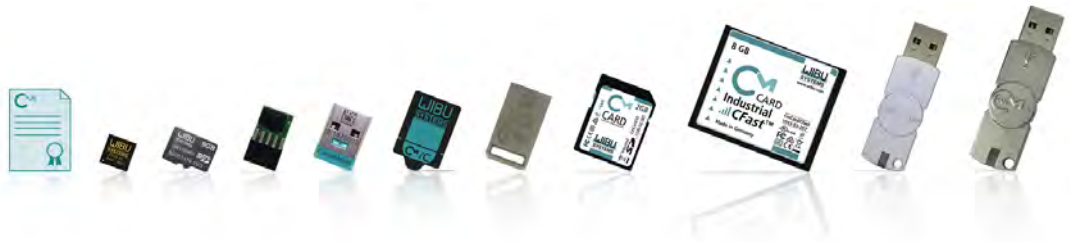


CodeMeter Developer Guide 7.0 - December 2019

Version 7.0 - December 2019

Wibu-Systems is member of:

PCMCIA since1993

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 Version 11

II About this Guide 12

1 Safety Instructions 13

2 Installation 14

3 Shipped CmDongles 14

4 Additional Help Documentation 14

5 Typographical Conventions 15

6 Support by Wibu-Systems 16

7 About Wibu-Systems 16

III Software Protection and License Management 18

1 CmContainer Types 20

1.1 CmDongle: CodeMeter Form Factors 20

1.2 CmActLicense: Binding and Activation 21

1.2.1 CmActLicense Binding 21

1.2.2 CmActLicense Activation 22

1.3 CmCloudContainer: credential based licensing 23

2 Operating Systems supported by CodeMeter 24

3 Additional Features 24

4 CodeMeter as Token 25

5 CodeMeter on Embedded Systems 25

IV The CodeMeter Concept 26

1 Product Item Options - Custom-made License Entries 27

1.1 Product Code 28

1.2 Text 29

1.3 License Quantity 29

1.4 Activation Time 30

1.5 Expiration Time 30

1.6 Usage Period 31

1.7 Unit Counter 31

1.8 Feature Map 31

1.9 Maintenance Period 32

1.10 Linger Time 33

1.11 Minimum Runtime Version 33

1.12 Named User 34

1.13 Customer Owned License Information (COLI) 34

1.14 User Data 34

1.15 Protected Data 35

1.16 Extended Protected Data 35

1.17 Hidden Data 35

1.18 Secret Data 36

1.19 Access Password 36

1.20 Maximum Encryption Rate 36

1.21 Universal Data 37

2 Allocation order of licenses 39

3 Module Item 39

4 Security with Capital S 39

5 License Models - Mapping Variety using CodeMeter 40

5.1 Implementing License Models 41

5.1.1 Local Single User Licenses 41

5.1.2 Concurrent-/ Floating License in the Network 41

5.1.3 Demo Versions 41
5.1.4 Modular Licenses 41
5.1.5 Leasing 42
5.1.6 Pay-per-use Licenses 42
5.1.7 Downgrade/Version Managment 42
5.1.8 Overflow 42
5.1.9 Hot / Cold Standby 42
5.1.10 Named User Licenses 43
5.1.11 Machine-bound Licenses 43
5.1.12 License Borrowing 43
6 License Transfer 43
7 Security by Encryption 44
7.1 Key Derivation - One License Entry - Many Keys *44*
8 Cryptography 46
8.1 Direct and Indirect Encryption *46*
8.2 Symmetric Enryption *46*
8.2.1 AES - Cipher Block Chaining Mode (CBC) (recommended)....46
8.3 Asymmetric Encryption *47*
8.3.1 ECC - Elliptic Curve Cryptography 47
8.3.2 ECIES - Elliptic Curve Integrated Encryption Scheme 47
8.3.3 ECDSA - Elliptic Curve Digital Signature Algorithm 47
8.3.4 RSA 47
8.4 Additional Encryption Algorithms *47*
V CodeMeter Start Center 48
1 Structure and Navigation 48
1.1 Menu Bar *48*
VI CodeMeter License Server 50
