20. `/pmrt5.20.1300` adds a *Minimum Runtime Version* Major Version 5, Minor Version 20 and Build Number 1300.

Command	**/pnmu** -Named User
	Adds, updates or deletes a *Named User* PIO of a *Product Item*. Accepts a `<check mode>` specification followed by `<configuration options>` and, depending on the specified mode, (1) a user name, (2) domain and user name or (3) a user-defined text as `<text>` argument. The `<configuration options>` also allow to determine whether the specified `<text>` is to be stored as plain text in the *CmContainer*. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pnmu-|:<check mode>[,<configuration options>]:<text>[,<TVB dep.>]`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/pnmu** -Named User
	Deletes the PIO (-).
	Valid `check mode` specifications cover:

`user`	the user name is checked
`userdomain`	domain and user name are checked
`userdefined`	user-defined text is checked

Valid `configuration options` cover:

`noplain`	plain text is not to be stored in the *CmContainer*, default
`plain`	plain text is to be stored in the *CmContainer*

e.g.

`/pnmu:user,plain:"Doe"`	Sets user name 'Doe', to be stored as plain text.
`/pnmu:userdomain,noplain:"MyDomain\Doe"`	Sets domain 'MyDomain' and user name 'Doe', no to be stored as plain text
`/pnmu:userdefined,plain:"MyText"`	Sets userdefined text 'MyText', to be stored as plain text.

Command	**/pnwc** - Network License Counter
	Deprecated, please use option `/plq` instead (identical syntax).

Command	**/ppd** - Protected Data
	Adds, updates of deletes the PIO *Protected Data* of a *Product Item*. Accepts a sequence of hexadecimal digits (up to 256 bytes) preceded by **0x**. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/ppd-|[0x<hex data>][,<TVB dep.>]`
	Deletes the PIO (-).

Command	**/psd** - Secret Data
	Adds, updates of deletes the PIO *Secret Data* of a *Product Item*. Input of the field index (type) [`0-127`] and input of an ID for an extended PIO type. You specify a data range. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).

Syntax	Fills the *Secret Data* PIO with user-defined data.
	`/psd<ext. type>,[<acc. code>][:0x<hex data>][,<TVB dep.>]`
	Fills the *Secret Data* PIO with `<count>` bytes of random data.
	`/psd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>]`
	Removes the PIO (-)
	`/psd<ext. type>-`

e.g.

`/psd15:0x1122334455`	Fills the field (type) *Secret Data* PIO with user-defined data. The specification of the hex number always has to be pair, i.e. `0x1` is invalid but not `0x01`
`/psd16:r32`	Fills the field (type) 16 of the *Secret Data* PIO with 32 bytes of random data.

Command	**/pt** - Text
	Adds, updates or deletes the PIO *Text* of a *Product Item*. Accepts a character string (up to 256 characters) enclosed in double quote characters as argument.
Syntax	`/pt -` `/pt:"<text>"`
	Removes the PIO (-).

Command	**/puc** - Unit Counter
	Adds, updates or deletes the PIO *Unit Counter* of a *Product Item*. Accepts an unsigned decimal value as argument. Depending on the chosen mode the argument is interpreted as (a)bsolute or (r)elative value. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/puc-|[a<value>|r<value>][,<TVB dep.>]`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/puc** - Unit Counter
	Removes the PIO (-) or sets an (a)bsolute or (r)elative value of the *Unit Counter*.
Command	**/puca** - Unit Counter, absolute
	Sets the *Unit Counter* to the specified value. Expects an unsigned decimal value smaller than or equal to `4294967294` as argument for Firmware versions newer than 1.18. For a firmware version prior to 1.18 the maximum value is `16777215`. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/puca<value>[,<TVB dep.>]`
e.g.	`/puca25` Sets the value of the *Unit Counter* PIO to the absolute value of 25.
Command	**/pucr** - Unit Counter, relative
	Increments the *Unit Counter* of a *Product Item* by the specified amount. Expects an signed decimal value in the range `[-2147483648, 2147483648]` as argument for Firmware versions newer than 1.18. For a firmware version prior to 1.18 the range is limited to `[-16777215, 16777215]`. Individual Trailing Validation Block (TVB) dependencies may be defined for this PIO: (d=(d)ata, s=(s)erial numbers, u=(u)pdate counter, or none).
Syntax	`/pucr<signed value>[,<TVB dep.>]`
e.g.	`/pucr10` Increases the value of the *Unit Counter* PIO by a value of 10.
Command	**/pud** - User Data
	Adds, updates or deletes the PIO *User Data* of a *Product Item*. Accepts a sequence of hexadecimal digits (up to 256 bytes) preceded by **0x**.
Syntax	`/pud-|[0x<hex data>]` Removes the PIO (-).
e.g.	`/pud0x1122334455` Assigns the specified value to the *User Data* PIO. The specification of the hex number always has to be pair, i.e. `0x1` is invalid but not `0x01`.
Command	**/pup** - Usage Period
	Adds, updates or deletes the PIO *Usage Period* of a *Product Item*. Input as integer value greater than or equal to a value of null. Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	`/pup-|[:<days>][,<TVB dep.>]` Removes the PIO (-).
e.g.	`/pup:30` adds or updates an *Usage Period* of 30 days.
Command	**/pupa** - Usage Period, absolute
	Sets the length of the *Product Item*'s Usage Period to the given number of `<days>`. Input as integer value greater than or equal to a value of null. Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	`/pupa[:]<days>[,<TVB dep.>]`
e.g.	`/pupr:30` sets the length of an *Usage Period* to 30 days.
Command	**/pupr** - Usage Period, relative

© Copyright 2018 WIBU-SYSTEMS AG

	Extends a Usage Period by the given number of `<days>`. If the PIO does not exist, yet, a Usage Period of the given length will be added. Input as integer value greater than or equal to a value of null. Requires *CodeMeter* Firmware Version 1.11 or higher.
Syntax	`/pupr[:]<days>[,<TVB dep.>]`
e.g.	`/pup:30` extends the length of an *Usage Period* by 30 days.

Command	**`/puvd`** - Universal Data
	Adds, updates or deletes a PIO *Universal Data* of a *Product Item*. Expects the *Universal Data*'s index as an unsigned decimal value as argument.
Syntax	`/puvd[:]<index>`
e.g.	`/puvd:15` add or update a *Universal Data* field in index 15.

Command	**`/puvdaes`** - Universal Data AES key storage
	Option for the setup of a PIO *Universal Data* of a *Product Item* as storage for AES keys. Expects the key length as argument followed by the specification of the key. It is also possible to initiate a random generation of the key within the *CmContainer* by the specification of the token '`r`'(andom).

Syntax	`/puvdaes<key length>:<key>`
	`/puvdaes<key length>:r`
	Currently the following key lengths are supported:
	• 256

e.g.	`/puvdaes256:r` generates a AES random key in the *CmContainer* with the key length 256 bits.

Command	**`/puvdrsa`** - Universal Data RSA key storage
	Option for the setup of a PIO *Universal Data* of a *Product Items* as storage for RSA keys. Expects the key length as argument followed by the specification of the keys. The keys can be set directly by the specification of `exponent`, `prime p` and `prime q`. It is also possible to initiate a random generation of the keys within the *CmContainer* by the specification of the token '`r`'(andom), optionally followed by the `exponent` to use. `Exponent`, `prime p` and `prime q` have to be specified as hexadecimal data in Little Endian byte order. Each block of hexadecimal data has to be preceded by '`0x`'.

Syntax	`/puvdrsa<key length>:<exponent>,<prime p>,<prime q>`
	`/puvdrsa<key length>:r[:<exponent>]`
	Currently the following key lengths are supported:
	• 4096

e.g.	`/puvdrsa4096:r` generates a RSA random key in the *CmContainer* with the key length 4096.

Command	**`/pwupidata`** - WUPI Data
	Adds or removes a sequence of *Hidden Data* PIOs that are used as WUPI data storage. New WUPI data storage either can be filled with the contents of a specified file or preset with a user-defined fill byte. In this case, the size of the WUPI data has to be specified. `ext. type`: number of the first *HiddenData* field to be programmed (default value of `0`). `block size`: number of bytes to be stored for each field (`0..256`, default and optimum value is `242`). The default and optimal data section entry length equals 242 bytes which is shorter than the maximum entry length of 256 bytes. Using this default length optimizes hardware resource performance in the *CmContainer*. Reading data is automatically done across entries, i.e. when an entry is completed by the maximum length automatically the next entry is read. `acc. code`: *HiddenData AccessCode* (default is calculated). `size`: number of bytes to be created as WupiData. `fillbyte`: fill bytes used to fill the reserved storage by WupiData.

© Copyright 2018 WIBU-SYSTEMS AG

Syntax	`/pwupidata:[e<ext. type>] [,b<block size>][,a<acc. code>]:<file>`	Allocates WUPI Data storage, filled with the contents of file `<file>`.
	`/pwupidata:s<size>[,e<ext. type>][,b<block size>][,a<acc. code>][,f<fill byte>]`	Allocates `<size>` bytes of WUPI Data storage, initialized with the value `<fill byte>`.
	`/pwupidata[:s<size>][,e<ext. type>][,b<block size>]-`	Removes the WUPI Data storage.

6.3.1.8 Special Commands

This section describes special options. The following options are available.

Command	**`/bkp`** - Backup File
	Enables the backup file viewer mode. expects a *CodeMeter* backup file. Only allowed in combination with list option `/l`.
Syntax	`/bkp:\"<Backup Datei>\"`
Command	**`/crac`** - Create Remote Activation Context File (*.WibuCmRaC)
	Enables the creation of a Remote Activation Context File (`*.WibuCmRaC`). Optionally the target file or the target directory can be specified. If a Remote Activation Context File (`*.WibuCmRaC`) is specified, the contents of every target *CmContainer* will be stored there. Otherwise, for every target a Remote Activation Context File "`<serial number>.WibuCmRaC`" will be created in the specified target directory or in the current directory if the argument is omitted. This option cannot be used in the Remote Activation mode.
Syntax	`/crac[:<*.WibuCmRaC file>|<*.WibuCmRaC target directory>]`
Command	**`/rcl`** - Cleanup Registry (*CmDongle* only)
	Deletes the *CodeMeter* related Windows registry entries of the chosen categories. The supported *CmDongle* form factors comprise: *CmStick* (c), *CmStick/M* (m), Removable Media (r), Unknown USB devices (u). If omitted all registry entries belonging the categories (c) and (m) will be deleted by default. This command requires administrator privileges. Use this feature with care. The cleanup may have unknown side effects. Cleaning the categories Removable Media or Unknown USB devices may have also the effect that entries not related to *CodeMeter* will be deleted.
Syntax	`/rcl[:cmru]`
Command	**`/log`** - Logging
	Enables logging. Expects the path to a logfile as argument. The optional mode specifier '+' has the effect that the log output will be appended to the log file, otherwise the file's contents will be overwritten.
Syntax	`/log:"<logfile>"` `/log[:]\"<logfile>\"[+]`
Command	**`/?`** - Help
	Issues further help on desired topics. If no topics are specified, the complete help list is issued.
Syntax	`/?[<topic>|<option>]`
Command	**`/v`** - Verbose Mode
	Activates the detailed display mode.
Syntax	`/v`

© Copyright 2018 WIBU-SYSTEMS AG

Command	**/val** - Validation Mode
	Activates the validation mode. In this mode a *CmContainer* returns a confirmation sequence after each successful programming operation. The received data is validated with the *Firm Security Box*. By default, this mode is deactivated increasing the performance. Using this option reactivates the validation routine.
Syntax	`/val`
Command	**/vslf** - Validation Of Signed Log Files
	Validates either the contents of a given signed *CodeMeter* log file or a sequence of log files located in a given directory. Expects the path to the file that contains the public keys to use for validation and the path to the log file respectively log directory as arguments.

Syntax	`/vslf:<public key file>,<log file>|<log directory>`
	The `<public key file>` musst be created as comma separated file (CSV).
	For this file the following notation is valid:
	`<major version>,<minor version>, <firm code>, 0x<public key>`
	You can specify several different values and *Public Keys* for separate runtime versions.
	The required values and *Public Key* can be obtained from the generated *CodeMeter* log files. Please note, that the *Public Key* differs for each *Firm Code* and each *CodeMeter* Runtime version, i.e. you must have access to the the same version as your customers to obtain this information.

e.g.	From the *CodeMeter* log file:
	`2014-06-24T06:06:19 SignedLogfile FirmCode:10, PublicKey:a809304778d517c44a22d65e1fcedd51a4e2a956fa89e93bb1a24e2100000000a2ad17e685306d6e15eb6b7ebc8cc72ebc97c0f52721b584836696de00000000, Runtime-Version:5.20.1432.500, LogfileId:1`
	the following `<public key file>` is derived:
	`5,20,10,0xa809304778d517c44a22d65e1fcedd51a4e2a956fa89e93bb1a24e2100000000a2ad17e685306d6e15eb6b7ebc8cc72ebc97c0f52721b584836696de00000000`

Programming examples

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /petr30 /puca1492 /pfm0x8000 /ca
```

Adding a *Product Item* with the *Product Code* 2001, a 30 days *Expiration Time*, an *Unit Counter* value of 1492, and the *Feature Map* 0x8000 to the *Firm Item* with the *Firm Code* 6000010 in *CmContainer* 1-1234.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /petr335 /pucr426 /pt: "Text" /cu
```

Updating the *Product Item* with the *Product Code* 2001. The *Expiration Time* is extended by 335 days, and the *Unit Counter* increased by 426 units. In addition, a text is added to the *Product Item*.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /pet /cd
```

Deleting the *Expiration Time* of the *Product Item* 2001.

```
CmBoxPgm /f6000010 /p2008 /ca /laf:1-1234.WibuCmRaC,1-1234.WibuCmRaU,1-1234.WibuCmRaM
```

Adding the *Product Item* 2008 per remote programming. Besides the remote Update File, also a Modified Context File is created allowing for later reprogramming.

```
CmBoxPgm /qs1-1234 /f6000010 /p2001 /plq5 /ca
```

Setting the *License Quantity* to 5 for *Product Item* 2001 on *Firm Item* 6000010. Sets in *Product Item* 2001 in *Firm Item* 206 the License Quantity to 5.

6.3.2 CodeMeter License Central

Ticket System

Integrating software protection into the software is one but fundamental aspect which strongly affects system security. At the same time, the integration of software protection into sales, production and support processes also determines whether a system is easy to operate, and thus is accepted by both customers and employees. The latter processes we summarize as Back Office Integration (BOI).

CodeMeter License Central is a ticket system with a standardized graphical user interface to create, manage, and deliver both *CmDongles* and *CmActLicenses*.

A detailed description of *CodeMeter License Central* please find in the manual to be downloaded in the developer area at www.wibu.com.

Editions of *CodeMeter License Central*

CodeMeter License Central is available in two editions:

© Copyright 2018 WIBU-SYSTEMS AG

- *CodeMeter License Central Desktop Edition*
- *CodeMeter License Central Internet Edition*

Both Desktop and Internet Edition are functionally identical, differing only in licensed use, integration, and support services.

The *Desktop Edition* can be used on a single server in your company. The operating system is Linux Ubuntu. The database runs on MySQL only. Access is via a browser-based front end. You get a VM image which meets the requirements and requires only the VMware Workstation or ESX/EXSi server to run.

The *Internet* Edition is designed for distributed installation on multiple servers. You can use an existing database server (MySQL or Microsoft SQL Server; for support for other database platforms please inquire directly). The core of *CodeMeter License Central*, based on an Apache Web Server and Tomcat Server, can be installed on other Linux distributions or Windows if you desire.

Sales Interface

When you program a *CmContainer* for a specific license, you send a related request with an item number to *CodeMeter License Central* and receive back a unique ticket. Since this scenario in most cases involves the selling of this item, we refer to this interface as the *Sales Interface*. The ticket contains the authorization to add the license to a *CmContainer*.

Depot Interface

You decide whether you instantly program the license yourself, deal with it later or transfer the ticket to your customer. If you decide to transfer, then your customer is able to collect the licenses bought at any time, for any *CmContainer*. We call the interface for collecting licenses the *Depot Interface*.

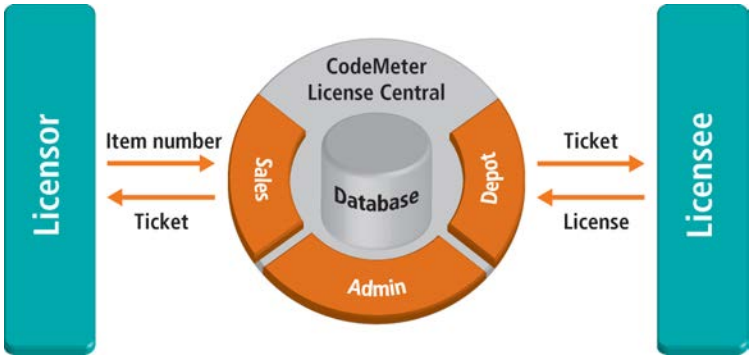


Figure 20: License Collection by Licensee

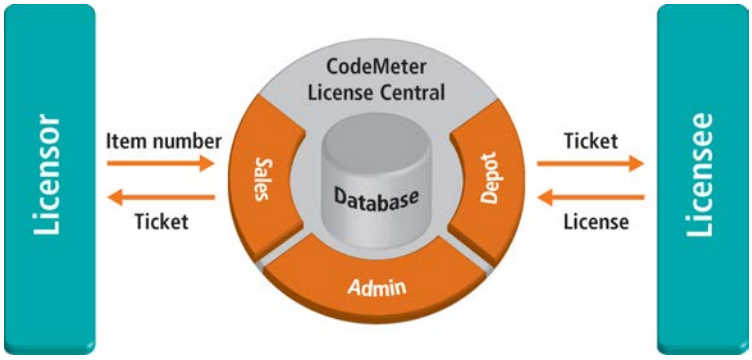


Figure 21: License Collection by Licensor

Admin Interface

Next to the *Depot Interface* and *Sales Interface*, *CodeMeter License Central* features the *Admin Interface*. The *Admin Interface* comprises functions for defining license properties (e.g. *Expiration Time*, *License Quantity*, etc.), for managing access rights, for generating statistics and reports, and for carrying out support activities.

The following figure shows an overview of interfaces and related functions in *CodeMeter License Central*.

© Copyright 2018 WIBU-SYSTEMS AG

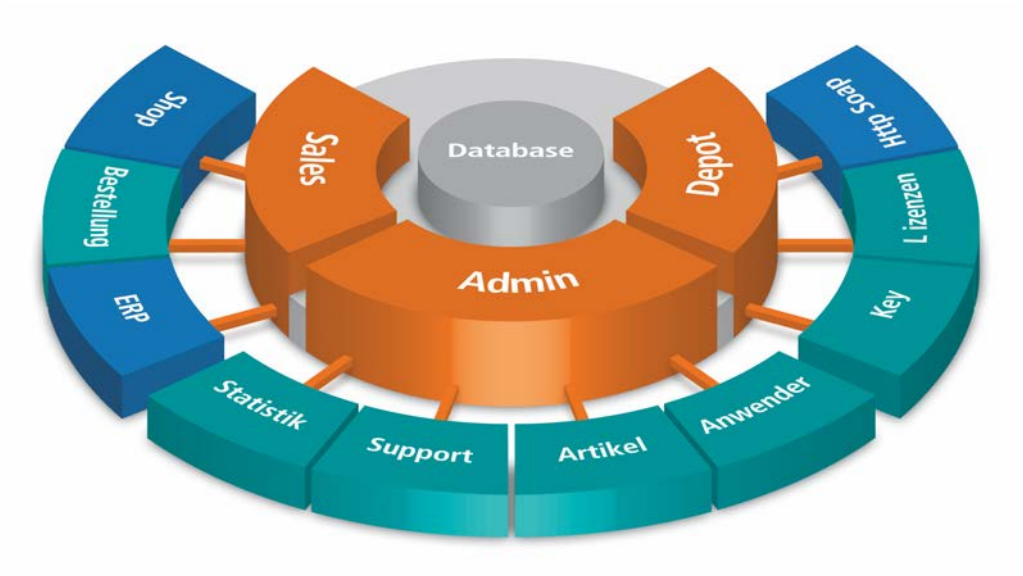


Figure 22: *CodeMeter License Central* - Interfaces and Functions

6.3.3 Programming by File Transfer

***CmDongle*-Licenses**

Remote updating a *CmDongle* requires some information on the *CmDongle* to be reprogrammed. This information is safely stored and transferred in a Context File (license request, `*.WibuCmRaC`).

Context File (`*.WibuCmRaC`) - License Request

The creation of a Context File (`*.WibuCmRaC`) is bound to the physical ownership of a *CmDongle*. On creation the *Firm Code* to be included is specified. Usually, the own *Firm Code* is specified because only the container holding it can be altered. In addition, the file holds the serial number of the *CmDongle*. When you as licensor receive the Context File (`*.WibuCmRaC`) from your licensee, you can see in detail which of your licenses and license options are stored in the *CmDongle*. The licensee generates this file in *CodeMeter Control Center* by the process of the license update.

Update File (`*.WibuCmRaU`) - License Update

On the basis of this Context File (`*.WibuCmRaC`), you as licensor, are able to generate a so-called Update File (`*.WibuCmRaU`), in order to modify existing licenses using the tools *CodeMeter License Editor*, *CmBoxPgm* or *CodeMeter License Central*. The provided options are the same as with physically existing *CmDongle* you can add new or alter existing licenses, e.g. extending *Expiration Time*, or delete licenses. The Update File (`*.WibuCmRaU`) holds the update sequences and is valid only for a specific *CmContainer*. A licensee is able to only one-time import the file into the specified *CmDongle*.

***Firm Update Counter* (FUC)**

After the successful import of the update file by the licensee in *CodeMeter Control Center* a specific counter, i.e. the *Firm Update Counter (FUC)*, at the *Firm Item* level is increased. By increasing the counter a repeated import of the Update File (`*.WibuCmRaU`) is invalid.

This is of special importance, for example, when the Update File (`*.WibuCmRaU`) holds programming commands which add a new license entry, increase a *Unit Counter* by a number of units, or extend an *Expiration Time* for a number of days.

Modified Context File (`*.WibuCmRaM`)

When creating an Update File (`*.WibuCmRaU`) automatically a so-called Modified Context File (`*.WibuCmRaM`) is created providing you with an image of the content your licensee owns when s/he imported the Update File (`*.WibuCmRaU`). In the case of a new update, e.g. license extension, you can either use a new Context File (`*.WibuCmRaC`) sent by the licensee, or you use the current Modified Context File (`*.WibuCmRaM`) as programming basis. Many licensors already in-house-create the Context File (`*.WibuCmRaC`) directly after programming, and can manage the update process without licensee interference.


If, in the meantime, the *CmDongle* has been reprogrammed by another licensor, all files keep valid.

The following figure illustrates this process.

© Copyright 2018 WIBU-SYSTEMS AG

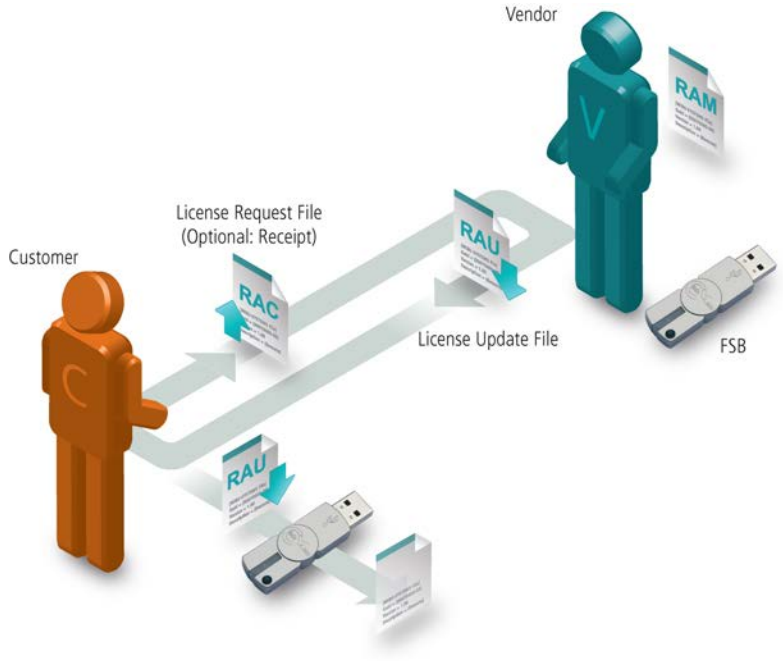


Figure 23: *CmFAS* - File-based Remote Update *CmDongle*

CmActLicense-Licenses

With two exceptions, remote programming of *CmActLicense* licenses largely follows the process for *CmDongles* as described above.

Firstly, before creating the initial Context File (license request, `*.WibuCmRaC`) the customer has to import an empty license container he receives by the vendor. This LIF file (License Information File) in the `*.WibuCmLIF` format holds information on the binding scheme 16 and additional activation options 16 of the *CmActLicense* license which are used to be able to uniquely bind the license to a computer or a device. Required hardware features of a computer or a device are detected and additional activation information transferred. Only on this bases the initial license request file is created. Following, based on this license request file the vendor reprograms this license request file into an Update File (license update, `*.WibuCmRaU`) the customer imports. Starting from this point the file exchange process between customer and vendor is the same for *CmDongle* and *CmActLicense* licenses.

Secondly, currently on reprogramming the context file into a update file a Modified Context File (`*.WibuCmRaM`) is not created.

The following figure illustrates this process.

© Copyright 2018 WIBU-SYSTEMS AG

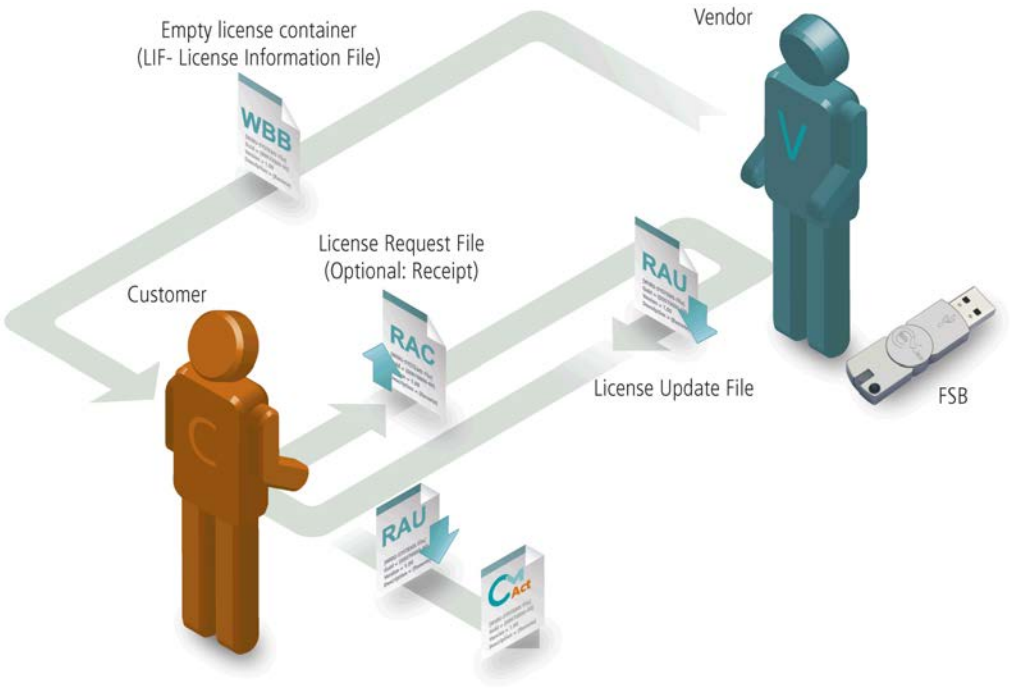


Figure 24: *CmFAS* - File-based Remote Update *CmActLicense*

© Copyright 2018 WIBU-SYSTEMS AG

6.4 Protection: How to encrypt using AxProtector CmE

Please note, that if you encrypt using *AxProtector* with *CmEmbedded* AxEngine for *CmActLicense Universal Firm Code*, you must for security reasons specify a Firmware version 1.14 or higher.
For *CmActLicense Firm Code* you must specify a Firmware version lower than 1.14.

6.4.1 Automatic Software Protection using AxProtector

No Programming Skills required

With *AxProtector* you have a tool at hand that can automatically encrypt already compiled executables. *AxProtector* allows you to integrate *CodeMeter*® into your application - quickly and smoothly - without the need to alter your source code. It is so easy to use, that integration can take place without any programming skills.

In just a few minutes, *AxProtector* encrypts and protects your application for a variety of project types

AxProtector is also available as a commandline variant 73 for Windows 32-bit / 64-bit, .NET, Linux, macOS, and Java applications. Using the *AxProtector* GUI is a simple way to generate a commandline that can be extended and used further to accomplish automatic protection.

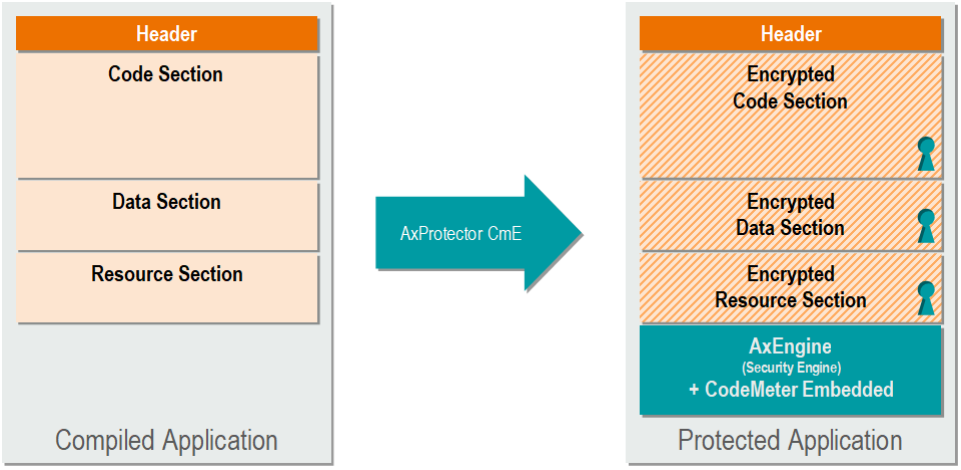


Figure 25: *AxProtector CmE*

The following table summarizes what kind of software applications can be encrpyted using various project types and tools for different operating systems:

Please note that applications protected by using *AxProtector* and then shipped are not identical to the original applications.
This may affect other already applied security measures, e.g. the use of signatures.
In such cases, you must perform other own security measures, e.g. the use of signatures after *AxProtector* has protected the application.

Application to be protected	Project type	GUI Windows	Commandline
Linux, Android	AxProtector Embedded		Windows commandline

AxProtector:

- supports the encryption of all existing *CodeMeter* license options (*Product Item Options*). Thus all necessary license information is integrated into the encryption, for example, network licenses, or license checks at runtime.
- features functions to identify debugger use: in the case a debugger is detected, a *CmContainer* can be locked.
- provides the feature of "on-demand-decryption", i.e. parts of the protected application (source code and resources) are decrypted only when accessed. This "on demand decryption" effectively protects against memory dumping and the extraction of unprotected versions.
- offers the use of freely customizable user message dialogs including the creation of individual texts for purchasing options or errors and also the embedding of company logos.

© Copyright 2018 WIBU-SYSTEMS AG

6.4.2 AxProtector Samples

The following *AxProtector* command line sample show how to use the basic parameter of licensing system (-kca 74), *Firm Code* / *Product Code* (-f 74/-p 74), CmEmbedded (-xca 73), framework (-cct 79) which have to be considered. For the other options see the *AxProtector* command line options 73.

Android CmActLicense

```
AxProtectorLin -o:P010 -kcm -f6000010 -p1210 -eurc1 -wu2 -cct:linux-armhf -u:UserMsgLin
-sl -nl -cav -ccs -xc -car3,1 -v -# helloCounter
```

Linux32/64, armv7EL CmActLicense

```
AxProtectorLin -o:P010 -kca -f6000010 -p1210 -eurc1 -wu2 -sl -nl  -cav -ccs -xca -car3,1
-v -# helloCounter
```

Linux armv7HF, armv8 CmActLicense