VII Automatic Software Protection using AxProtector (Tool of CodeMeter Protection Suite) 53
1 Structure and Navigation 54
1.1 Menu Bar *54*
1.2 Navigation Window *56*
1.3 Input Window *56*
1.4 Note and Error Window *56*
1.5 Project type area *56*
2 Project Dialog 56
3 Project Types 56
4 AxProtector Tab 57
4.1 Windows Application or DLL *57*
4.1.1 File to protect 58
4.1.2 Licensing Systems 58
4.1.2.1 *Licensing system - Add licenses* 61
4.1.3 Runtime Settings 64
4.1.3.1 *Advanced Runtime Settings* 65
4.1.4 Security Options 67
4.1.4.1 *Advanced Security Options* 69
4.1.5 Error Messages 71
4.1.6 Advanced Options 72
4.1.6.1 *License Lists* 72
4.1.6.2 *IxProtector* 77
4.1.6.3 *File Enryption* 79
4.1.7 Summary 81
4.2 .NET Assembly *82*
4.2.1 File to protect 84
4.2.2 Licensing Systems 84
4.2.2.1 *Licensing Systems - Add licenses* 87
4.2.3 Runtime Settings 90

4.2.3.1 *Advanced Runtime Settings* 91
4.2.4 Security Options 93
4.2.5 Error Messages 95
4.2.6 .NET Options 96
4.2.7 Advanced Options 97
4.2.7.1 *License Lists* 97
4.2.7.2 *IxProtector* 102
4.2.8 Summary 103
4.3 .NET Standard 2.0 Assembly *105*
4.3.1 File to protect 107
4.3.2 Licensing Systems 107
4.3.2.1 *Licensing Systems - Add licenses* 110
4.3.3 Runtime Settings 113
4.3.3.1 *Advanced Runtime Settings* 114
4.3.4 Security Options 116
4.3.5 Error Messages 118
4.3.6 .NET Options 119
4.3.7 Advanced Options 120
4.3.7.1 *License Lists* 120
4.3.7.2 *IxProtector* 125
4.3.8 Summary 126
4.4 macOS Application or Dylib *128*
4.4.1 File to protect 128
4.4.2 Licensing Systems 129
4.4.2.1 *Licensing Systems - Add licenses* 131
4.4.3 Runtime Settings 134
4.4.3.1 *Advanced Runtime Settings* 135
4.4.4 Error Messages 137
4.4.5 Security Options 137
4.4.5.1 *Advanced Security Options* 139
4.4.6 Advanced Options 141
4.4.6.1 *License Lists* 141
4.4.6.2 *IxProtector* 146
4.4.7 Summary 148
4.5 Java Application (jar file) *150*
4.5.1 File to protect 151
4.5.2 Licensing Systems 151
4.5.2.1 *Licensing Systems - Add licenses* 154
4.5.3 Runtime Settings 157
4.5.3.1 *Advanced Runtime Settings* 158
4.5.4 Security Options 160
4.5.5 Error Messages 162
4.5.6 Java Options 163
4.5.7 Advanced Options 164
4.5.7.1 *License Lists* 165
4.5.7.2 *IxProtector* 170
4.5.8 Summary 172
4.6 Linux Application or Shared Object *174*
4.6.1 File to protect 175
4.6.2 Licensing Systems 175
4.6.2.1 *Licensing Systems - Add licenses* 178
4.6.3 Runtime Settings 181
4.6.3.1 *Advanced Runtime Settings* 182
4.6.4 Security Options 184
4.6.4.1 *Advanced Security Options* 186
4.6.5 Error Messages 187
4.6.6 Advanced Options 188
4.6.6.1 *License Lists* 188
4.6.6.2 *IxProtector* 192
4.6.7 Summary 194