```
AxProtectorLin -o:P010 -kca -f6000010 -p1210 -eurc1 -wu2 -cct:linux-armhf -sl -nl -cav -
ccs -xca -car3,1 -v -# helloCounter
```

6.4.3 Commandline Options for AxProtector

The commandline application comes in several versions you find in the directory "%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin":

6.4.3.1 Basic Options

Option	-X
valid for	

Links the static library of the licensing system to the application to be protected.


This option is set by default.

Setting this option increases the security compared to linking the dynamic library.

Option	-XC
valid for	

Allows using *AxProtector* Linux x86/x86_64 with *CmEmbedded* instead of *CodeMeter License Server*.
This allows starting with Version 9.11 to protect Linux binaries (architectures ARM / Intel (32-64 bit)).


Please note, that this is supported for *CmDongle* only.

Option	-XCA
valid for	

Allows using *AxProtector* Linux x86/x86_64 with *CmEmbedded* including the programmed CmEmbedded Adapter instead of *CodeMeter License Server*. This allows starting with Version 9.11 to protect Linux binaries (architectures ARM / Intel (32-64 bit)).
A dynamic library 'libCmActAdapter.so' is sought (must locate in the same directory as the binary to protect) and added to the encrypted binary. If the library is not found, an error message informing on the cancelled encryption displays. Then the license access for CmActLicense uses this library.


Please note, that this is supported for *CmActLicense* only.

Option	-A[AES]
valid for	

Specifies the encryption algorithm (*CodeMeter* only).

By default, the AES encryption algorithm in CBC mode is used (Default).

© Copyright 2018 WIBU-SYSTEMS AG

6.4.3.2 Options for the Licensing System

Option	-K([CA][CM][WK])

valid for 

Specifies the licensing system.

Parameter	-KCA

uses *CmActLicense*.

Parameter	-KCM

uses *CmDongle* (Default).

Parameter	-KWK[-x]

uses *WibuKey*. x stands for:

1	uses encryption algorithm 1 (*WibuKey* only).
2	uses encryption algorithm 2 (*WibuKey* only).
3	uses encryption algorithm 3 (*WibuKey* only).
4	uses encryption algorithm 4 (*WibuKey* only) (Default).
5	uses encryption algorithm 5 (*WibuKey* only).



The following options should directly locate after the -K option since they refer to the actual defined licensing system.
When you use both licensing systems for an executable file, the options set are valid for the defined licensing system.

Option	-Fx

valid for 

Specifies the *Firm Code*(x) to be used.
Expects the input of an unsigned integer value <n>.

Option	-Px

valid for 

Specifies the *Product Code* (x) to be used.
Expects the input of an unsigned integer value <n>.

Option	-CFx

valid for 

Specifies the *Feature Code* (x) of the *Feature Map* to be used.



Only valid when using *CodeMeter*.

Expects the input of an unsigned integer value <n>.

Option	-RD([YYYYMonDD[,HH:MM:SS[:<Timezone>]]][:now]) or according ISO-8601 with T(ime) and Z(one) parameter: -RD([<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z][±hh:mm or ±hhmm] [±hh]]][:now])

valid for 

defines the *Release Date* for encrypting and decrypting (only *CmDongle* and *CmActLicense* only).
Specification is in format year, month, and optional hours, minutes, seconds, and the timezone
The input of [:now] applies the current date.

Option	-D:v

valid for 

Specifies the minimum driver version (v).
Input of v using (x.y).
Default setting: *CodeMeter* 4.20.

Option	-FW:v

valid for 

Defines the minimum firmware version(v).
Input of v using (x.y).
Default setting: *CodeMeter* 1.0

Option	-S([L][N|W]|C)

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Option	-S([L][N|W]|C)

Specifies the search order of the subsystem when searching for licenses.
The options `N` and `W` may be used alternatively only.

Parameter	-SL

Uses the local subsystem (local).

Parameter	-SN

Uses the network subsystem (network).

Parameter	-SLN

Uses first the local subsystem (local), then the network subsystem (network).

Parameter	-SNL

Uses first the network subsystem (network), then the local subsystem (local).

Parameter	-SW

Uses the Wide Area Network subsystem (WAN).

Parameter	-SLW

Uses first the local subsystem (local), then the Wide Area Network subsystem (WAN).

Parameter	-SWL

Uses first the Wide Area Network subsystem (WAN), then the local subsystem (local).

Parameter	-SC

First searches the local and then the network subsystem, and if a network was found the network drive.

Option	-N[C[A]|L[A]|N|X|X[A]]

valid for 

Specifies the network access.

Parameter	-NC[A]

convenient mode (compatibility mode): here each started instance on the network allocates a normal user limit, and the local access is unlimited (no user limit).

Parameter	-NL[A]

normal user limit: here each started instance allocates a license regardless whether the *CmContainer* is found locally or on the network.

Parameter	-NN

no user limit: here any number of instances can be started locally or on the network. No licenses are allocated.

Parameter	-NNI

aditionally uses the flag "Ignore Linger Time".
This license option allows to define an allocation time of the license after a protected application has been released or finished (more Information in the *CodeMeter* Developer Guide).

Parameter	-NS

station share: here several started applications on a client allocate only a single license.



You use this option when allowing the end-user to start the protected application several times. On terminal server each session allocates a single license. In virtual machines each machine allocates a single license.

Parameter	-NX[A]

exclusive mode: here only a single started instance per PC is allowed. Each access allocates a single license.
If the exclusive access is defined, the automatic runtime check is no longer internally activated. Using option '-car' 78 for the runtime check has now to be set explicitly.



Please note that *AxProtector*  does not support the option `[A]`.

Option	-SIG(Fx):(Px):(CFx):(SD|HD)DN[:CP]

valid for 

Selects a private key to calculate the signature.

Parameter	Fx

specifies the *Firm Code* (x) used for signature.

© Copyright 2018 WIBU-SYSTEMS AG

Option	-SIG(Fx):(Px):(CFx):(SD|HD)DN[:CP]

Parameter	Px

specifies the *Product Code* (x) used for signature.

Parameter	CFx

specifies the *Feature Code* (x) used for signature.

Parameter	SD

specifies whether a *Secret Data* datafield contains the signature key.

Parameter	HD

specifies whether a *Hidden Data* data field contains the signature key.

Parameter	DN

specifies the number / index of the *Hidden* or *Secret Data field.*

Parameter	CP

specifies the path to the certificate used to sign the executable.

Option	-SIG:PK[,CP]

valid for 

Defining the path to the private key file and the certificate used to sign and check the executable for code integrity.

Parameter	PK

specifies the path to the private key file

Parameter	CP

specifies the path to the certificate used to sign the executable

6.4.3.3 Options for Encrypting and Decrypting

Option	-CA[[A[!]],[Ct[,u]],[D[m]],[E],[G[!,[,1]]],[L],[M],[R[t][,m]],[S[p]],[T[t][,u]],[V],[Z]]

Encrypts the executable file using automatic encryption.

Parameter	-CAA <1>

valid for 

Activates the security options (Advanced Protection Schemes, APS).
<1> covers the options [0, 15]



When applying more than one security option (APS), you can combine the options 1, 2, 4 and 8 with "or".
are mutually exclusive. If both options are set, automatically -CAA8 is applied.

Option	Description
0	No resource encryption is performed.
1	Resource encryption applies (APS 1)
2	Static modification applies (APS 2)
4	Dynamic modification applies (APS 3)
8	Extended static modification applies (APS 4)
	CAA6 applies APS 2 and 3 CAA13 applies APS 1, 4 and 8

Parameter	-CACt<,u>

valid for 

Checks the *CmContainer* system time related to the PC time. A protected application runs only when the PC time in a time window is t minutes younger and, optionally, u minutes older than the *CmContainer* system time.

Parameter	-CAD<m>

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CAD<m>

Specifies the file access mode for the automatic encryption of files which have been encrypted using the option -CD.

0	Decrypts the file's content block by block (4Kb) on demand (on demand).
1	Decrypts the file's complete content at once on first access (at once). Depending on the file size an access may result in a delay.
2	Prevents that on file encryption the protected application is generally able to write data to the hard drive (vie only). Here all writing to a file is prevented.
4	Decrypts the content of very huge files (e.g. large MPG3 files with a size of 500 MB) used with file encryption (read and decrypt on demand). In this mode, by default, writing back is deactivated.

Parameter	-CAE

valid for 

Activates instantly the detection that a has been removed from the PC ('plug-out') (*CmDongle* only).


If the connection to a *CmDongle* should fail or the license cannot be accessed, you can assign a reasonable number of "ignores" allowing the end-user to continue working without a license access.

Parameter	-CAG<1>

valid for 

Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC).
<I> covers the options [0,367]


When applying more than one Anti-Debugging mechanism (ADC), you can combine options with "or".

Option	Description
1	Checks whether a debugger is attached to your application. In the case a debugger is detected, the application does not start (ADC1).
2	Checks additionally for Kernel debugger programs, e.g. "SoftICE". In the case a debugger is detected, the application does not start (ADC2).
4	Checks in an extended search for debugger programs running parallel to your applications. Cracker tools, such as IMPREC are detected. In the case a debugger is detected, the application does not start (ADC3).
8	Checks for all debugger programs. Then no debugger programs are allowed, i.e. also within developer environments (IDE), e.g. VISUAL STUDIO, DELPHI. In the case a debugger is detected, the application does not start (ADC4).
16	Locking of the license entry and thus of the hardware when a debugger program has been detected (ADC5).  Using this option requires that the developer has programmed the *CmContainer* with a *Firm Access Counter* (FAC). The *Firm Access Counter* (FAC) is located at the *Firm Item* level of a *CmContainer*. This counter allows you to control whether a *Firm Item* can be used for encryption and decryption operations. By default, the FAC is deactivated and has a value of 65535 (0xFFFF). It can be programmed to any other value between 1 and 65534. The *CmContainer* is locked when the FAC has a value of 0. Depending on the settings all licenses of a vendor can be locked if a hacker attack occurs. The owner / end-user of the locked *CmContainer* must contact the software vendor for unlocking codes. How and how often unlocking is granted depends on the vendor's policy.
32	Adds a mechanism to the application preventing the attachment of a debugger program to the application at runtime (generic debugger detection) (ADC6).  Please note, that this mechanism of a generic debugger detection is not supported neither for the AxProtector project type 'File Encryption", nor *SmartShelter SDL*.
64	Detects whether the application is to be started in a virtual machine and prevents this (ADC7).
128	Hardware locking is performed only with a valid *Firm Access Counter* (only in combination with ADC5 and *CmContainer*).
256	*Firm Access Counter* decrementing by 1 is performed (only in combination with ADC5 and *CmContainer*).

Parameter	-CAL

valid for 

Limits the automatic encryption to specified areas.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CAM

valid for 

Adds the menu items 'Control' and 'About' to the application's system menu.

Parameter	-CAR<t>,<m>

valid for 

Adds a runtime check to the automatic encryption.
The check occurs every <t> seconds. he default setting is 300 seconds (5 minutes).
<m> specifies how often the end-user is able to ignore a failed check (threshold).

Parameter	-CAS<p>

valid for 

Specifies the size of the protected application to be encrypted. You enter the length, in percent, anywhere from 0 to 100%.
The default setting is 75 percent.

Parameter	-CAT(t)(,u)

valid for 

On each start of the application, a *Certified Time* update is triggered. Then the application starts regardless of whether the update was successful or not, and writes it into the *CmContainer*.
Then the application starts when the time difference between the *Certified Time* and the system's PC-Time is not greater than <t> specified in hours.
<u> specifies the valid time span in hours within which the difference between the *Certified Time* and the system time is allowed to range without a new *Certified Time* update (*CodeMeter* only).

 <t> has to be equal or greater than <u>.

Parameter	-CAV

valid for 

Adds a code integrity check to the automatically encrypted application.
The code integrity check may also cover several executable files / libraries. Then each file is able to check the integrity of all other files.

For this the files must be written into a separate section [CheckCodeIntegrity DLLs] of the *.wbc file. Please export the *.wbc file by using the *AxProtector* GUI menu item "**File | Export**" and change the file meeting your requirements, for Windows e.g.:

```
[CheckCodeIntegrity DLLs]
Image1=draw.exe
Image2=rectangle.dll
Image3=circle.dll
Image4=triangle.dll
```

Moreover, on encrypting a library also a list of applications can be transferred allowed to load these libraries. On loading the library then it is checked whether the process name includes one of the specified names.

If not, an error message displays and subsequently the application closes.

This list of process / application names is also specified in the section [CheckCodeIntegrity DLLs]. Please use the parameter AllowedExe1, AllowedExe2 which are available, e.g.:

```
AllowedExe1=PlainApp.exe
AllowedExe2=ExtendedApp.dll
```

 If the same application name is also specified in the image 1, image2, etc. list above, a code integrity check is also performed for this application.

Option	-CC[[A[a:s],[B],[D],[E],[F],[H],[I],[K][M],[O],[Q],[R],[S],[T],[X]]

Sets compatibility parameters.

Parameter	-CCA

valid for

Defines the target system / subsystem in combination with the option -CCX for .NET executables including debugging of encrypted applications.

<a> contains the target platform [1,2].	1: x86 / Intel 32 Bit
	2: ArmV4i
<s> contains the subsystem [9]	9: Windows CE System

Parameter	-CCB

valid for

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CCB
	Disables usage of proprietary (B)ase relocation representation.

Parameter	-CCE
valid for	
	Specifies that the PE is not enlarged.

Parameter	-CCF<number>
valid for	
	Specifies extended protection / special handling using a numeric value. <number> holds numerical value 1: AutoCAD 2011 - ARX files (plug-ins)

Parameter	-CCH
valid for	
	Prevents all global hooking in a protected application.

Parameter	-CCI
valid for	
	Allows to use the protected application in a way that the added protection does not change the eventual existing loading sequence for a protected application file. The import directory of the protected application is not changed. This replaces the option "-ccm" which is no longer required.

Parameter	-CCK
valid for	
	does not explicitly unload DLL, Windows XP only.

Parameter	-CCO
valid for	
	Activates special handling for ActiveX / OCX images.

Parameter	-CCQ
valid for	
	Clears license use for the protected applications not when WM_QUIT is called but with the call of ExitProcess().

Parameter	-CCR
valid for	
	Deactivates the renaming of sections.

Parameter	-CCS
valid for	
	Specifies that all licenses must locate in the same *CmContainer* connected to the same PC as it was the case with the first license found.

Parameter	-CCT:<selector>
valid for	
	allows specifying the AxEngine by using <selector> as selected architecture. The entries cover: linux-armhf, embedded-micro, android-so.  -ctt:linux-armhf: specifies the AxEngine for the linux-armhf architecture (e.g. Raspberry).

Parameter	-CCX
valid for	
	States that also so-called mixed-mode assemblies are protected. This allows to encrypt .NET assemblies which cannot be encrypted using *AxProtector .NET*. Next to Win32 also Win64/x64 mixed-mode assemblies can be protected using the native *AxProtector*. The library wbcor32/64.dll is required to allow running the protected assembly.

Option	-CC[D[flags]],[[K],[S]]
valid for	

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-CCD

valid for 

Defines flags for dlopen when loading shared objects. The flags may be `or`'d using the Linux constants:

- `RTLD_LAZY` `0x00001`
- `RTLD_NOW` `0x00002`
- `RTLD_NOLOAD` `0x00004`
- `RTLD_DEEPBIND` `0x00008`
- `RTLD_GLOBAL` `0x00100` (not set means `RTLD_LOCAL`)

Option	-CD[C][H](K([CA][CM][WK]Fx[Py])

valid for

Encrypts a file 1:1 file and adds a header holding encryption information. These files can automatically decrypted by an automatically encrypted application.

Parameter	-CDC

valid for

Applies the file name extension from the `*.wbc` file.

Parameter	-CDH

valid for

Specifies that the license access is kept open when the player closes the file, and a handle is kept open. This option is valid for single, separate files.

Parameter	-CDK([CA]|[CM]|[WK])

valid for

Specifies the used licensing system:
`CA` uses *CmActLicense*
`CM` uses *CmDongle*

Parameter	-CDK([CA]|[CM]|[WK])F

valid for

Specifies the *Firm Code* (`x`) required for the encrypted application to be able to open the encrypted file.

Parameter	-CDK([CA]|[CM]|[WK])P

valid for

Specifies the *Product Code* (`y`) required to open the protected application. The *Firm Code* must be previously set. More than one *Firm Code* - *Product Code* pair can be set.

Option	-CI[H][N][D]

valid for

Encrypts explicitly defined source code fragments within the executable file to be used with *IxProtector*.

Parameter	-CIH

valid for 

Defines that Wupi`XXX` functions functions are NOT dynamically hooked.

Parameter	-CIN

valid for 

Defines that no error messages are displayed when an error occurs.

Parameter	-CIP

valid for

Defines that no pointer based IxProtector table is searched.

Option	-CP<l>

gilt für

Installs a cleanup mechanism deleting all created files and registry entries on exiting an application if it has been started on a *CmContainer*.
`<l>` , causes that all deleted entries are written to a log file into the directory where the application locates.

Option	-E[A(C||R)],[E(C||R)],[F],[M],[T],[U(S(C|R)[n]|R(C|R)[n]|I)])

Defines additional checks while encryption and decryption operations are performed.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-EA

valid for

Activates an *Activation Time* check (*CodeMeter* only) .

- `C` Checks if the Product Item Option *Activation Time* exists.
- `I` Ignores the Product Item Option *Activation Time* (*CodeMeter* only).
- `R` Requires the Product Item Option *Activation Time*.

Parameter	-EE

valid for

Activates an *Expiration Time* check.

- `C` Checks if the Product Item Option *Expiration Time* exists.
- `I` Ignores the Product Item Option *Expiration Time* (*CodeMeter* only).
- `R` Requires the Product Item Option *Expiration Time*.

Parameter	-EF

valid for

Activates the decrement of the *Firm Access Counter* (*CodeMeter* only).

Parameter	-EM

valid for

Activates an *Maintenance Period* check.

- `C` Checks, if the Product Item Option *Maintenance Period* exists.
- `R` Requires the Product Item Option *Maintenance Period*.

Parameter	-ET

valid for

Enforces an *Certified Time* after the *CmContainer* is activated.


This option requires an activated *Expiration Time*.

Parameter	-EU

valid for

Activates an *Unit Counter* check and the counter decrementing by the specified value `<n>`.

- `S` Checks and decrements at the start of the protected application only.
- `R` Checks and decrements on each runtime check.
  The option `R` includes the option `S`.

  For options S and R the following options exist:

  - `C` Checks whether the Product Item Option *Unit Counter* exists (default setting).
  - `R(R)` Requires the Product Item Option *Unit Counter*.
  - `<n>` specifies the decrement. The default setting is `0`.
- `I` Ignores the Product Item Option *Unit Counter* (*CodeMeter* only).


-`eurr2` activates a required *Unit Counter* on each runtime check and decrements it each time by the value of `2`.

Option	-RIDx[,y]

valid for

Specifies the number of RID variants (`x`) and traps (`y`).
If RID=0 is set automatically, then the default value of 256 is used.

Option	-RIDIXx[,y]

valid for

Specifies the number of RID variants (`x`) and traps (`y`) when using *IxProtector* (WUPI).
If RID is set to a value of `0`, automatically the default value (64/8) is applied.

Option	-G[o,l]][:"Marker",]]

valid for

© Copyright 2018 WIBU-SYSTEMS AG

Option	-G[o,l][:"Marker",][]

Excludes the specified range from the encryption.

`<o>` Defines the exclusion at the beginning of the range.

`<l>` Defines the length of the range to be excluded
`"Marker"` identifies the text marker within the source code identifying the beginning of the range to be excluded from the encryption.

Option	-FW

valid for 

Sets the minimum Firmware version in the encryption.

Option	-W[C[t]]E[t]][P][U[c]]

valid for 

Specifies the threshold of issued warnings.

Parameter	-WC[t]

valid for 

Specifies the threshold `<t>` in hours for the *Certified Time*.

Parameter	-WE[t]

valid for 

Specifies the threshold <t> in days for the *Expiration Time*.

Parameter	-WP

valid for 

Activates a warning if the *Usage Period* has not yet been activated.

Parameter	-WU[c]

valid for 

Specifies the threshold `<c>` in units for the *Unit Counter*.

6.4.3.4 Runtime Options

Option	-I

valid for

Specifies that the exception handling for plug-in DLL files is used.
This option exclusively works with DLL files. When the plug-in is loaded and no licensing system linked, the plug-in does not closes the complete application, and does not issue error messages.

Option	-L:xx

valid for 

Specifies the language of the user-defined message texts.
`cn`: sets Chinese
`de`: sets German
`fr`: sets French
`jp`: sets Japanese
`us`: sets English (default setting)

Option	-M[A][C][E][I][S][T][U][W[C|P|T|U][L][R][V]]:"msg"

valid for 

Specifies the output text of user messages of the protected application.
`"msg"` holds the string for the desired event.
Line breaks, inverted comma, tabs, etc. can be specified in the output text. Type `"\n"`, `"\r"`, `"\t"`, `"\"` in the string at the desired position.

Parameter	-MA

valid for 

Specifies the application name which is transferred to the server and displayed in *CodeMeter WebAdmin*. No standard name is set. If this option is not set, the internal name of the executable file is used.

© Copyright 2018 WIBU-SYSTEMS AG

Parameter	-MC

valid for 

Holds the text displayed in the system menu item 'About'.

Parameter	-ME

valid for 

Holds the text, if an error has occurred.

Parameter	-MI

valid for 

Holds the text displayed, if the required driver of the licensing systems is not installed.
The return value to the User Message DLL corresponds to the already existing WUPI error code `wibu::UpiErrorLicenseModuleNotLoaded`.

Parameter	-MS

valid for 

Holds the text displayed while the protected application is started.

Parameter	-MT

valid for 

Holds the text displayed, if the date of the *Expiration Time* has been reached

Parameter	-MU

valid for 

Holds the text displayed, if the *Unit Counter* has reached a value of 0.

Parameter	-MWC

valid for 

Holds the text displayed, if the time difference between *Certified Time* and System Time is too big. This option requires the activated option `-wc<t>`.

Parameter	-MWP

valid for 

Holds the text displayed on application start, if an existing *Usage Period* has not yet been activated.

Parameter	-MWT

valid for 

Holds the text displayed, if the end of the *Expiration Time* or the *Usage Period* has been reached.

Parameter	-MWU

valid for 

Holds the text displayed, if the *Unit Counter* is about to reach a value of 0.

Option	-U[:FileName]

valid for 

Calls the file `UserMsgXX.dll` where XX stands for a country placeholder, e.g. `De`, `Us`, etc..
`[:FileName]`
Calls the specified class for the message handling.
The class must be a secondary class to the `com.wibu.xpm.MessageHandler`. For example, `com.wibu.xpmSwingMessageHandler` as default standard message handler of the runtime package.

Option	-UM[:FileName]

valid for 

Calls the user-defined message assembly UserMsg if this assembly exists.
If `[:FileName]` is specified the implemented message assembly holds the name `<FileName>.dll`.

 Specify the name without a file extension `*.dll`.

Option	-UI

valid for 

© Copyright 2018 WIBU-SYSTEMS AG

Option	-UI

Implements the message assembly inline configured by an `*.ini` file.

Option	-UDT[C|A|:"<text>"]

valid for 

Sets user defined text in AxEngine CmAccess calls. If set, overrides user defined text set by a message DLL.

C	Use computer name
A	Use application name
"text"	Use specified text

Option	-O[:FileName]

valid for 

Specifies the path and the name of the encrypted destination file.

6.4.3.5 Operational Options

Option	-!

valid for 

Creates a command file (*.wbc).

Option	-V

valid for 

Activates the verbose mode.

 In the case of  use `-vn`.

Option	-#[File]

valid for 

Prints the logging to the specified `[File]`. This option exists next to automatic output to the AxProtector.log. `[//Documents and Settings\user]`.

 In the case of  on specifying a file name, please note that the relative path information always bases on the directory of the input assembly.

Option	-? or -h

valid for 

Shows the options in commandline mode.

Option	@cmds.wbc

valid for 

Adds a specified `*.wbc` file holding parameters for the automatic encryption of an executable file.

6.5 CodeMeter Core API: Licensed access to applications on the embedded device

CmEmbedded is the runtime counterpart of *CodeMeter License Server* in the embedded realm and allows direct access to licenses securely saved in a *CmContainer* from within your application.

The Application Programming Interface (API) required to perform license-related operations is fully compatible with the *CodeMeter Core API* of the *CodeMeter Software Development Kit*. However, depending on modules implemented only a subset of the functions can be used.

Please refer to the separate documentation of the *CodeMeter Core API* adapted *to CodeMeter Embedded.* Please download the webhelp-based documentation from the Wibu-Systems website.

© Copyright 2018 WIBU-SYSTEMS AG

6.5.1 Code Samples

Here you find some code samples to show how licensed access to applications on the embedded device for selected *CodeMeter Core* API functions.

6.5.1.1 Encrypting using CmCrypt2

```
void do_direct_crypt(void)
{
  CMACCESS2 cmAcc;
  HCMSysEntry hcmse;
  int res;
  unsigned int cbDest = 16;
  unsigned char pbDest[16];
  CMCRYPT2 cmCrypt;

  memset(&cmCrypt, 0, sizeof(cmCrypt));
  memset(&cmAcc, 0, sizeof(cmAcc));

  cmAcc.mflCtrl |= CM_ACCESS_NOUSERLIMIT;
  cmAcc.mulFirmCode = 100003;
  cmAcc.mulProductCode = 1;
  hcmse = CmAccess2(CM_ACCESS_LOCAL, &cmAcc);
  if(0 == hcmse) {
    /* No matching license entry or CmContainer found. */
    return;
  }

  cmCrypt.mcmBaseCrypt.mflCtrl |= CM_CRYPT_FIRMKEY;
  cmCrypt.mcmBaseCrypt.mflCtrl |= CM_CRYPT_AES;
  cmCrypt.mcmBaseCrypt.mulEncryptionCodeOptions |= CM_CRYPT_UCIGNORE;
  /* set a random number as Encryption Code (part of the key) */
  cmCrypt.mcmBaseCrypt.mulEncryptionCode = 12345;

  res = CmCrypt2(hcmse, CM_CRYPT_DIRECT_ENC, &cmCrypt, pbDest, cbDest);
  if(0 != res) {
    /* The number of en-/decrypted bytes was returned. */
  } else {
    /* Some errors occurred. */
  }
  CmRelease(hcmse);
}
```

6.5.1.2 Remote Programming

6.5.1.21 Create a Context File

```
/* Store the content of ctx_buf in an ASCII file with .WibuCmRaC ending. */
size_t create_context(CMUSHORT mask, CMUINT serial, CMUINT firm_code,
                      unsigned char *ctx_buf, size_t max_size)
{
  CMACCESS2 cm_access;
  HCMSysEntry hbox;
  int ret = 0;
  size_t written = 0;
  memset( &cm_access, 0, sizeof(cm_access) );
  cm_access.mflCtrl = CM_ACCESS_Force;
  cm_access.musBoxMask = mask;
  cm_access.mulSerialNumber = serial;
  hbox = CmAccess2( CM_ACCESS_LOCAL, &cm_access );
  if (NULL != hbox) {
    if (CmGetRemoteContextBuffer(hbox, 0, &firm_code, 1,
                                 ctx_buf, max_size, &written) != 1) {
       printf("Create context failed, err=%d\n", CmGetLastErrorCode());
    }
```


© Copyright 2018 WIBU-SYSTEMS AG

```
    CmRelease(hbox);
  }
  return written;
}
```

6.5.1.22 Exeute a CmContainer Update

```
/* You have to open the WibuCmRaU file and store the content in update_buf before */
int execute_update(unsigned char *update_buf, size_t size)
{
  CMACCESS2 cm_access;
  CMFAS cmfas_res;
  HCMSysEntry hsubsys;
  int ret = 0;
  size_t nresult = 0;
  memset( &cm_access, 0, sizeof(cm_access) );
  cm_access.mflCtrl = CM_ACCESS_SUBSYSTEM;
  hsubsys = CmAccess2( CM_ACCESS_LOCAL, &cm_access );
  if (NULL != hsubsys) {
    if (!CmExecuteRemoteUpdate(hsubsys, CMFAS_CTL_CMD_UPDATE, update_buf, size,
                               0, 0, & cmfas_res, 1, &nresult)) {
      printf("Execute update failed, result=%d\n", cmfas_res.mulResult);
      ret = -1;
    }
    CmRelease(hsubsys);
  }
  return ret;
}
```

6.5.1.23 Import a CmActLicense file

```
int import_cmact_license_file(char *filename)
{
  CMACCESS2 cm_access;
  HCMSysEntry cm_handle;
  void * file_contents = NULL;
  size_t file_bytesize = 0;
  CMFAS dummy_result;
  CMUINT dummy_result_count;
  memset( &cm_access, 0, sizeof(cm_access) );
  cm_access.mflCtrl = CM_ACCESS_SUBSYSTEM;
  cm_handle = CmAccess2( CM_ACCESS_LOCAL, &cm_access );
  if( NULL!=cm_handle ) {
    /* Load test file (WibuCmRaU/wbc) */
    /* allocated memory for file_contents and load content of file to his buffer */
    file_contents = load_file_content( filename, &file_bytesize );

    if (file_contents) {
      CmSetLastErrorCode( CMERROR_NO_ERROR );
      api_result = CmExecuteRemoteUpdate( cm_handle, CMFAS_CTL_CMD_UPDATE,
                                          file_contents, file_bytesize, NULL, 0,
                                          &dummy_result, 1, &dummy_result_count );
      error_code = CmGetLastErrorCode();
      if (!api_result) {
        printf("Update failed, res=%d, err=%d\n", api_result, error_code);
      }
      CmRelease(cm_handle);
      free_file_buffer(file_contents);
    }
  }
  return 0;
}
```

© Copyright 2018 WIBU-SYSTEMS AG

6.5.1.3 Access a network license

```
void lan_access(char *ip)
{
  CMACCESS2 cm_access;
  HCMSysEntry hlan;
  int ret = 0;
  memset(&cm_access, 0, sizeof(cm_access));
  cm_access.mulFirmCode = 10;
  cm_access.mulProductCode = 13;
  strncpy(cm_access.mszServerName, ip, sizeof(cm_access.mszServerName)-1);
  hlan = CmAccess2(CM_ACCESS_LAN, &cm_access);
  if (NULL != hlan) {
    /* Connection to server has been established */
    CmRelease(cm_handle);
  }
}
```


© Copyright 2018 WIBU-SYSTEMS AG

7 Appendix

This appendix includes how to build CmEmbedded from source 141 and a description of the communication of ASIC and SPI (Serial Peripheral Interface) 149.

7.1 How to build CmEmbedded from source

Directory source

```
|-- asn1
|-- base
|-- base_defs
|-- bci
|-- box
    |-- fileio
    |-- hid
            |-- android
                |-- linux
                |-- qnx
                |-- vxworks
                |-- windows
    |-- spi
                |-- linux
|-- cert
|-- cmact
        |-- adapter
                |-- file
                act_adapter_bindext.c
                act_adapter_dyndata.c
                act_adapter_licread.c
                act_adapter_licwrite.c
                act_adapter_util.c
                act_adapter_util.h
|-- cmact_lt
|-- cmake
|-- cmlan
|-- coreapi
|-- corecm
|-- crypto_algorithms
|-- file_handling
|-- pkcs7
|-- posix
|-- tools
|-- unittest
|-- wibucert
cm_config.h
cmact.cmake
CMakeLists.txt
CmEmbedded.def
CodeMeterCompact.h
CodeMeterEmbedded.h
CmEmbedded.def
w_version.h
```

cm_config.h

```
#ifndef W_CM_CONFIG_H
#define W_CM_CONFIG_H

/* Configure max number of internal handles */
#define HANDLE_STORE_SIZE 32

/* Configure max number of CM Container (CmDongle, CmAct, CmActLt) that can be managed by the library */
#define CM_CONTAINER_MAX 32

/* Configure general support for CodeMeter dongles */
#define ENABLE_DONGLE_SUPPORT ON

/* Configure support for CodeMeterAct */
#define ENABLE_CMACT ON

/* Configure support for CodeMeterAct UFC */
#define ENABLE_CMACT_LT ON
```

© Copyright 2018 WIBU-SYSTEMS AG

```
/* Configure support for CmLAN module */
#define ENABLE_CMLAN ON

/* Configure support for Runtime Bridge (depends on CmLAN module) */
#define ENABLE_CM_RUNTIME_BRIDGE ON

/* Enable or disable API functions       */
#define ENABLE_CM_GET_PUBLIC_KEY      ON
#define ENABLE_CM_CALCULATE_SIGNATURE ON
#define ENABLE_CM_VALIDATE_SIGNATURE  ON
#define ENABLE_CM_CHECK_SIGNATURE     ON
#define ENABLE_CM_GETLASTERRORTEXT2   ON
#define ENABLE_CM_CONVERT_STRING      ON
#define ENABLE_CM_CMCRYPT_ECIES       ON

/* Enable or disable legacy API functions       */
#define ENABLE_CM_ACCESS  ON
#define ENABLE_CM_CMCRYPT ON

#define ENABLE_CM_GETBOXCONTENTS    ON
#define ENABLE_CM_BOXENTRY          ON
#define ENABLE_CM_GETLASTERRORTEXT  ON
#define ENABLE_CM_GETVERSION        ON
#define ENABLE_CM_GETINFO_ENTRYINFO ON

#define ENABLE_CM_CORE_API_STUBS ON

#define ENABLE_CUSTOM_STRNSTR    ON
#define ENABLE_CUSTOM_GMTIME     ON
#define ENABLE_CUSTOM_TIME       OFF
#define ENABLE_CUSTOM_STRFTIME   OFF
#define ENABLE_CUSTOM_STRCASESTR ON
#define ENABLE_CUSTOM_STRCASECMP ON

#define ENABLE_CUSTOM_GETTIMEOFDAY ON
#define ENABLE_CUSTOM_USLEEP       ON
#define ENABLE_CUSTOM_RANDOM       ON

#define ENABLE_CUSTOM_OPEN    OFF
#define ENABLE_CUSTOM_CLOSE   OFF
#define ENABLE_CUSTOM_READ    OFF
#define ENABLE_CUSTOM_WRITE   OFF
#define ENABLE_CUSTOM_UNLINK  OFF
#define ENABLE_CUSTOM_DIRENT  OFF
#define ENABLE_CUSTOM_STAT    OFF
#define ENABLE_CUSTOM_SYSCONF ON

/* Configure License Cache feature */
#define ENABLE_LICENSE_CACHE ON

#define ENABLE_CMCORE_LOGGING OFF

#define CM_RUNTIME_MAJOR_VERS 6
#define CM_RUNTIME_MINOR_VERS 30
#define CM_RUNTIME_BUILD_NR   2262

#define ENABLE_LIB_CONSTR_DESTR ON

#define ON  1
#define OFF 0

/* CmDongle specific configuration options */
#if ENABLE_DONGLE_SUPPORT

#define ENABLE_HID_SUPPORT             ON
#define ENABLE_SPI_SUPPORT             ON
#define ENABLE_64k_CODEMTR_IO_SUPPORT  ON

/* Enable communication between CmEmbedded and a CmDongle */
#define ENABLE_ENCR_DONGLE_COMM        ON

#define ENABLE_BCI_LOGGING             OFF
#define BCI_LOG_BUF_SIZE               4096
#define BCI_LOG_BDT_TIME               ON

#if ENABLE_SPI_SUPPORT
#define ENABLE_SPI_LOGGING OFF
#define SPI_TIMEOUT_OFF OFF
#define SPI_MAX_TRANSFER_SIZE 4162
#endif
```


© Copyright 2018 WIBU-SYSTEMS AG

```
#define DEBUG_DUMP_TEMPLATES OFF

#endif /* ENABLE_DONGLE_SUPPORT */

/* CmAct and CmAct UFC specific configuration options */
#if ENABLE_CMACT || ENABLE_CMACT_LT

#define ENABLE_CMACT_CONVERT_WBC_TO_WBB ON
#define ENABLE_CMACT_ADAPTER_FILE       OFF
#define ENABLE_CMACT_ADAPTER_CUSTOM     ON

/* Adapter implementation for XMC 4xxx microcontrollers */
#define ENABLE_CMACT_ADAPTER_XMC_STATIC OFF
#define ENABLE_CMACT_ADAPTER_XMC        OFF

/* Load the adapter implementation from a shared library instead of using a compiled-in one: */
#define ENABLE_CMACT_ADAPTER_DYNAMIC    OFF
#define ENABLE_CMACT_PRELOADED_ADAPTER  OFF

#define CMACT_BINDING_EXTENSION_NAME    "FB13376PlugIn"
/* Maximum time difference (in sec) after which the timestamp is written into the dynamic data */
#define CMACT_SAVE_BOXTIME_THRESHOLD    300

#define USE_ADAPTER_TRACER              OFF
#endif /* ENABLE_CMACT || ENABLE_CMACT_LT */

/* CmLAN specific configuration options */
#if ENABLE_CMLAN
#define CMLAN_CONNECTION_POOL_MAX  16
#define CMLAN_TIMEOUT              10
#define CMLAN_ENABLE_LOGGING       OFF
#define CMLAN_SERVER_PORT          22350
#define CMLAN_LOG_BUF_SIZE         4096
#endif /* ENABLE_CMLAN */

#if ENABLE_CMCORE_LOGGING
#define CM_CORE_LOG_BUF_SIZE  4096
#define CM_CORE_LOG_MODE      0
#endif

/* Runtime Bridge specific configuration options */
#if ENABLE_CM_RUNTIME_BRIDGE
#define CM_RUNTIME_BRIDGE_PORT 22350
#endif

/* License Cache specific configuration options */
#if ENABLE_LICENSE_CACHE
#define ENABLE_SHARED_CM_LOCK 1
#endif

/* CodeMeter MicroEmbedded specific configuration options */
#if defined(ENABLE_MICRO_EMBEDDED)
#define ENABLE_WBC_SUPPORT OFF
#endif