5 IxProtector Tab 196

5.1 Windows Application or DLL *196*

5.1.1 File to protect 197

5.1.2 Error Messages 198

5.1.3 Advanced Options 199

5.1.3.1 *License Lists* 199

5.1.3.2 *IxProtector* 203

5.1.4 Summary 205

5.2 .NET Assembly *206*

5.2.1 File to protect 207

5.2.2 Error Messages 208

5.2.3 .NET Options 209

5.2.4 Advanced Options 210

5.2.4.1 *License Lists* 210

5.2.4.2 *IxProtector* 214

5.2.5 Summary 215

5.3 .NET Standard 2.0 Assembly *217*

5.3.1 File to protect 217

5.3.2 Error Messages 218

5.3.3 .NET Options 219

5.3.4 Advanced Options 220

5.3.4.1 *License Lists* 220

5.3.4.2 *IxProtector* 224

5.3.5 Summary 225

5.4 macOS Application or Dylib *227*

5.4.1 File to protect 227

5.4.2 Error Messages 228

5.4.3 Advanced Options 229

5.4.3.1 *License Lists* 229

5.4.3.2 *IxProtector* 233

5.4.4 Summary 235

5.5 Linux Application or Shared Object *237*

5.5.1 File to protect 237

5.5.2 Error Messages 238

5.5.3 Advanced Options 239

5.5.3.1 *License Lists* 240

5.5.3.2 *IxProtector* 243

5.5.4 Summary 245

6 Other Tab 247

6.1 File Encryption *247*

6.1.1 File to protect 247

6.1.2 Licensing Systems 248

6.1.2.1 *Licensing Systems - Add licenses* 251

6.1.3 Advanced Options 254

6.1.3.1 *License Lists* 254

6.1.3.2 *File Encryption* 259

6.1.4 Summary 261

7 Commandline Options for AxProtector 263

7.1 Basic Options *263*

7.2 Options for the Licensing System *263*

7.3 Options for Encrypting and Decrypting *266*

7.4 Runtime Options *277*

7.5 Java-specific Settings *280*

7.6 Operational Options *285*

8 Advanced AxProtector Options 286

8.1 Translocated Execution *286*

8.2 License allocation in license lists *287*

8.3 IP Protection - protecting know how *287*

VIII Individual Software Protection 289

1 Handles 289

2 IxProtector (Tool of CodeMeter Protection Suite) and Software Protection API (WUPI) 290

3 WUPI Functions 290

3.1 WUPI: example of index-based placeholders *293*

3.1.1 Definition of Modules 294

3.1.2 Placeholders in IxProtector License and Functions Lists 294

3.1.3 Programming the CmContainer 296

3.1.4 Integration into the Source Code 297

3.1.5 Encryption using AxProtector 297

4 The CodeMeter Core API 297

4.1 Functional Areas *298*

4.1.1 Access API 298

4.1.2 Authentication API 298

4.1.3 Encryption API 298

4.1.4 Error Management API 299

4.1.5 Management API 299

4.1.6 Programming API 299

4.1.7 Remote Update API 299

4.1.8 Time Management API 300

4.1.9 License Transfer API 300

4.2 CodeMeter API Guide *300*

4.2.1 Structure and Navigation 300

4.2.2 Menu Bar 301

4.2.3 Tabs 301

4.2.4 Tree View 302

4.2.5 Handle Display Window 302

4.2.6 Interactive Area 302

4.2.7 Source Code Area 302

4.2.8 Record Area 302

4.3 Sample Applications: CmDemo, CmCalculator, WupiCalculator *303*

4.3.1 CmDemo 303

4.3.2 CmCalculator 304

4.3.3 WupiCalculator 304