#endif /* W_CM_CONFIG_H */
```

7.1.1 Building from Source Code

This section describes how to create the *CmEmbedded* libraries based on the source package.



Wibu-Systems natively supports NINJA (GNinja) and GNU (G"Unix Makefiles") as CMake generators.

For generating of, for example, Visual Studio Projects, Wibu-Systems does not guarantee that these will produce correct builds using CMake.

General steps

The creation of the *CmEmbedded* libraries based on the source package cover the following general steps

- Unpack *CmEmbedded* source.
- Create build folder outside the source tree.

© Copyright 2018 WIBU-SYSTEMS AG

7.1.1.1 Building on Windows

Win32/64

Requirements

- Microsoft Visual Studio 2005 or later
- Microsoft Windows SDK e.g. Version 7.1
- CMake

Build steps

- Open the command line window, for example, using `cmd.exe`.
- Set the Visual C ++ environment using `vcvarsall.bat`, for example: `C:\>"C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\vcvarsall.bat" x86`.
  Alternatively, one of the VS20XX x86/x64 native tools command prompt links can be used.
- Change to the previously created build folder.
- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel path/to/cm_embedded`
- ninja

VxWorks

Requirements

- VxWorks 6.9.x Installation

Build steps

- The supplied `WIBU-VxWorks.cmake` file in the CMake installation folder within `\cmake\modules\Platform`
- Open VxWorks Development Shell from the Windows Start menu or Workbench
- Change to the previously created build folder.
- `cmake -G"Unix Makefiles" -DCMAKE_BUILD_TYPE=MinSizeRel -`
- `DCMAKE_TOOLCHAIN_FILE=path/to/cm_embedded/cmake/toolchains/vxworks6.9-x86gnu.cmake path/to/cm_embedded`
- make

7.1.1.2 Building on Linux

x86/86_64

Requirements

- Build tool, such as GNU Make or ninja
- GCC or Clang Toolchain

Build steps

- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel path/to/cm_embedded`

Raspberry Pi

Requirements for Pi 1

- Build tools for x86/86_64
- GCC or Clang based ARM Cross toolchain
- If a GCC hardfloat based toolchain is used, the included toolchain file `linux-armv6raspi-gnu.cmake` can be used.

© Copyright 2018 WIBU-SYSTEMS AG

Build steps

- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel -DCMAKE_TOOLCHAIN_FILE=path/to/cm_embedded/cmake/toolchains/linux-armv6raspignu.cmake path/to/cm_embedded`
- ninja

Requirements for Pi 2 and Pi 3

- Build tools for x86/86_64
- GCC or Clang based ARM Cross toolchain
- If a GCC hardfloat based toolchain is used, the included toolchain file `linux-armv7hf-gnu.cmake` can be used. In general, this toolchain file can be used with ARMv7 Cortex-A based hardfloat systems.
- The Compiler `arm-linux-gnueabihf-gcc`.

Build steps

- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel -DCMAKE_TOOLCHAIN_FILE=path/to/cm_embedded/cmake/toolchains/linux-armv7hf-gnu.cmake path/to/cm_embedded`
- ninja

VxWorks

Requirements

- VxWorks 6.9.x Installation

Build steps

- Copy the supplied `WIBU-VxWorks.cmake` file into the CMake Installation folder (the file can be found in `cmake/Modules/Platform`).
- Open the VxWorks Development Shell from the Workbench, or set the environment variables in a Terminal by using the `wrenv.sh` script.
- Switch to the previously created folder.
- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel -DCMAKE_TOOLCHAIN_FILE=path/to/cm_embedded/cmake/toolchains/vxworks6.9-x86gnu.cmake path/to/cm_embedded`
- ninja

Android

Requirements

- Build tool, such as GNU Make or ninja
- Clang-based standalone toolchain generated using Android NDK toolset, e.g. `$ANDROID_NDK/build/tools/make_standalone_toolchain.py --arch arm --api 26 --install-dir /tmp/android-toolchain-26`
- CMake with minimum version 3.9.1

Build steps

- Set environment variable `ANDROID_STANDALONE_TOOLCHAIN`, e.g. `ANDROID_STANDALONE_TOOLCHAIN=/tmp/android-toolchain-26`
- `cmake -GNinja -DCMAKE_BUILD_TYPE=MinSizeRel -DCMAKE_TOOLCHAIN_FILE=path/to/cm_embedded/cmake/toolchains/android-armv7-clang.cmake path/to/cm_embedded`
- ninja

© Copyright 2018 WIBU-SYSTEMS AG

7.1.1.3 Building on macOS

Requirements

- Build tool Make
- AppleClang Toolchain
- XCode, Cmake

Build steps

- `/Applications/Cmake.app/Contents/bin/cmake -DCMAKE_BUILD_TYPE=MinSizeRel path/to/cm_embedded`
- `make`

7.1.2 CMake

With *CmEmbedded* 2.0, a new build system based on CMake, replacing the GNU make-based build system used up to and including version 1.7.

CMake is a cross-platform build system generator. Projects specify their build process with platform-independent CMake listfiles included in each directory of a source tree with the name `CMakeLists.txt`.

CMake is capable of creating a build system for a large number of IDEs and Build Tools. All build system generators supported by CMake can be used with *CmEmbedded*.



Wibu-Systems natively supports NINJA (GNinja) GNU (G"Unix Makefiles") as CMake generators.

Wibu-Systems recommends to use NINJA2 as a build tool.

CmEmbedded sets a version of 3.0.23 in advance. The build system is cross platform capable.

Structure

The `CMakeLists.txt` file located in the *CmEmbedded* root directory contains the two library `targets lib_static` and `lib_dynamic`. Each additional source directory also contains `CMakeLists.txt` files that generate the object files of the two libraries.

```
cm_embedded
(...)
    CMakeLists.txt     coreapi
        CMakeLists.txt (...)
```

Toolchain files

CMake tries to recognize the existing build environment of the host as best as possible. However, this only links reliably when there is a toolchain of the compiler, linker, and archive tool that can generate binaries for the host.

1. CMake
2. Ninja
3. CMake Documentation

If you want to use a cross-compile, where a build is for a different target than the system that it is being built from, CMake variables (toolchain files) are used. The CMake preset variables include information about the target system, the target architecture, the compiler, linker and archive tool.

CmEmbedded provides the following toolchain files:

- `android-armv7-clang.cmake`
- `android-x86_64-clang.cmake`
- `linux-aarch64-gnu.cmake`
- `linux-armv6raspi-gnu.cmake`
- `linux-armv7-gnu.cmake`
- `linux-armv7hf-gnu.cmake`
- `linux-x86-gnu.cmake`

© Copyright 2018 WIBU-SYSTEMS AG

- qnx-armv7-gnu.cmake
- qnx-x86-gnu.cmake
- vxworks6.9-armv7-gnu.cmake
- vxworks6.9-e500v2ppc-gnu.cmake
- vxworks6.9-x86-gnu.cmake
- vxworks7-armv7-gnu.cmake
- vxworks7-e500v2ppc-gnu.cmake
- vxworks7-x86_64-gnu.cmake
- vxworks7-x86-gnu.cmake
- windows-x86_64-msvc.cmake
- windows-x86-msvc.cmake

The files are named after the operating system, architecture, and compiler schema. If you want to translate *CmEmbedded* with a combination not listed here, you are free to create your own toolchain file. Generally, as soon as a cross compiler is used, a toolchain file should be used.

CMake options

The following CMake options are available:

Option	Description
CMAKE_BUILD_TYPE=Debug|Release|MinSizeRel|	specifies the build type (configuration) that will be built in this build tree:
CMAKE_TOOLCHAIN_FILE=	specifies the toolchain file from cm_embedded/cmake/toolchains. CMake evaluates the toolchain file before CMakeLists.txt.
CMAKE_INFO=ON|OFF	Output of additional CMake variables such as: • CMAKE_FIND_ROOT_PATH, • CMAKE_C_FLAGS, • CMAKE_BINARY_DIR or • CMAKE_INSTALL_PREFIX with default setting OFF.
BUILD_SHARED_LIBS=ON|OFF	If this option is disabled, no dynamic library is created (default ON).
CMAKE_POSITION_INDEPENDENT_CODE=ON|OFF	determines whether position independent executables or shared libraries will be created (default OFF).

7.1.3 CMake Example

The minimum *Core API* example initializes the *CmEmbedded* library, determines its version, and creates a handle on the local subsystem, releases it, and deinitializes the library.

The example assumes the following folder structure:

```
|-- include
|   |-- CodeMeterCompact.h
|-- lib
|   |-- libwibucmembedded.a
|-- demo
|   |-- CMakeLists.txt |   |-- demo.c
```

If you have an *CmEmbedded* binary package, you only have to copy the static library and the header to the appropriate location. For a source code package, you must first create the static library as described above, and then copy the library and header.

Binary package prerequisites

- The Windows binary package requires minimally Visual Studio 2012
- The Linux binary package requires a GLIBC version 2.4 or later.

© Copyright 2018 WIBU-SYSTEMS AG

Minimum Sample Code

```
#include <CodeMeterCompact.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

int main(void)
{
  HCMSysEntry hsub;
  CMACCESS2 cmacc;
  CMVERSION cmvers;

  CmStaticLibOnInit();

  CmGetInfo(NULL, CM_GEI_VERSION, &cmvers, sizeof(cmvers));

printf("Using CmEmbedded %hu.%02hu.%04hu\n",   cmvers.musVersion,cmvers.musSubVersion,
             cmvers.musBuild);

  memset(&cmacc, 0, sizeof(cmacc));

cmacc.mflCtrl |= CM_ACCESS_NOUSERLIMIT;
cmacc.mflCtrl |= CM_ACCESS_SUBSYSTEM;

hsub = CmAccess2(CM_ACCESS_LOCAL, &cmacc);

  if (hsub == NULL) {
    fprintf(stderr, "CmAccess2 to subsystem failed (e=%d)\n", CmGetLastErrorCode());         CmStaticLibOnExit();
return EXIT_FAILURE;
  }

  printf ("CmAccess2 to subsystem was successful\n");

  CmRelease(hsub);

  CmStaticLibOnExit();

  return EXIT_SUCCESS;
}
```

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.0.2)

project(CmEmbeddedDemo C)

include_directories(../include)

add_definitions(-D_CODEMETER_API_STATIC_LIB)

set(SOURCES demo.c)

include(${CMAKE_SYSTEM_NAME}.cmake)

find_library(CMEMBEDDED wibucmembedded
PATHS ../lib
NO_SYSTEM_ENVIRONMENT_PATH
NO_DEFAULT_PATH)

add_executable(${PROJECT_NAME} ${SOURCES})

target_link_libraries(${PROJECT_NAME} ${CMEMBEDDED} ${LIBS})
```

Linux.cmake

```
set(LIBS "-lrt -lpthread")
```

Windows.cmake

```
string(REPLACE "/MDd" "/MTd" CMAKE_C_FLAGS_DEBUG "${CMAKE_C_FLAGS_DEBUG}")
string(REPLACE "/MD" "/MT" CMAKE_C_FLAGS_RELEASE "${CMAKE_C_FLAGS_RELEASE}")
string(REPLACE "/MD" "/MT" CMAKE_C_FLAGS_MINSIZEREL "${CMAKE_C_FLAGS_MINSIZEREL}")
string(REPLACE "/MD" "/MT" CMAKE_C_FLAGS_RELWITHDEBINFO "${CMAKE_C_FLAGS_RELWITHDEBINFO}")

set(LIBS "ws2_32.lib setupapi.lib")
```


© Copyright 2018 WIBU-SYSTEMS AG

7.1.4 Create example

A new build folder is created on the same level as `lib`, `include` and `demo`. Then switch to this folder and run CMake.

- `cmake -GNinja ../demo`
- You can use ninja to create the example.
  `ninja`

7.2 CmDongle and SPI (Serial Peripheral Interface)

7.2.1 SPI (Serial Peripheral Interface) Integration Version 2

Interface functions and structures

The interface Version 2 between *CmEmbedded* and the hardware specific SPI Module is specified by `box/spi/spi.h`. It defines SPI error codes, a `struct spi_device` as used in *CmEmbedded*, and seven interface functions.

struct spi_device Structure

Elements	Description
`device_handle`	represents, depending on the target system, a file descriptor, a HANDLE, or a generic pointer.
`user_defined`	points to a user defined buffer and the length of the buffer is `user_def_len`. In the sample implementation for Linux/RaspberryPi `user_defined` is used to hold the device path.
`speed_hz`	member is set by *CmEmbedded* to the maximum clock rate in Hz at which can be communicated with the CmAsic at this point. This value can change before and after enumeration.
`features`	member can be STANDARD or TIMEOUT_EXTENSION. The latter has to be set in the enumeration function `io_spi_enum_devices()` if special production features have been chosen, namely a configurable, larger timeout. Keep in mind that a larger timeout interval can result in severe performance issues.
`timeout_fallback_ms` and `timeout_configure_ms`	have to be set in the enumeration function as well. They are device specific (fallback value is production parameter and may be up to 16 seconds) and system specific (`timeout_configure_ms` cannot be changed for a running *CmEmbedded* instance or exchanged between concurrently running instances). Values are in milliseconds.
`timeout_current_ms`	holds the currently assumed timeout interval for this device.
`protocol_version`	holds the version of the protocol that the CmAsic communicates with, as communicated in the handshake.
`max_payload`	holds the maximum size for the SPI Payload, which can be negotiated via the handshake (since protocol Version 2).
`next`	points to the next device in a list.

InterfaceFunctions

Function	Description
`io_spi_device_init`	initializes the referenced `struct spi_device` to all zeros and allocates buffer of size `user_def_Len` to `user_defined`.
`io_spi_device_definite`	frees the device's `user_defined` buffer.
`io_spi_open`	opens the specified device and sets SPI parameters such as mode, bits per word (8), clock rate, MSB first. Returns `TRUE=1` if the device was successfully opened,

© Copyright 2018 WIBU-SYSTEMS AG

Function	Description
	`FALSE=0` otherwise.
`io_spi_close`	closes the referenced `spi_device`. Returns `TRUE=1` on success, `FALSE=1` on failure,
`io_spi_transfer_bytes`	transfers a byte-stream of given length to the specified device and receives the duplex bytes. Returns `TRUE=1` if the complete transfer was successful, `FALSE=0` if not.
`io_spi_enum_devices`	creates a list of connected `spi_devices`. Returns a pointer to the first device of the list.
`io_spi_free_devices_list`	frees a list of devices that starts with the given device.
`io_spi_last_error`	is defined in `spi.c` and is used to handle low level errors.

Error Codes

Code	Description
`SPI_ERROR_NO_ERROR`	set, if no error occurred.
`SPI_ERROR_DEVICE_OPEN_FAILED`	set, if an error during device opening occurs, such as the device doesn't exist.
`SPI_ERROR_DATA_WRONG`	set, if the function parameters are set incorrectly.

SPI Parameters and Chip Specifications

Defaults

All lines are `ACTIVE_LOW`, so `HIGH` when idle.

Element	Description
Mode, BPW, Bit justification	The SPI mode the *CodeMeter* Chip operates is usually Mode 3 (Polarization: `ACTIVE_LOW`, Phase: Read value at rising clock signal edge, change data at falling edge). The number of bits per word / word-length is the standard 8 bpw and the MSB is sent first.
Chip Select line	The Chip Select line has to be deselected after every block of data.
Clock frequency	The maximum SPI clock frequency the *CmAsic* can handle is currently 11MHz. Communication starts at a lower rate (before handshake ascertains actual maximum), which should be no more than 1 MHz.
Baudrate	After every transmitted byte there has to follow a break of at least one and a half clock cycles. This reduces the maximum data transmission rate of theoretical 11 MBit/s to actual 9.26 MBit/s.
Pullup resistors	The *CmAsic* has pull-up resistors activated for MOSI,CLK, and CS. The Master has to provide the pull-up for the MISO line.

Raspberry Pi 2 as Master

Pinout

The following physical pins on the Raspberry Pi 2 have to be connected to the *CmAsic*:

												CS							
2	4	6	8	10	12	14	16	18	20	22	**24**	26	28	30	32	34	36	38	40
.1	3	5	7	9	11	13	15	17	**19**	**21**	**23**	25	27	29	31	33	35	37	**39**
3.3v									**MOSI**	**MISO**	**SCLK**								**GND**

Load SPI Kernel module

© Copyright 2018 WIBU-SYSTEMS AG

In order for the Raspi (running Raspbian) to control these pins as SPI Master, the SPI Driver has to be loaded (disabled by default). To enable it, remove the blacklisting for `spi-bcm2708` in `/etc/modprobe.d/raspi-blacklist.conf` and add or uncomment `dtparam=spi=on` in `/boot/config.txt. Reboot`.

Alternatively, use the 'Advanced Options' in `sudo raspi-config`.

spidev.h commands

The demo low level SPI communication of *CmEmbedded* on the Raspberry Pi employs the `ioctl()` commands as specified in `spidev.h`.

Change Parameters during Runtime via Environment Variables

Export the following environment variables on your RaspberryPi2 to set various SPI Parameters in the Raspi SPI Demo Module:

- default `SPI_PARAM_BITS_PER_WORD=8`
  export `SPI_PARAM_BITS_PER_WORD=8`
- default `SPI_PARAM_MODE=3`
  export `SPI_PARAM_MODE=3`
- default `SPI_PARAM_LSB_FIRST=0`
  export `SPI_PARAM_LSB_FIRST=0`
- default `SPI_PARAM_CS_CHANGE=1`
  export `SPI_PARAM_CS_CHANGE=1`
- default `SPI_PARAM_DEV_PATH=/dev/spidev0.0` for RaspberryPi2
  export `SPI_PARAM_DEV_PATH=/dev/spidev0.0`
- default `SPI_PARAM_TIMEOUT_EXT_FALLBACK=500` means `STANDARD` *CmAsic*,
  `500 < x < 16000` means `TIMEOUT_EXTENSION`; in milliseconds
  export `SPI_PARAM_TIMEOUT_EXT_FALLBACK=500`
  or e.g. export `SPI_PARAM_TIMEOUT_EXT_FALLBACK=10000`
- no default `SPI_PARAM_CONF_TIMEOUT_MS`, only valid when `TIMEOUT_EXTENSION`
  export `SPI_PARAM_CONF_TIMEOUT_MS=1000`

Reset Line

The *CmAsic* makes use of another line in, additionally to the SPI pins. This extra line is the Chip Reset line, which by default is on `HIGH`. When toggled to `LOW` and back again, the chip is being cold reset.

Building CmEmbedded library with SPI support

Set CMake Variable `ENABLE_SPI_SUPPORT` to `ON`.

To control the maximum SPI transfer size, namely the maximum number of bytes that can be transferred in one transfer "burst", set the CMake Variable `SPI_MAX_TRANSFER_SIZE`. This also controls the size of the heap allocated communication buffers, and may be relevant if you're working with little memory. This option is only relevant starting with protocol Version 2. If the variable is not set or protocol Version 1 is active, the maximum transfer size is 512 bytes.

usleep

If working on an operating system for which *CmEmbedded* doesn't implement a POSIX-like usleep function, or on a platform without operating system, such a function has to be provided by the user. In this case, the build configuration switch `CONF_IMPLEMENT_OWN_USLEEP` has to be set to `ON`. Then a user-implemented usleep function can be linked with the library.

7.2.2 SPI (Serial Peripheral Interface) Protocol Specification Version 2

This protocol Version 2 describes the communication between *CmEmbedded* and the *CmAsic* via the SPI bus.

It is available on request at Wibu-Systems.