IX Programming of CmContainer and Licensing Management 305

1 CodeMeter License Editor 306

1.1 Structure and Navigation *307*

1.1.1 Menu Bar 307

1.1.2 Symbol Bar 308

1.1.3 Tree View 308

1.1.4 Display Window 309

1.1.5 Output Window 309

1.2 Working with CodeMeter License Editor *309*

1.2.1 Starting CodeMeter License Editor 309

1.2.2 Display of connected CmDongles 309

1.2.2.1 *Refreshing Display* 309

1.2.2.2 *Open Context Files* 309

1.2.3 Creating, Editing and Deleting a Firm Code 310

1.2.4 Creating, Editing and Deleting a Product Code 311

1.2.5 Creating, Editing and Deleting a License Option 311

2 CmBoxPgm 315

2.1 Commandline Syntax *315*

2.2 Using CmBoxPgm *316*

2.3 Basic Commands *316*

2.4 CmContainer Options *317*

2.5 Firm Item Options *318*

2.6 Product Item Options *319*

2.7 CmActLicense Options ... 328
2.8 CmCloud options ... 335
2.9 FSB Entry Options ... 335
2.10 Special Commands ... 335
3 CodeMeter License Central ... 337
3.1 The Principle ... 337
3.2 The Architecture ... 338
3.3 Functions ... 339
3.3.1 Sales Interface ... 339
3.3.1.1 *Connectors* ... 339
3.3.1.2 *Gateway* ... 340
3.3.2 Depot Interface ... 341
3.3.3 Admin Interface ... 341
3.4 Application Scenarios CodeMeter License Central ... 341
4 Programming by File Transfer ... 342
X Deployment 345
1 Installation packages for Non-Windows Operating Systems ... 345
2 Deployment on Windows Operating Systems ... 346
2.1 Pre-configured Installation Packages ... 346
2.2 Customizing Options for Installation Packages ... 347
3 Mobile Installation on CmDongle (Windows) ... 350
4 CodeMeter Copy Installation on Windows ... 351
XI Advanced CodeMeter Features 353
1 Implicit Firm Item (IFI) ... 353
2 Enabling ... 353
2.1 Enabling Blocks as On/Off switches ... 354
2.2 Access Type - Simple or Time PIN ... 354
2.3 Enabling Mode ... 354
2.4 Deleting and Editing Enabling Blocks ... 355
3 Mapping (Lookup) of Enabling Blocks ... 355
3.1 Privileges - Enabling Level ... 355
3.2 Required Flag ... 355
4 Using Own Keys ... 356
5 Time Server: System Times and Certified Time ... 357
6 Locking a CmContainer ... 359
7 Backup of CmDongle Content ... 360
8 CodeMeter in a Wide Area Network (WAN) ... 361
8.1 WAN Infrastructure ... 361
8.2 CodeMeter-sided Implementation ... 362
8.2.1 Programming of licenses (CmBoxPgm) ... 362
8.2.2 License usage via API (CodeMeter API Guide) ... 363
8.2.3 license usage via API Automatic Encryption (CodeMeter Encryption Suite) ... 364
8.2.4 Configuring CmWAN network communication ... 365
8.2.4.1 *CodeMeter WebAdmin Configuration* ... 365
8.2.4.2 *Profiling in Registry or in server.ini File* ... 365
9 The use of write filters and CmActLicense ... 366
XII Manual 368
1 First important Information ... 368
2 Installation ... 370
2.1 Installation on 32/64-bit Windows ... 370
2.1.1 Installed files on 32/64-bit Windows ... 371
2.1.2 Uninstalling on 32/64-bit Windows ... 372