© Copyright 2018 WIBU-SYSTEMS AG

8 Glossary

Term	Description
AxProtector	Automatic protection of applications using *AxProtector* as secure basic protection without changing the source code including runtime checks, effective anti-debug mechanisms, modification of resources, and locking of *CmContainer* if crack attempts are detected. Available for different project types and as a commandline version.
CmActLicense	Completely software-based variant of the protection and licensing technology *CodeMeter*. Licenses are bound to an individual computer.
CmBoxPgm	Commandline tool to create, edit, and delete licenses and their components (*Firm Item*, *Product Item*, and *Product Item Options*) in *CmContainer*. You can also apply scripts and batch files for mass production and test automation. Programming is simultaneously applied in one passe to several *CmContainer*.
CmContainer	Summarizing notion for describing the license carriers of both *CodeMeter* variants. *CmDongle* in the case of the hardware-based licensing system and *CmActLicense* n the case of the software-based licensing system.
CmDongle	Hardware-based variant of the protection and licensing technology *CodeMeter*. Available in many form factors for a variety of interfaces.
CmDust	The *CodeMeter* Enduser Support Tool logs important system and *CodeMeter* settings and helps Wibu-Systems Support to find remedies for eventually occurring errors.
CmFAS	*see CodeMeter Field Activation Service*
cmu	Commandline alternative to perform many *CodeMeter Control Center* functions (*CodeMeter Universal Support Tool*).
CodeMeter API Guide	Graphical tool to generate source code fragments. You create and test API functions with all related parameters and necessary structures for the programming language of your choice. Currently, the programming languages C, C++, C#, CB6, VB.NET, Delphi, and Java are supported.
CodeMeter Field Activation Service	see *File-based Remote Programming*
CodeMeter Control Center	*CodeMeter Control Center* provides the protected software to access the *CodeMeter* runtime environment. It displays information on connected *CmContainer*, and presents options to configure connected *CmContainer*. Moreover, an assistant creates license request files and imports license update files (*CmFAS Assistant*).
CodeMeter License Central	Ticket-based system for creating, managing, and delivering licenses for software and digital content. Available in a *Desktop* and an *Internet* Edition.
CodeMeter License Editor	Graphical tool allowing you to create, edit or delete licenses and their components (*Firm Item*, *Product Item*, and *Product Item Options*) in *CmDongles*. Next to programming of locally connected *CmCongles* also file-based remote programming (*CodeMeter Field Activation Service*, CmFAS) is supported. Suitable for testing license strategies.
CodeMeter License Server	Runtime environment (`CodeMeter.exe`) for the protection and licensing technology *CodeMeter*.
CodeMeter Start Center	Start screen tool to access and open most of the *CodeMeter* applications and tools.
CodeMeter WebAdmin	Graphical *CodeMeter* tool displaying information on connected *CmContainer* and related license entries in a browser. In addition, configuration and analyzing options for the *CodeMeter* runtime environment (*CodeMeter License Server*) are provided.
CodeMeter	Wibu-Systems' technology for protecting and licensing of software and digital content.
File-based Remote Programming	Remote updating a *CmContainer* requires some information on the *CmContainer* to be reprogrammed. This information is safely stored and transferred in a context file, i.e. `*.WibuRaC` file (license request file). Based on this license request file use the *CodeMeter* programming tools to create an update file (`*.WibuRaU`) (license update). Subsequently, this file is safely transferred into the *CmContainer*. In addition, on creating the *.WibuRaU file automatically also a `*.WibuRaM` file is created which maps the *CmContainer* content at the time the licenses have been updated. An CmFAS *Assistant* in *CodeMeter Control Center* supports the licensee when updating licenses.
Firm Code	The *Firm Code* presents a unique number each licensor receives from Wibu-Systems. It ensures that each licensor is individually identified when protecting and licensing software or digital content.
Firm Item	Logical and hierarchical item level in the *CmContainer*. The *Firm Item* level holds entries which are unique for each licensor and includes the individual *Firm Code*.
Firm Key	Secret key which influences almost all encryption and decryption processes of licenses, their authentication, and the creation, editing and deleting of license entries at the level of *Product Items*. The *Firm Key* is initially shipped with the *Firm Security Box*.
Firm Security Box	Master *CmDongle* which allows to program other *CmContainer*. The FSB is unique for each licensor.
FSB	see *Firm Security Box*
HIP	*High Level Programming API* see *Programming API*
IFI	see *Implicit Firm Item*

© Copyright 2018 WIBU-SYSTEMS AG

Term	Description
Implicit Firm Item	The *Implicit Firm Item* level in the *CmContainer* features the same characteristic as usual *Firm Items*). It simply has some distinct features. While all other level are characterized by the existence of an exclusive *Firm Code* which is unique for each licensor, the *Implicit Firm Item* level has the *Firm Code* of 0. This implies that each owner of the *CmContainer* has licensor privileges for the *Implicit Firm Item* level including write access.
IxProtector	Individual advanced protection technology applied for software and digital content. 'Real' source code fragments are encrypted and decrypted by interfaces (*Software Protection API*, WUPI) and security mechanisms. Suited to implement modular software protection.
Core API	Powerful interface to communicate with *CmContainer* at runtime of *CodeMeter License Server*. All other APIs and protection mechanisms (*AxProtector*, *IxProtector*, *Software Protection API* WUPI) base on *Core API* functions. Thus using this interface complements existing protection options (encryption and decryption of data, personalization, reading additional data).
License Activation	see File-based Remote Programming
License Information File (`*.WibuCmLIF`)	This file corresponds for *CmActLicense* to an empty license container however holds specifications on binding schemes and additional activation options to be used for unique binding of a license to the computer or the device.
Update File (license update) (`*.WibuCmRaU`)	The Update File for a *CmContainer* valid only for a single unique *CmContainer* can be imported only once.
Context File (license request) (`*.WibuCmRaC`)	The Context File of a *CmContainer* mirroring the as-is status of license entries serves as basis for license updating in the process of file-based (remote) programming.
SmartBind	Binding scheme used in *CmActLicense* licensing system optimizes assuring the validity of *CmActLicense* licenses, if hardware properties of the PC change to which the licenses are bound.
PIO	see *Product Item Options*
Product Code	The *Product Code* represents a number free to choose and identifies the products to be protected and licensed.
Product Item Options (PIO)	License entries at the *Product Item* level. They hold the *Product Code* also further options defining the actual characteristics of a license, such as, how many licenses may be simultaneously used in a network, how long a license is valid, which functions are accessible and billed, etc. Moreover, several other data fields are available holding additional binary information and differ in their access privileges. These optional characteristics are combinable in a variety of ways, and constitute the basis for the mapping of any imaginable license strategy.
Product Item	Logical hierarchical entry level in a *CmContainer* below the *Firm Item* level. At the *Product Item* level you find the single license entries, i.e. the *Product Codes* and further *Product Item Options*.
Programming API	This class-oriented interface allows you to access any object or process required to program or organize license entries in a *CmContainer*, and features extended customizing for the integration of *CodeMeter* into own applications. The *Programming API* is available for many programming languages.
Software Protection API	Interface which decrypts segments protected by *IxProtector* at runtime available as WUPI (WIBU *Universal Protection Interface*). It is lean, comprises only a few but essential functions, and is standardized and applicable for a variety of programming languages.
Wibu Universal Protection Interface	see *Software Protection API*
WUPI	see *Wibu Universal Protection Interface*

© Copyright 2018 WIBU-SYSTEMS AG

9 Version

CmEmbedded Guide Version 2.30, 11/13/2018.

Copyright © 2007-2018

by WIBU-SYSTEMS AG, Karlsruhe / Germany

All rights reserved.

Wibu-Systems contact information:

Address:	WIBU-SYSTEMS AG Rueppurrer Strasse 52-54 D-76137 Karlsruhe, Germany
Phone:	+49 (0)-721-93172-0
Internet:	http://www.wibu.com
E-mail:	support@wibu.com

Flot

Project	Runtime component CodeMeter WebAdmin (CodeMeter License Server)
Version	0.8.1 (flot)
Operating system	Windows, macOS, Linux

```
Copyright (c) 2007-2013 IOLA and Ole Laursen

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use,
copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following
conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

JQuery

Project	Runtime component CodeMeter WebAdmin (CodeMeter License Server)
Version	v1.11.3 (jQuery)
Operating system	Windows, macOS, Linux

```
Copyright jQuery Foundation and other contributors, https://jquery.org/

This software consists of voluntary contributions made by many
individuals. For exact contribution history, see the revision history
available at https://github.com/jquery/jquery-ui

The following license applies to all parts of this software except as
documented below:

====

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
```


© Copyright 2018 WIBU-SYSTEMS AG

```
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

====

Copyright and related rights for sample code are waived via CC0. Sample
code is defined as all source code contained within the demos directory.

CC0: http://creativecommons.org/publicdomain/zero/1.0/

====
All files located in the node_modules and external directories are
externally maintained libraries used by this software which have their
own licenses; we recommend you read them, as their terms may differ from
the terms above.
```

go-macaron framework

Project	Runtime component CodeMeter WebAdmin (CodeMeter License Server)
Version	1.8.3 (go)
Operating System	Windows, macOS, Linux

```
go-macaron/macaron is licensed under Apache License 2.0

How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate
notice, with the fields enclosed by brackets "[]" replaced with your own
identifying information. (Don't include the brackets!) The text should be
enclosed in the appropriate comment syntax for the file format. We also
recommend that a file or class name and description of purpose be included on
the same "printed page" as the copyright notice for easier identification within
third-party archives.

   Copyright 2014 The Macaron Authors

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

     http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
limitations under the License.

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and
distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright
```

© Copyright 2018 WIBU-SYSTEMS AG

owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

© Copyright 2018 WIBU-SYSTEMS AG

You must give any other recipients of the Work or Derivative Works a copy of this License; and
You must cause any modified files to carry prominent notices stating that You changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

© Copyright 2018 WIBU-SYSTEMS AG

```
END OF TERMS AND CONDITIONS
```

golang.org (/x/sys / and /x/crypto)

Project	Runtime component CodeMeter WebAdmin (CodeMeter License Server)
Version	1.8.3 (go)
Operating System	Windows, macOS, Linux

```
Copyright (c) 2009 The Go Authors. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
* Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

libusb Linux

Project	Runtime component Linux - CodeMeter License Server
Version	1.0-1.0.20
Operating system	Linux

The programming library serves to read and write USB devices on Linux.

```
                  GNU LESSER GENERAL PUBLIC LICENSE
                       Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

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

                            Preamble

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

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

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
```


© Copyright 2018 WIBU-SYSTEMS AG

you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

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

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

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

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

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

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

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

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

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

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

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

The precise terms and conditions for copying, distribution and

© Copyright 2018 WIBU-SYSTEMS AG

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

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

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

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

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

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

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

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

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

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

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

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

These requirements apply to the modified work as a whole. If

© Copyright 2018 WIBU-SYSTEMS AG

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

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

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

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

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

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

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

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

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

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

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

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

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

6. As an exception to the Sections above, you may also combine or

© Copyright 2018 WIBU-SYSTEMS AG

link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

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

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

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

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

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

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

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

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

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

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

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

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

© Copyright 2018 WIBU-SYSTEMS AG

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

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

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

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

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

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

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

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

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

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

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

© Copyright 2018 WIBU-SYSTEMS AG

```
                            NO WARRANTY

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

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

           How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

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

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

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

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301  USA

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

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:
  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

The code of the programming library **libusb** can be downloaded at the project website http://www.libusb.org/. On request the source can also be electronically obtained free of charge contacting Wibu-Systems (support@wibu.com).

libcurl

Project	Runtime component CodeMeter License Server
Version	7.28.0

© Copyright 2018 WIBU-SYSTEMS AG

Operating system	Windows, macOS, Linux

```
COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2017, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS
file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee
is hereby granted, provided that the above copyright notice and this permission notice appear in
all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT
NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE
FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising
or otherwise to promote the sale, use or other dealings in this Software without prior written
authorization of the copyright holder.
```

Google Protocol Buffers

Project	Runtime component	SDK component AxProtector
Version	2.6.1	2.6.1
Operating system	Windows, macOS, Linux	Windows, macOS, Linux

```
This license applies to all parts of Protocol Buffers except the following:

  - Atomicops support for generic gcc, located in
    src/google/protobuf/stubs/atomicops_internals_generic_gcc.h.
    This file is copyrighted by Red Hat Inc.

  - Atomicops support for AIX/POWER, located in
    src/google/protobuf/stubs/atomicops_internals_power.h.
    This file is copyrighted by Bloomberg Finance LP.

Copyright 2014, Google Inc.  All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
    * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it.  This code is not
standalone and requires a support library to be linked with it.  This
support library is itself covered by the above license.
```

© Copyright 2018 WIBU-SYSTEMS AG

ASN.1 Compiler

Project	Runtime component CodeMeter License Server	SDK component Programming API (HIP)
Version	1.0	1.0
Operating system	Windows, macOS, Linux	Windows, macOS, Linux

```
Copyright (c) 2003-2016  Lev Walkin <vlm@lionet.info> and contributors.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

Bootstrapper dotnetInstaller

Project	Runtime component dotnetInstaller	SDK component dotnetInstaller
Version	2.3	2.3
Operating system	Windows	Windows

```
MIT License (MIT)

Copyright (c) 2009-2012 Davide Icardi, Daniel Doubrovkine and Contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and
associated documentation files (the "Software"), to deal in the Software without restriction,
including without limitation the rights to use, copy, modify, merge, publish, distribute,
sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT
NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT
OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

LLVM compiler and toolchain technology

Project	SDK component AxProtector
Version	3.8.0
Operating system	Windows, macOS, Linux

```
==============================================================================
LLVM Release License
==============================================================================
```


© Copyright 2018 WIBU-SYSTEMS AG

```
University of Illinois/NCSA
Open Source License

Copyright (c) 2003-2015 University of Illinois at Urbana-Champaign.
All rights reserved.

Developed by:

    LLVM Team

    University of Illinois at Urbana-Champaign

    http://llvm.org

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal with
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is furnished to do
so, subject to the following conditions:

    * Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimers.

    * Redistributions in binary form must reproduce the above copyright notice,
      this list of conditions and the following disclaimers in the
      documentation and/or other materials provided with the distribution.

    * Neither the names of the LLVM Team, University of Illinois at
      Urbana-Champaign, nor the names of its contributors may be used to
      endorse or promote products derived from this Software without specific
      prior written permission.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
CONTRIBUTORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS WITH THE
SOFTWARE.

==============================================================================
Copyrights and Licenses for Third Party Software Distributed with LLVM:
==============================================================================
The LLVM software contains code written by third parties.  Such software will
have its own individual LICENSE.TXT file in the directory in which it appears.
This file will describe the copyrights, license, and restrictions which apply
to that code.

The disclaimer of warranty in the University of Illinois Open Source License
applies to all code in the LLVM Distribution, and nothing in any of the
other licenses gives permission to use the names of the LLVM Team or the
University of Illinois to endorse or promote products derived from this
Software.

The following pieces of software have additional or alternate copyrights,
licenses, and/or restrictions:

Program             Directory
-------             ---------
Autoconf            llvm/autoconf
                    llvm/projects/ModuleMaker/autoconf
Google Test         llvm/utils/unittest/googletest
OpenBSD regex       llvm/lib/Support/{reg*, COPYRIGHT.regex}
pyyaml tests        llvm/test/YAMLParser/{*.data, LICENSE.TXT}
ARM contributions   llvm/lib/Target/ARM/LICENSE.TXT
md5 contributions   llvm/lib/Support/MD5.cpp llvm/include/llvm/Support/MD5.h
```

ASM Java Program Library

Project	SDK component AxProtector
Version	5.1
Operating system	Windows, macOS, Linux

```
Copyright (c) 2000-2011 INRIA, France Telecom
```

© Copyright 2018 WIBU-SYSTEMS AG

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holders nor the names of its
   contributors may be used to endorse or promote products derived from
   this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

Apache Commons

Project	SDK component AxProtector Java; AxProtector .NET (.NET 2.0 Standard)
Version	cli 1.3.1, io 2.4, lang 3-3.4
Operating system	Windows, macOS, Linux

Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
      otherwise, or (ii) ownership of fifty percent (50%) or more of the
      outstanding shares, or (iii) beneficial ownership of such entity.

      "You" (or "Your") shall mean an individual or Legal Entity
      exercising permissions granted by this License.

      "Source" form shall mean the preferred form for making modifications,
      including but not limited to software source code, documentation
      source, and configuration files.

      "Object" form shall mean any form resulting from mechanical
      transformation or translation of a Source form, including but
      not limited to compiled object code, generated documentation,
      and conversions to other media types.

© Copyright 2018 WIBU-SYSTEMS AG

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must

© Copyright 2018 WIBU-SYSTEMS AG

include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

© Copyright 2018 WIBU-SYSTEMS AG

```
      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

MIT

Project	SDK component AxProtector .NET (.NET 2.0 Standard)
Version	.NET Standard 2.0
Operating system	Windows, macOS, Linux

```
The MIT License (MIT)
Copyright (c) .NET Foundation and Contributors

All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or
substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

Public Suffix List

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Version	https://publicsuffix.org/
Operating System	Windows, macOS, Linux