2.2 Installation on macOS ... *372*
2.2.1 Installed files on macOS ... 372
2.2.2 Uninstalling on macOS ... 373
2.3 Installation on Linux ... *373*
2.3.1 Uninstalling on Linux ... 374
3 Profiling - CodeMeter License Server settings ... 375
3.1 General ... *375*
3.2 AccessControl ... *381*
3.3 Backup ... *381*
3.4 HTTP ... *382*
3.5 HTTPS ... *383*
3.6 ServerSearchList ... *383*
3.7 TripleModeRedundancy ... *384*
4 CodeMeter Control Center ... 384
4.1 Structure and Navigation ... *386*
4.2 Menu Bar ... *387*
4.3 License Tab ... *389*
4.4 Events Tab ... *391*
4.5 Borrowing Tab ... *392*
4.6 Status and Starting CodeMeter WebAdmin ... *393*
5 Importing and Updating Licenses ... 393
5.1 The CmFAS Assistant in CodeMeter Control Center ... *394*
5.1.1 Create License Request File ... 394
5.1.1.1 *Extend Existing License* ... 395
5.1.1.2 *Add a License of a new Producer* ... 396
5.1.2 Import License Update ... 397
5.1.3 Create Receipt ... 398
6 CodeMeter WebAdmin ... 399
6.1 Basics ... *400*
6.2 Open CodeMeter WebAdmin ... *401*
6.3 Operating ... *402*
6.4 Dashboard ... *403*
6.5 Container ... *404*
6.5.1 Licenses ... 404
6.5.2 Firm Item Details ... 405
6.5.3 Product Item Details ... 405
6.5.4 CmContainer Info ... 407
6.5.5 User data ... 408
6.5.6 User Data Details ... 409
6.5.7 Backup and Restore ... 409
6.6 License Monitoring ... *411*
6.6.1 License Monitoring Details ... 412
6.6.2 Sessions ... 413
6.6.3 License Tracking ... 413
6.7 Diagnosis ... *416*
6.7.1 Events ... 416
6.8 Configuration ... *417*
6.8.1 Server Search List ... 417
6.8.2 Proxy ... 419
6.8.3 WebAmin ... 420
6.8.4 Backup ... 425
6.8.5 Server Access ... 425
6.8.6 License Access Permissions ... 426
6.8.7 Prepared License Borrowing ... 433
6.8.8 Time Server ... 434
6.8.9 Extra ... 435
6.9 Info ... *437*
6.10 License Transfer ... *437*

6.10.1 Licenses 439
6.10.1.1 *Move 'n' from 'n' licenses* 439
6.10.1.2 *Return 'n' from 'm' licenses* 441
6.10.2 License Borrowing 443
6.10.2.1 *Borrow* 443
6.10.2.2 *Return* 446
6.11 Module Items *448*
7 CmDust **450**
8 CMU - CodeMeter Universal Support Tool **452**
9 CodeMeter License Tracking **456**
9.1 Requirements and Configuration *457*
9.2 Logfile Format *458*
9.2.1 Definitions and Value Ranges 458
9.3 Entry Types *459*
9.3.1 List of Licenses Entry 459
9.3.2 License Entry 459
9.3.3 Access Entry 459
9.3.4 Release Entry 460
9.3.5 Borrow Access Entry 460
9.3.6 Borrow Return Entry 460
9.3.7 Denial Entry 460
9.3.8 Administrative Entry 460
9.3.9 SignedLogfile Entry 460
9.3.10 Signature Entry 461
10 HID Support **461**
10.1 Set from Mass Storage to HID *461*
10.2 Set from HID to Mass Storage *463*
10.3 Linux Kernel Settings *464*
XIII Glossary **466**
XIV Copyright information of software licenses used **468**
Index **505**

1 Version

CodeMeter Developer Guide Version 7.0, 12/13/2019.

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

2 About this Guide

CodeMeter® is the technology of Wibu-Systems providing secure protection and effective license management of software and digital content.

> 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, at times, the terms "licensor" and "licensee" are used. The term "licensor" may be replaced by "developer" or "vendor", while "licensee" refers to the software "end user" or a user of digital content.

The *CodeMeter* Developer Guide is divided into separate parts.

The 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.

Part II 18 sketches the outstanding features of *CodeMeter* in the areas of security, hardware and software-based software protection, and flexible license management. Part III 26 follows describing how the concept of *CodeMeter* meets protection, licensing, and security requirements. Moreover, basic terms are introduced.

Part IV 48 describes *CodeMeter Start Center*, the communication turntable to open single *CodeMeter* tools, while Part V 50 turns the attention to *CodeMeter License Server* as the central component of *CodeMeter* designed to run as a service on each computer, where *CodeMeter* protected digital content is used.