```
Mozilla Public License Version 2.0
==================================

1. Definitions
--------------
```

© Copyright 2018 WIBU-SYSTEMS AG

1.1. "Contributor"
    means each individual or legal entity that creates, contributes to
    the creation of, or owns Covered Software.

1.2. "Contributor Version"
    means the combination of the Contributions of others (if any) used
    by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
    means Covered Software of a particular Contributor.

1.4. "Covered Software"
    means Source Code Form to which the initial Contributor has attached
    the notice in Exhibit A, the Executable Form of such Source Code
    Form, and Modifications of such Source Code Form, in each case
    including portions thereof.

1.5. "Incompatible With Secondary Licenses"
    means

    (a) that the initial Contributor has attached the notice described
        in Exhibit B to the Covered Software; or

    (b) that the Covered Software was made available under the terms of
        version 1.1 or earlier of the License, but not also under the
        terms of a Secondary License.

1.6. "Executable Form"
    means any form of the work other than Source Code Form.

1.7. "Larger Work"
    means a work that combines Covered Software with other material, in
    a separate file or files, that is not Covered Software.

1.8. "License"
    means this document.

1.9. "Licensable"
    means having the right to grant, to the maximum extent possible,
    whether at the time of the initial grant or subsequently, any and
    all of the rights conveyed by this License.

1.10. "Modifications"
    means any of the following:

    (a) any file in Source Code Form that results from an addition to,
        deletion from, or modification of the contents of Covered
        Software; or

    (b) any new file in Source Code Form that contains any Covered
        Software.

1.11. "Patent Claims" of a Contributor
    means any patent claim(s), including without limitation, method,
    process, and apparatus claims, in any patent Licensable by such
    Contributor that would be infringed, but for the grant of the
    License, by the making, using, selling, offering for sale, having
    made, import, or transfer of either its Contributions or its
    Contributor Version.

1.12. "Secondary License"
    means either the GNU General Public License, Version 2.0, the GNU
    Lesser General Public License, Version 2.1, the GNU Affero General
    Public License, Version 3.0, or any later versions of those
    licenses.

1.13. "Source Code Form"
    means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
    means an individual or a legal entity exercising rights under this
    License. For legal entities, "You" includes any entity that
    controls, is controlled by, or is under common control with You. For
    purposes of this definition, "control" means (a) the power, direct
    or indirect, to cause the direction or management of such entity,
    whether by contract or otherwise, or (b) ownership of more than
    fifty percent (50%) of the outstanding shares or beneficial
    ownership of such entity.

© Copyright 2018 WIBU-SYSTEMS AG

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license:

(a) under intellectual property rights (other than patent or trademark) Licensable by such Contributor to use, reproduce, make available, modify, display, perform, distribute, and otherwise exploit its Contributions, either on an unmodified basis, with Modifications, or as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer for sale, have made, import, and otherwise transfer either its Contributions or its Contributor Version.

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution become effective for each Contribution on the date the Contributor first distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under this License. No additional rights or licenses will be implied from the distribution or licensing of Covered Software under this License. Notwithstanding Section 2.1(b) above, no patent license is granted by a Contributor:

(a) for any code that a Contributor has removed from Covered Software; or

(b) for infringements caused by: (i) Your and any other third party's modifications of Covered Software, or (ii) the combination of its Contributions with other software (except as part of its Contributor Version); or

(c) under Patent Claims infringed by Covered Software in the absence of its Contributions.

This License does not grant any rights in the trademarks, service marks, or logos of any Contributor (except as may be necessary to comply with the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to distribute the Covered Software under a subsequent version of this License (see Section 10.2) or under the terms of a Secondary License (if permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its Contributions are its original creation(s) or it has sufficient rights to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under applicable copyright doctrines of fair use, fair dealing, or other equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any

© Copyright 2018 WIBU-SYSTEMS AG

Modifications that You create or to which You contribute, must be under the terms of this License. You must inform recipients that the Source Code Form of the Covered Software is governed by the terms of this License, and how they can obtain a copy of this License. You may not attempt to alter or restrict the recipients' rights in the Source Code Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code Form, as described in Section 3.1, and You must inform recipients of the Executable Form how they can obtain a copy of such Source Code Form by reasonable means in a timely manner, at a charge no more than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this License, or sublicense it under different terms, provided that the license for the Executable Form does not attempt to limit or alter the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice, provided that You also comply with the requirements of this License for the Covered Software. If the Larger Work is a combination of Covered Software with a work governed by one or more Secondary Licenses, and the Covered Software is not Incompatible With Secondary Licenses, this License permits You to additionally distribute such Covered Software under the terms of such Secondary License(s), so that the recipient of the Larger Work may, at their option, further distribute the Covered Software under the terms of either this License or such Secondary License(s).

3.4. Notices

You may not remove or alter the substance of any license notices (including copyright notices, patent notices, disclaimers of warranty, or limitations of liability) contained within the Source Code Form of the Covered Software, except that You may alter any license notices to the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Software. However, You may do so only on Your own behalf, and not on behalf of any Contributor. You must make it absolutely clear that any such warranty, support, indemnity, or liability obligation is offered by You alone, and You hereby agree to indemnify every Contributor for any liability incurred by such Contributor as a result of warranty, support, indemnity or liability terms You offer. You may include additional disclaimers of warranty and limitations of liability specific to any jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Software due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be placed in a text file included with all distributions of the Covered Software under this License. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically if You fail to comply with any of its terms. However, if You become compliant, then the rights granted under this License from a particular Contributor are reinstated (a) provisionally, unless and until such Contributor explicitly and finally terminates Your grants, and (b) on an ongoing basis, if such Contributor fails to notify You of the non-compliance by some reasonable means prior to 60 days after You have

© Copyright 2018 WIBU-SYSTEMS AG

come back into compliance. Moreover, Your grants from a particular Contributor are reinstated on an ongoing basis if such Contributor notifies You of the non-compliance by some reasonable means, this is the first time You have received notice of non-compliance with this License from such Contributor, and You become compliant prior to 30 days after Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent infringement claim (excluding declaratory judgment actions, counter-claims, and cross-claims) alleging that a Contributor Version directly or indirectly infringes any patent, then the rights granted to You by any and all Contributors for the Covered Software under Section 2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or Your distributors under this License prior to termination shall survive termination.

```
************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
*                                                                      *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

© Copyright 2018 WIBU-SYSTEMS AG

```
10. Versions of the License
---------------------------

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section
10.3, no one other than the license steward has the right to modify or
publish new versions of this License. Each version will be given a
distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version
of the License under which You originally received the Covered Software,
or under the terms of any subsequent version published by the license
steward.

10.3. Modified Versions

If you create software not governed by this License, and you want to
create a new license for such software, you may create and use a
modified version of this License if you rename the license and remove
any references to the name of the license steward (except to note that
such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary
Licenses

If You choose to distribute Source Code Form that is Incompatible With
Secondary Licenses under the terms of this version of the License, the
notice described in Exhibit B of this License must be attached.

Exhibit A - Source Code Form License Notice
-------------------------------------------

  This Source Code Form is subject to the terms of the Mozilla Public
  License, v. 2.0. If a copy of the MPL was not distributed with this
  file, You can obtain one at https://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a LICENSE
file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
---------------------------------------------------------

  This Source Code Form is "Incompatible With Secondary Licenses", as
  defined by the Mozilla Public License, v. 2.0.
```

Licenses in Runtime and SDK indirect via Qt

The listed components are directly or indirectly required via using Qt. Only the 'lite' installer on Linux and the 'reduced' installer on Windows require no Qt.

Text Codecs: EUC-JP, ISO 2022-JP (JIS), Shift-JIS

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qeucjpcodec.html http://doc.qt.io/qt-5/qtcore-attribution-qjiscodec.html http://doc.qt.io/qt-5/qtcore-attribution-qsjiscodec.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 1999 Serika Kurusugawa, All rights reserved.


Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
```


© Copyright 2018 WIBU-SYSTEMS AG

```
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

Text Codec: GBK

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qbkcodec.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 2000 TurboLinux, Inc.  Written by Justin Yu and Sean Chen.
Copyright (C) 2001, 2002 Turbolinux, Inc.  Written by James Su.
Copyright (C) 2001, 2002 ThizLinux Laboratory Ltd.  Written by Anthony Fok.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

PCRE2

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qbkcodec.html

© Copyright 2018 WIBU-SYSTEMS AG

Operating System	Windows, macOS, Linux

```
PCRE2 LICENCE
-------------
PCRE2 is a library of functions to support regular expressions whose syntax
and semantics are as close as possible to those of the Perl 5 language.

Release 10 of PCRE2 is distributed under the terms of the "BSD" licence, as
specified below, with one exemption for certain binary redistributions. The
documentation for PCRE2, supplied in the "doc" directory, is distributed under
the same terms as the software itself. The data in the testdata directory is
not copyrighted and is in the public domain.

The basic library functions are written in C and are freestanding. Also
included in the distribution is a just-in-time compiler that can be used to
optimize pattern matching. This is an optional feature that can be omitted when
the library is built.

THE BASIC LIBRARY FUNCTIONS
---------------------------

Written by:       Philip Hazel
Email local part: ph10
Email domain:     cam.ac.uk

University of Cambridge Computing Service,
Cambridge, England.

Copyright (c) 1997-2017 University of Cambridge
All rights reserved.

PCRE2 JUST-IN-TIME COMPILATION SUPPORT
--------------------------------------

Written by:       Zoltan Herczeg
Email local part: hzmester
Emain domain:     freemail.hu

Copyright(c) 2010-2017 Zoltan Herczeg
All rights reserved.

STACK-LESS JUST-IN-TIME COMPILER
--------------------------------

Written by:       Zoltan Herczeg
Email local part: hzmester
Emain domain:     freemail.hu

Copyright(c) 2009-2017 Zoltan Herczeg
All rights reserved.

THE "BSD" LICENCE
-----------------

Redistribution and use in source and binary forms, with or without
```


© Copyright 2018 WIBU-SYSTEMS AG

```
modification, are permitted provided that the following conditions are met:

    * Redistributions of source code must retain the above copyright notices,
      this list of conditions and the following disclaimer.

    * Redistributions in binary form must reproduce the above copyright
      notices, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.

    * Neither the name of the University of Cambridge nor the names of any
      contributors may be used to endorse or promote products derived from this
      software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

EXEMPTION FOR BINARY LIBRARY-LIKE PACKAGES
------------------------------------------

The second condition in the BSD licence (covering binary redistributions) does
not apply all the way down a chain of software. If binary package A includes
PCRE2, it must respect the condition, but if package B is software that
includes package A, the condition is not imposed on package B unless it uses
PCRE2 independently.
```

Parts of QTemporaryFile

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qtemporaryfile.html)
Operating System	Windows, macOS, Linux

```
Copyright (c) 1987, 1993
     The Regents of the University of California.  All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors
   may be used to endorse or promote products derived from this software
   without specific prior written permission.
```

© Copyright 2018 WIBU-SYSTEMS AG

```
THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE.
```

QEventDispatcher on macOS

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qeventdispatcher-cf.html)
Operating System	macOS

```
Copyright (c) 2007-2008, Apple, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimer in the documentation
    and/or other materials provided with the distribution.

  * Neither the name of Apple, Inc. nor the names of its contributors
    may be used to endorse or promote products derived from this software
    without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Cocoa Platform Plugin

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-cocoa-platform-plugin.html

© Copyright 2018 WIBU-SYSTEMS AG

Operating System	macOS

Copyright (c) 2007-2008, Apple, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice,
    this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice,
    this list of conditions and the following disclaimer in the documentation
    and/or other materials provided with the distribution.

  * Neither the name of Apple, Inc. nor the names of its contributors
    may be used to endorse or promote products derived from this software
    without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

IAccessible2 IDL Specification

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-iaccessible2.html
Operating System	macOS

Copyright (c) 2013 Linux Foundation
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

 1. Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.

 2. Redistributions in binary form must reproduce the above
    copyright notice, this list of conditions and the following
    disclaimer in the documentation and/or other materials
    provided with the distribution.

```
  3. Neither the name of the Linux Foundation nor the names of its
     contributors may be used to endorse or promote products
     derived from this software without specific prior written
     permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This BSD License conforms to the Open Source Initiative "Simplified
BSD License" as published at:
http://www.opensource.org/licenses/bsd-license.php
```

qtmain Library

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtmain.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 2016 The Qt Company Ltd.

Commercial License Usage
Licensees holding valid commercial Qt licenses may use this file in
accordance with the commercial license agreement provided with the
Software or, alternatively, in accordance with the terms contained in
a written agreement between you and The Qt Company. For licensing terms
and conditions see https://www.qt.io/terms-conditions. For further
information use the contact form at https://www.qt.io/contact-us.

BSD License Usage
Alternatively, you may use this file under the terms of the BSD license
as follows:

"Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
  * Redistributions of source code must retain the above copyright
    notice, this list of conditions and the following disclaimer.
  * Redistributions in binary form must reproduce the above copyright
    notice, this list of conditions and the following disclaimer in
    the documentation and/or other materials provided with the
    distribution.
  * Neither the name of The Qt Company Ltd nor the names of its
    contributors may be used to endorse or promote products derived
```


© Copyright 2018 WIBU-SYSTEMS AG

```
    from this software without specific prior written permission.


THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."
```

Anti-aliasing rasterizer from FreeType 2

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype.html
Operating System	Windows, macOS, Linux

```
Portions of this software are copyright <2006>.
The FreeType Project (www.freetype.org).
All rights reserved.
```

FreeType 2

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype.html
Operating System	Windows, macOS, Linux

```
Portions of this software are copyright <2006>.
The FreeType Project (www.freetype.org).
All rights reserved.
```

LibJPEG-turbo

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-libjpeg.html
Operating System	Windows, macOS, Linux

```
This software is based in part on the work of the Independent JPEG Group.
libjpeg-turbo Licenses
======================
...
The Modified (3-clause) BSD License
===================================

Copyright (C)2009-2017 D.R.Commander.  All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:
```

© Copyright 2018 WIBU-SYSTEMS AG

```
- Redistributions of source code must retain the above copyright notice,
  this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice,
  this list of conditions and the following disclaimer in the documentation
  and/or other materials provided with the distribution.
- Neither the name of the libjpeg-turbo Project nor the names of its
  contributors may be used to endorse or promote products derived from this
  software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS",
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.
```

Contributions to the Cocoa Platform Plugin Files

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/mac-licensing.html
Operating System	macOS

```
Copyright (C) 2007-2008, Apple, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met:

•       Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.

•       Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials provided with
the distribution.

•       Neither the name of Apple, Inc. nor the names of its contributors may be used to endorse or
promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

Freetype 2 - Bitmap Distribution Format (BDF) support

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-bdf.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 2001-2002 by Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
```


© Copyright 2018 WIBU-SYSTEMS AG

```
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

*** Portions of the driver (that is, bdflib.c and bdf.h):

Copyright 2000 Computing Research Labs, New Mexico State University
Copyright 2001-2002, 2011 Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT
OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

Freetype 2 - Portable Compiled Format (PCF) support

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-bdf.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 2000 by Francesco Zappa Nardelli

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
```

© Copyright 2018 WIBU-SYSTEMS AG

```
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

HarfBuzz

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-harfbuzz.html
Operating System	Windows, macOS, Linux

```
HarfBuzz was previously licensed under different licenses.  This was
changed in January 2008.  If you need to relicense your old copies,
consult the announcement of the license change on the internet.
Other than that, each copy of HarfBuzz is licensed under the COPYING
file included with it.  The actual license follows:


Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the
above copyright notice and the following two paragraphs appear in
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE.  THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

HarfBuzz-NG

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-harfbuzz-ng.html
Operating System	Windows, macOS, Linux

```
HarfBuzz is licensed under the so-called "Old MIT" license.  Details follow.
For parts of HarfBuzz that are licensed under different licenses see individual
files names COPYING in subdirectories where applicable.
```


© Copyright 2018 WIBU-SYSTEMS AG

```
Copyright © 2010,2011,2012  Google, Inc.
Copyright © 2012  Mozilla Foundation
Copyright © 2011  Codethink Limited
Copyright © 2008,2010  Nokia Corporation and/or its subsidiary(-ies)
Copyright © 2009  Keith Stribley
Copyright © 2009  Martin Hosken and SIL International
Copyright © 2007  Chris Wilson
Copyright © 2006  Behdad Esfahbod
Copyright © 2005  David Turner
Copyright © 2004,2007,2008,2009,2010  Red Hat, Inc.
Copyright © 1998-2004  David Turner and Werner Lemberg

For full copyright notices consult the individual files in the package.


Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the
above copyright notice and the following two paragraphs appear in
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE.  THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

HarfBuzz-NG

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-harfbuzz.html
Operating System	Windows, macOS, Linux

```
HarfBuzz was previously licensed under different licenses.  This was
changed in January 2008.  If you need to relicense your old copies,
consult the announcement of the license change on the internet.
Other than that, each copy of HarfBuzz is licensed under the COPYING
file included with it.  The actual license follows:


Permission is hereby granted, without written agreement and without
license or royalty fees, to use, copy, modify, and distribute this
software and its documentation for any purpose, provided that the
above copyright notice and the following two paragraphs appear in
all copies of this software.

IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE TO ANY PARTY FOR
```

© Copyright 2018 WIBU-SYSTEMS AG

```
DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN
IF THE COPYRIGHT HOLDER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.

THE COPYRIGHT HOLDER SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING,
BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE.  THE SOFTWARE PROVIDED HEREUNDER IS
ON AN "AS IS" BASIS, AND THE COPYRIGHT HOLDER HAS NO OBLIGATION TO
PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.
```

OpenGL ES 2 Headers

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-opengl-es2-headers.html
Operating System	Windows, macOS, Linux

```
Copyright (c) 2013-2014 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and/or associated documentation files (the
"Materials"), to deal in the Materials without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Materials, and to
permit persons to whom the Materials are furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
```

OpenGL Headers

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-opengl-headers.html
Operating System	Windows, macOS, Linux

```
Copyright (c) 2013-2014 The Khronos Group Inc.

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and/or associated documentation files (the
"Materials"), to deal in the Materials without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Materials, and to
permit persons to whom the Materials are furnished to do so, subject to
the following conditions:
```


© Copyright 2018 WIBU-SYSTEMS AG

```
The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Materials.

THE MATERIALS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
MATERIALS OR THE USE OR OTHER DEALINGS IN THE MATERIALS.
```

XCB

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-xcb.html
Operating System	Windows, macOS, Linux

```
Copyright © 2000 Keith Packard
Copyright © 2006 Jamey Sharp
Copyright © 2007-2008 Vincent Torri <vtorri@univ-evry.fr>
Copyright © 2007 Bart Massey
Copyright © 2008-2009 Julien Danjou <julien@danjou.info>
Copyright © 2008 Arnaud Fontaine <arnau@debian.org>
Copyright © 2008 Bart Massey <bart@cs.pdx.edu>
Copyright © 2008 Ian Osgood <iano@quirkster.com>
Copyright © 2008 Jamey Sharp <jamey@minilop.net>
Copyright © 2008 Josh Triplett <josh@freedesktop.org>
Copyright © 2008 Ulrich Eckhardt doomster@knuut.de

Permission is hereby granted, free of charge, to any person
obtaining a copy of this software and associated documentation
files (the "Software"), to deal in the Software without
restriction, including without limitation the rights to use, copy,
modify, merge, publish, distribute, sublicense, and/or sell copies
of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF
CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors or
their institutions shall not be used in advertising or otherwise to
promote the sale, use or other dealings in this Software without
prior written authorization from the authors.
```

© Copyright 2018 WIBU-SYSTEMS AG

xkbcommon

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-xkbcommon.html
Operating System	Windows, macOS, Linux

```
The following is a list of all copyright notices and license statements which
appear in the xkbcommon source tree.

If making new contributions, the first form (i.e. Daniel Stone, Ran Benita,
etc) is vastly preferred.

All licenses are derivative of the MIT/X11 license, mostly identical other
than no-endorsement clauses (e.g. paragraph 4 of The Open Group's license).

These statements are split into two sections: one for the code compiled and
distributed as part of the libxkbcommon shared library and the code
component of all tests (i.e. everything under src/ and xkbcommon/, plus the
.c and .h files under test/), and another for the test data under test/data,
which is distributed with the xkbcommon source tarball, but not installed to
the system.


BEGINNING OF SOFTWARE COPYRIGHT/LICENSE STATEMENTS:


--------------------------------------------------------------------------------

Copyright © 2009-2012 Daniel Stone
Copyright © 2012 Ran Benita <ran234@gmail.com>
Copyright © 2010, 2012 Intel Corporation
Copyright © 2008, 2009 Dan Nicholson
Copyright © 2010 Francisco Jerez <currojerez@riseup.net>

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next
paragraph) shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```


© Copyright 2018 WIBU-SYSTEMS AG

Copyright 1985, 1987, 1988, 1990, 1998 The Open Group

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN
ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the names of the authors or their
institutions shall not be used in advertising or otherwise to promote the
sale, use or other dealings in this Software without prior written
authorization from the authors.

Copyright (c) 1993, 1994, 1995, 1996 by Silicon Graphics Computer Systems, Inc.

Permission to use, copy, modify, and distribute this
software and its documentation for any purpose and without
fee is hereby granted, provided that the above copyright
notice appear in all copies and that both that copyright
notice and this permission notice appear in supporting
documentation, and that the name of Silicon Graphics not be
used in advertising or publicity pertaining to distribution
of the software without specific prior written permission.
Silicon Graphics makes no representation about the suitability
of this software for any purpose. It is provided "as is"
without any express or implied warranty.

SILICON GRAPHICS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SILICON
GRAPHICS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL
DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE
OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH
THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Digital not be
used in advertising or publicity pertaining to distribution of the
software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL
DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS
SOFTWARE.

Copyright (C) 2011 Joseph Adams <joeyadams3.14159@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

END OF SOFTWARE COPYRIGHT/LICENSE STATEMENTS

© Copyright 2018 WIBU-SYSTEMS AG

```
BEGINNING OF LICENSE STATEMENTS FOR UNDISTRIBUTED DATA FILES IN test/data,
derived from xkeyboard-config:



-------------------------------------------------------------------------

Copyright 1996 by Joseph Moss
Copyright (C) 2002-2007 Free Software Foundation, Inc.
Copyright (C) Dmitry Golubev <lastguru@mail.ru>, 2003-2004
Copyright (C) 2004, Gregory Mokhin <mokhin@bog.msu.ru>
Copyright (C) 2006 Erdal Ronahî

Permission to use, copy, modify, distribute, and sell this software and its
documentation for any purpose is hereby granted without fee, provided that
the above copyright notice appear in all copies and that both that
copyright notice and this permission notice appear in supporting
documentation, and that the name of the copyright holder(s) not be used in
advertising or publicity pertaining to distribution of the software without
specific, written prior permission.  The copyright holder(s) makes no
representations about the suitability of this software for any purpose.  It
is provided "as is" without express or implied warranty.

THE COPYRIGHT HOLDER(S) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
EVENT SHALL THE COPYRIGHT HOLDER(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR
CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.
```

Freetype 2 - zlib

Project	Runtime component CodeMeter Control Center (CodeMeter License Server)
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-zlib.html
Operating System	Windows, macOS, Linux

```
Copyright (C) 1995-2002 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied
warranty.  In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
```

© Copyright 2018 WIBU-SYSTEMS AG

```
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu
```

musl

Project	CmEmbedded
Version	2.3
Operating system	Android

```
musl as a whole is licensed under the following standard MIT license:

Copyright © 2005-2014 Rich Felker, et al.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```


© Copyright 2018 WIBU-SYSTEMS AG

Index

- * -

*.WibuCmLIF (license infromation file) 17

- A -

Access Password
- Product Item Option 29

Activation Time
- Product Item Option 22

Allocation order of licenses 30

Android
- Requirements CmEmbedded 44

Apache Commons
- Open Source 168

ASM Java Program Library
- Open Source 167

ASN.1 Compiler
- Open Source 166

AxProtector Commandline
- -! (Create *.wbc file) 84, 137
- -# (Logging) 84, 137
- -? -h (Options and help) 84, 137
- -@cmds.wbc (Parameter in executable file) 84, 137
- -A[AES] (Encryption algorithm) 73, 126
- AutoCAD2011 - ARX files (pliug-in) 79, 132
- -CAA (Security options) 76, 129
- -CACT (CmContainer SystemTime) 76, 129
- -CAD (File Access Mode) 76, 77, 129, 130
- -CAE (Plug-Out) 77, 130
- -CAG (Anti-Debugging) 77, 130
- -CAL (areas of encryption) 77, 130
- -CAM (System Menu) 78, 131
- -CAR (Runtime Check) 78, 131
- -CAS (Percentage Encrypted) 78, 131
- -CAT (Certified and System Time) 78, 131
- -CAV (Code Integrity Check) 78, 131
- -CCA (Target platform and option -CCX) 78, 131
- -CCB (.reloc Section translation) 78, 79, 131, 132
- -CCC (ActiveX / OCX Images) 79, 132
- -CCD (Shared Objects options) 80, 133
- -CCE (PE is not enlarged) 79, 132
- -CCF (Extended protection / special handling) 79, 132
- -CCH (Preventing global hooking) 79, 132
- -CCI (Load sequence) 79, 132
- -CCK (DLL unload Windows XP) 79, 132
- -CCQ (Clear licenses on Exit Process) 79, 132
- -CCR (Deactivation renaming of Sections) 79, 132
- -CCS (Licenses from first connected CmContainer) 79, 132
- -CCT (Specification architecture AxEngine) 79, 132
- -CCX (Mixed Mode Assemblies) 79, 132
- -CDC (File name extension for file encrypting) 80, 133
- -CDH (Access to license file encryption) 80, 133
- -CDK (Licensing system file encryption) 80, 133
- -CFx (Feature Code) 74, 127
- -CIH (WUPI and Hooking) 80, 133
- -CIN (No error messages IxProtector) 80, 133
- -CIP (IxProtector) 80, 133
- -CP (Cleanup mechanism Windows) 80, 133
- -D (Driver version) 74, 127
- -EA (Activation Time check) 81, 134
- -EE (Expiration Time check) 81, 134
- -EF (Decrement Firm Access Counter) 81, 134
- -EM (Maintenance Period required) 81, 134
- -ET (Enforce Certified Time update) 81, 134
- -EU (Unit Counter check) 81, 134
- -FW (Firmware Version) 74, 127
- -FW (minim. Firmware on encryption) 82, 135
- -Fx (Firm Code) 74, 127
- -G (Exclude areas fropm encryption) 81, 82, 134, 135
- -I (Exception handling Plugin DLLs) 82, 135
- -K (Licensing System) 74, 127
- -L (Language message texts) 82, 135
- -M (Outout texts for message texts) 82, 135
- -N (Network access mode) 75, 128
- -NNI Ignore Linger Time 75, 128
- -O (Name and path of encrypted target file) 84, 137
- -Px Product Code) 74, 127
- -RD (Release Date) 74, 127
- -RID (Numberl RID variants) 81, 134
- -RIDI (Number of RID and Trap variants IxProtector) 81, 134
- -S (Search order for licenses) 74, 75, 127, 128
- -SIG (Signature Private Key Certificate) 75, 76, 128, 129
- -SW (Search order for licenses, WAN) 74, 75, 127, 128
- -U (Calling message DLL) 83, 136
- -UDT (user defined text) 84, 137
- -UI (Inline message assembly, .NET) 83, 84, 136, 137
- -UM (Call message assembly, .NET) 83, 136
- -V (Verbose mode) 84, 137
- -WC (Threshold Certified Time) 82, 135
- -WE (Threshold Expiration Time) 82, 135
- -WP (Threshold Usage Period) 82, 135
- -WU (Threshold Unit Counter) 82, 135
- -X (Static linking) 73, 126
- -XC AxProtector Linux using CmEmbedded (CmDongle) 73, 126
- -XCA AxProtector Linux using CmEmbedded (CmActLicense) 73, 126

- B -

Binding Extension 16

Binding scheme 16

Bootstrapper dotnetInstaller
- Open Source 166

- C -

Certificate signing
- AxProtector Commandline 75, 76, 128, 129

CmActLicense
- Activating licenses 17
- additional activation options 16
- Binding Extension 16
- Binding scheme 16
- None-Binding 16
- Protection-Only license model 16
- Smart Bind 102
- SmartBind 16
- SmartBind behaviour in VM 102
- Trial License license model 16

CmActLicense License
- Programming (CmBoxPgm) 103

CmBoxPgm 52, 100
- <public key file> 67, 120
- Access Password -papwd 57, 110
- Activation by Phone CmActLicense 102
- Activation Time -pat 57, 110
- Activation Time, absolute -pata 57, 110
- Activation Time, relative -patr 57, 110
- Add / Update -cau 52, 100
- Add -ca 52, 100
- Allowed CmAct Ids -lacids 101
- Backup File -bkp 66, 119
- Box Index -qb 53
- Box Index Range -qnx 53
- Box Passwort -pwd 53
- Cleanup Registry -rcl 66, 119
- CmActLicense Activation Code -lac 101
- CmActLicense Binding Value-lbind 101
- CmActLicense in VM -lopt:vm 103
- CmActLicense License ID -lpid 103
- CmActLicense License Information File (phone) -lip 102
- CmActLicense License Information File -lif 102
- CmActLicense License Options -lopt 103
- CmActLicense reimport -lopt:reimport 103
- CmActLicense Target Operating System -los 103
- Create RaC File -crac 66, 119
- Customer Owned License Information -pcoli 58, 110
- Delete -cd 52, 100
- Delete if possible -cdx 53, 101

CmBoxPgm 52, 100
Display CmActLicense Binding Scheme -lfs 102
Display CmActLicense Installation ID -ldi 102
Display CmActLicense License File -ldf 101
Expiration Time -pet 58, 111
Expiration Time, absolute -peta 58, 111
Expiration Time, relative -petr 58, 111
Extended Protected Data -ped 58, 111
Feature Map -pfm 59, 111
File-based Activation CmActLicense 102
Firm Access Counter -fac 55, 108
Firm Code -f 55, 108
Firm Item License Transfer Permissions 55, 108
Firm Item Text -ft 55, 108
Firm Key -fk 56, 109
Firm Precise Time, absolute -fpta 55, 108
Firm Precise Time, relative -fptr 55, 108
Firm Update Counter -fuc 56, 108
FSB Entry -fsb 56, 109
Help -? 66, 119
Hiidden Data -phd 59, 112
License Quantity -plq 59, 112
License Quantity, absolute -plqa 60, 112
License Quantity, relative -plqr 60, 113
License Transfer 61, 113
Licensé Transfer Depth -pltdepth 61, 113
License Transfer LTK targets -pltltktarg 61, 114
License Transfer Maximum Borrow Period -pltmbp 61, 114
License Transfer Permission -pltperm 61, 114
License Transfer Targets-plttarg 61, 114
License Transfer Type -plttype 61, 114
Linger Time -plt 60, 113
List -l 53, 101
Logging -log 66, 119
Maintenance Period -pmp 62, 115
Maintenance Period, Date -pmpd 62, 115
Maintenance Period, Integer-pmpi 62, 115
Maximum Encryption Rate -pmer 61, 114
Merge Remote Activation Update File -mrau 54
Minimum required runtime (CmActLicense) 103
Minimum Runtime Version -pmrt 62, 115
Module Item -pmi 62, 114
Named User -pnu 63, 115
Network License Counter -pnwc 63, 116
Product Code -p 57, 109
Protected Data -ppd 63, 116
Recursive Removal -r 53
Remote Activation Update -rau 54
Secret Data -psd 63, 116
Secure License Tracking 67, 120
Serial number -qs 53
Smart Bind 102
Text -pt 63, 116
Triple Mode Redundacy 59, 60, 112, 113
Unit Counter -puc 64, 116
Unit Counter, absolute -puca 64, 117
Unit Counter, relative -pucr 64, 117
Universal Data AES key storage -puvdaes 65, 118
Universal Data -puvd 65, 118
Universal Data RSA key storage -puvdrsa 65, 118
Universal Firm Code License Programming Options -lpo 103
Universal Firm Code License Update File -laf 101
Update -cu 52, 100
Usage Period -pup 64, 117
Usage Period, absolute -pupa 65, 117
Usage Period. relative -pupr 65, 118
User Data -pud 64, 117
Validation Mode -val 67, 120
Validation of Signed Log Files -vslf 67, 120
Verbose Mode -v 67, 119
WUPI Data -pwupidata 65, 118
CmEmbedded
Delivery packages 47
Requirements: Android 44
Requirements: Linux 44
Requirements: macOS 47
Requirements: QNX 45
Requirements: VxWorks 45
Requirements: Windows CE 47
version information 49
CmFirm.wbc 7
CodeMeter
Concept 17
Form factors 15
CodeMeter License Central 67, 120
CodeMeter Start Center 39
Customer Owned License Information (COLI)
Product Item Option 27
- D -
Driver version (minim.)
AxProtector Commandline 74, 127
- E -
Encryption 35
Asymmetric 38
Direct 37
Indirect 37
Key derivation 35
Symmetric 38
Encryption Code Options 35
Expiration Time
Product Item Option 22
Extended Protected Data
Product Item Option 28
- F -
Feature Code 24
AxProtector Commandline 74, 127
Feature Map
Product Item Option 24
Version Management 24
FIO (Firm Item Options) 17
Firm Code 17
AxProtector Commandline 74, 127
Firm Item 17
Firm Item Options (FIO) 17
Firm Item Text
define 52, 100
Firm Key 17
Firm Security Box (FSB) 19
Firmware Version
AxProtector Commandline 74, 127
Flot
Open Source 154
Form Factors
CodeMeter 15
Human Interface Device (HID) 15
FSB (Firm Security Box) 19
- G -
golang.org (/x/sys / and /x/crypto)
Open Source 158
go-macaron
Open Source 155
Google Protocol Buffers
Open Source 165
- H -
HID (Human Interface Device) 15
Hidden Data
Product Item Option 28
Human Interface Device (HID) 15
- I -
Implicit Firm Item 17

- J -
JQuery
libusb Linux 158
Open Source 154
- K -
Key Derivation
Black Key 35
Feature Map 35
Firm Code 35
Product Code 35
Release Date 35
- L -
libcurl
Open Source 164
License information file (*.WibuCmLIF) 17
License Model
Concurrent License 32
Demo Version 32
Downgrade Management 34
Floating License 32
Hot / Cold Standby 34
License Borrowing 35
Local Single User Licenses 32
Machine-bound Licenses 35
Modular Licenses 33
Named User Licenses 35
Network License 32
Overflow Licenses 34
Pay-per ... 31, 33
Renting, Leasing 33
Standard 31
Version Management 34
License Quantity
Product Item Option 21
License Transfer
CmBoxPgm 61, 113
Linger Time
Product Item Option 25
Linux
Requirements CmEmbedded 44
LLVM compiler and toolchain technology
Open Source 166
Locale license
Remote Desktop 21
Server operating systems 21
- M -
macOS
Requirements CmEmbedded 47
Maintenance Period
Product Item Option 25
Maximum Encryption Rate
Product Item Option 29
Minimum Runtime Version
Product Item Option 26
MIT
Open Source 171
Module Item
concept 30
musl
Open Source 194
- N -
Named User
Product Item Option 26
Network access mode
AxProtector Commandline 75, 128
None-Binding 16
- O -
On-demand Decryption 35
Open Source
Anti-aliasing rasterizer from FreeType 2 183
Apache Commons 168
ASM Java Program Library 167
ASN.1 Compiler 166
Bootstrapper dotnetInstaller 166
Cocoa Platform Plugin 180
Contributions to the Cocoa Platform Plugin Files 184
Flot 154
FreeType 2 183
Freetype 2 - Bitmap Distribution Format (BDF) support 184
Freetype 2 - Portable Compiled Format (PCF) support 185
Freetype 2 - zlib 193
golang.org (/x/sys / and /x/crypto) 158
go-macaron 155
Google Protocol Buffers 165
HarfBuzz 186
HarfBuzz NG 186, 187
IAccessible2 IDL Specification 181
JQuery 154
libcurl 164
LibJPEG-turbo 183
libusb Linux 158
LLVM compiler and toolchain technology 166
MIT 171
musl 194
OpenGL ES 2 Headers 188
OpenGL Headers 188
Parts of QTemporaryFile 179
PCRE2 177
Public Suffix List 171
QEventDispatcher on macOS 180
qtmain Library 182
Text Codec: GBK 177
Text Codecs: EUC-JP, ISO 2022-JP (JIS), Shift-JIS 176
XCB 189
xkbcommon 190
- P -
PIO (Product Item Options) 19
Product Code 17
AxProtector Commandline 74, 127
Product Item Option 20
Product Item 17
Product Item Option
Access Password 29
Activation Time 22
Customer Owned License Information (COLI) 27
Expiration Time 22
Extended Protected Data 28
Feature Map 24
Hidden Data 28
License Quantity 21
Linger Time 25
Maintenance Period 25
Maximum Encryption Rate 29
Minimum Runtime Version 26
Named User 26
Product Code 20
Protected Data 27
Secret Data 28
Text 21
Unit Counter 23
Usage Period 23
User Data 27
Product Item Options (PIO) 17, 19
Programming (CmBoxPgm)
CmActLicense License 103
Programming of CmContainer
*.wbb 69, 122
*.WibuCmRaC 69, 122
*.WibuCmRaM 69, 122
*.WibuCmRaU 69, 122
CmBoxPgm 52, 100

Programming of CmContainer
CodeMeter License Central 67, 120
LIF, License Information File 69, 122
Programming via file transfer 69, 122
Programming Samples
Samples 8
Protected Data
Product Item Option 27
Protection Only license
CmActLicense binding 16
Programming example 106
Public Suffix List
Open Source 171
- Q -
QNX
Requirements CmEmbedded 45
- R -
Release Date 25
AxProtector Commandline 74, 127
- S -
Samples
Programming 8
Search order for licenses
AxProtector Commandline 74, 75, 127, 128
Search order for licenses (WAN)
AxProtector Commandline 74, 75, 127, 128
Secret Data
Product Item Option 28
Server operating systems
local license 21
Smart Bind
CmActLicense 102
CmBoxPgm 102
SmartBind 16
SPI integration 149
SPI Protocol Specification 151
Support
Wibu-Systems 9
- T -
Text
Product Item Option 21
Trial license
CmActLicense binding 16
Programming example 105
Triple Mode Redundancy 59, 60, 112, 113
- U -
Unit Counter
Product Item Option 23
Universal Firm Code
evaluation license 7
Usage Period 23
Product Item Option 23
User Data
Product Item Option 27
- V -
Version information
CmEmbedded 49
VxWorks
Requirements CmEmbedded 45
- W -
wbc file 7
Windows CE
Requirements CmEmbedded 47