Part VI 53 and Part VII 289 point to the automatic and individual integration of the protection into your software. On the one hand, *AxProtector* for integrating automatic software protection using the graphical user interface (GUI) or the commandline for different project types. On the other hand, *IxProtector* for integrating individual software protection with the *Software Protection API* (WUPI) and the basic *CodeMeter Core API*.

Part VIII 305 comprises the applications you use to create, manage, and deliver *CodeMeter* licenses of protected digital products: *CodeMeter License Editor*, *CmBoxPgm*, and *Code Meter License Central*. Part IX 345 follows with a description of deployment options: what does your customer need for running the protected software?

Part X 353 informs you on advanced *CodeMeter* features, such as, ***Implicit Firm Item***, ***Enabling***, using own keys, and the backup / restore of *CmContainer* contents.

Finally, Part XI 368 is designed as an Administrator Guide holding *CodeMeter* installation information for different operating systems, the tools *CodeMeter WebAdmin*, *CodeMeter Control Center*, *CmDust* and *cmu* which support the administrator in the daily use of *CodeMeter*.

The Guide closes with a glossary and an index.

Generally, the Guide is structured along the lines as shown in the figure below.

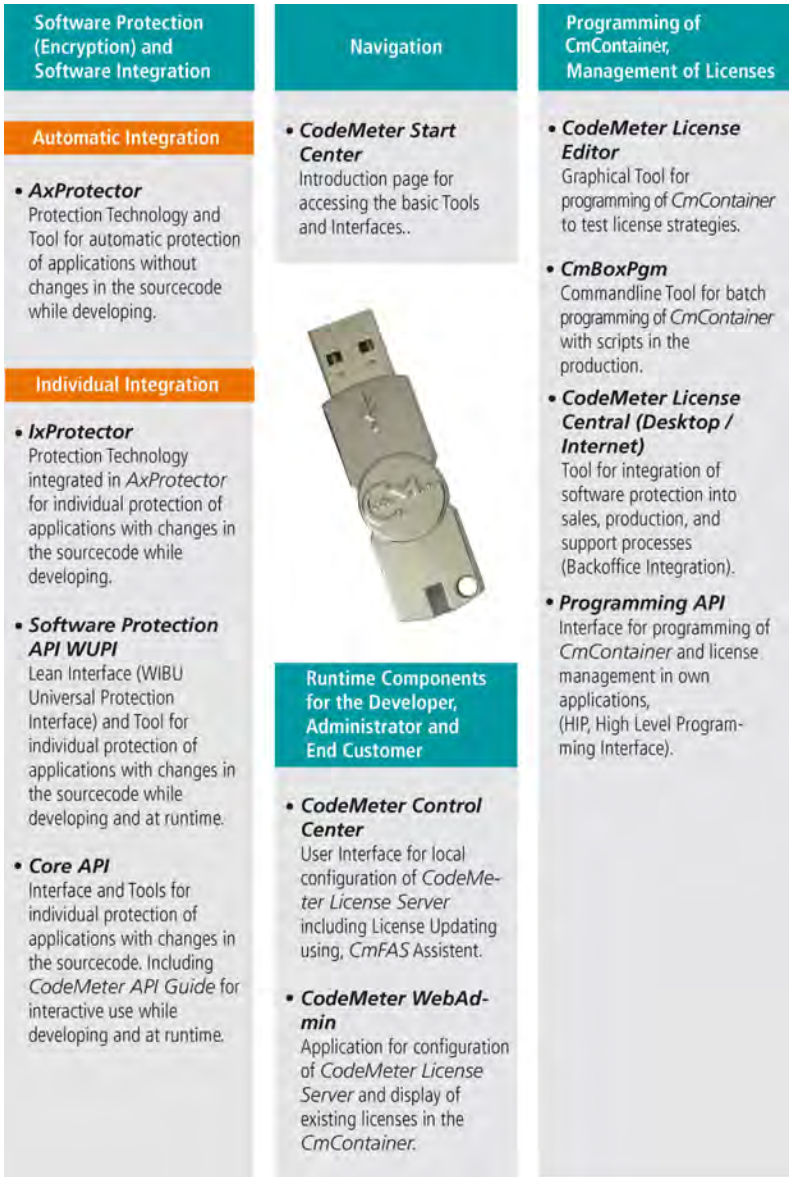


Figure 2: Documentation Structure

2.1 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 (https://www.wibu.com/support/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.

2.2 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.

2.3 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*.

> An entry with an ***Universal Firm Code*** evaluation license `6000010` for *CmDongles* and *CmActLicense* are pre-programmed.

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`
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* [452].

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* [306], *CmBoxPgm* [315] or *CodeMeter License Central* [337] are available (see Programming of CmContainer [305]).

2.4 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

Help File	Accessible by:	
	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
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

2.5 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.

2.6 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)
- operating system
- version of the *CodeMeter* software installed
- *CodeMeter* variant used
- detailed error description

2.7 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 130, 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, France, 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

Contact Information		
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

3 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.

WibuKey

Along with *CodeMeter*, Wibu-Systems offers *WibuKey. WibuKey* also encrypts software and secures licenses of digital products. The hardware (*WibuBox*) is very versatile and available in many form factors. Form factors range from PC Card and USB, and older interfaces, such as, COM and LPT, to integrated circuits (ASIC). Most of the applications, interfaces, and tools available for *CmDongle* and *CmActLicense* also work with *WibuKey.* For more detailed information please visit Wibu-Systems at www.wibu.com.

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.

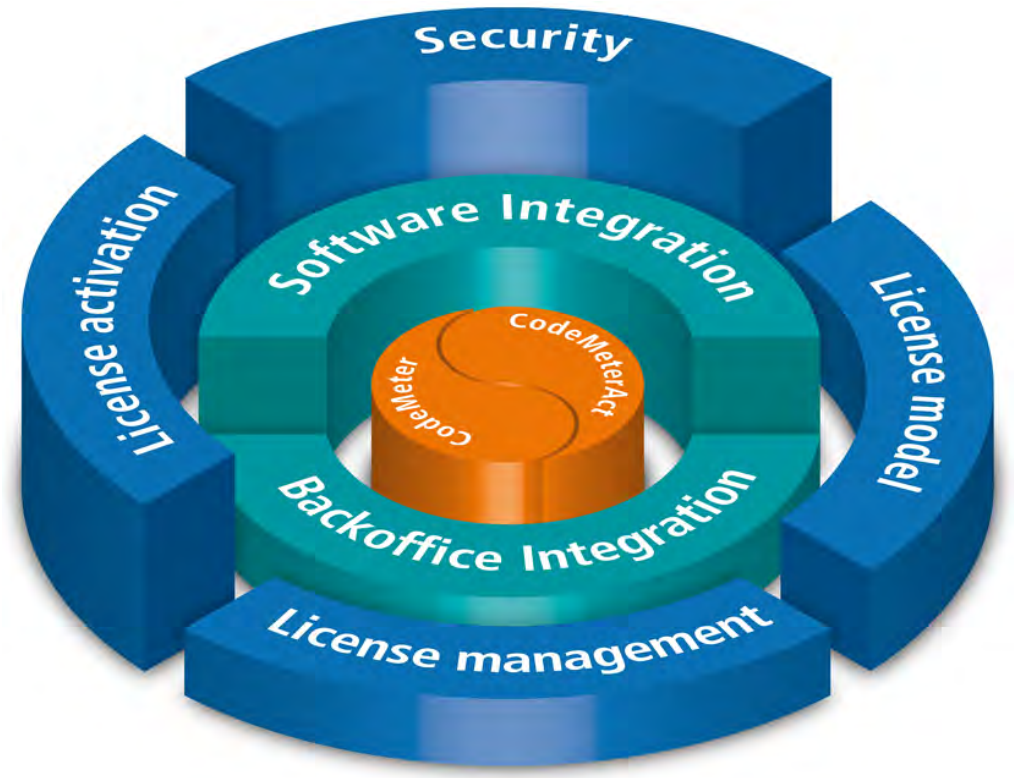


Figure 3: Overview - Software Protection and License Management using *CodeMeter*

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.
  - Individual advanced protection while developing an application using *IxProtector* by encryption and decryption of "real" source code fragments supported by interfaces (*Software Protection API, WUPI*) and security mechanisms.
- 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
  - 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, macOS, Java, .NET).
  - open customizable dialogs.
  - creation and further use of a commandline for *AxProtector* commandline.
- Individual integration of the protection into the software as additional protection results in ultimate flexibility and additional protection at runtime of an application.
  - Definition and protection of single areas and functions in the source code and, subsequently, link-up with variable license entries at runtime of the application using the protection technology *IxProtector* integrated in *AxProtector.*

    For an increase in protection, Wibu-Systems recommends the combination of automatic and individual integration. Moreover, security mechanisms of *AxProtector* and *IxProtector* are constantly developed and improved. After updates a recompilation of the application is not required, only a re-encryption with *AxProtector* or *IxProtector*.

  - Decryption and encryption of *IxProtector* protected areas at runtime using WUPI (*WIBU Universal Protection Interface)*. This lean *Software Protection API* providing few but essential functions is universally applicable for many programming languages.
- 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.
- 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 CmContainer Types

3.1.1 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

Form factor	Description	
	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 4: *CmDongle* Form Factors

1) This form factor can alternatively be configured as Human Interface Device (HID). For requirements and details see here 461 .

3.1.2 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.

3.1.2.1 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**

The dynamically weighted binding using the scheme SmartBind 329 optimizes assuring the validity of licenses, in the case of changing hardware properties of the computer or device to which the licenses are bound.

CodeMeter SmartBind uses a variety of hardware features and weighs it on the basis of internal algorithms tolerating minor changes without the need to always reactivate a license. The computer or device is still uniquely identified.

CodeMeter SmartBind provides an easy and secure way to bind a license to a computer. Using a variety of dynamically selected features it provides both reliability and security preventing manipulation. For more information on this technology see the separate document "SmartBind Whitepaper" available for download at the Wibu-Systems website.

In single cases, you are also able to set a tolerance level. It defines the allowed variation between the initial hardware configuration of the computer or device when the license was activated the first time and the current configuration.

> Wibu-Systems recommends *SmartBind* and the default tolerance level 2 (medium) as default binding scheme.
> For programming of *CmActLicense* licenses using the binding scheme *SmartBind* with *CmBoxPgm* see here 331 .

For single cases *CmActLicense* also supports binding schemes which refer either to specify fix 330 or configurable 330 hardware features of a computer or a device. However, Wibu-Systems recommends to contact Wibu-Systems support before using these options.

***SmartBind* on Linux armhf**

For *Firm Codes* greater than 6000000, *CodeMeter* Version 6.80 or higher also supports *CodeMeter SmartBind*. The setting of tolerance levels 329 depends on the kernel version used.

Kernel Versioh	Description
smaller than 3.18.6	defining the tolerance level: `1` (=tight), `2` (=medium) or `3` (=loose).
higher and equal to 3.28.6	defining the tolerance level: `3` (=loose).

This has been tested for Raspberry Pi. If you have any questions about the use of Raspberry Pi for other single-board computers and possible adaptations, please contact Support.

SmartBind and Azure

For Windows systems running on the Azure cloud computing platform, newly created *CmActLicense* licenses of Version 6.90 with the *CodeMeter SmartBind* binding scheme are now explicitly bound to the cloud computing platform. For Linux systems running on Azure this feature requires at least *CodeMeter* Version 7.0.

None Binding

Using the binding scheme None 330 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 333 " 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 334 " license model.

For both 'None-Bind' based license models a separate license entry in the *Firm Security Box* 27 (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.

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 330 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, starting with *CodeMeter* Version 4.40 you are able to create and deliver pre-calculated license 329 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.

3.1.2.2 CmActLicense Activation

Largely, activating *CmActLicense* licenses is based on the standardized *CodeMeter* procedure for file-based remote programming of *CmDongles* 342. 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 21 and additional activation options 22 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:

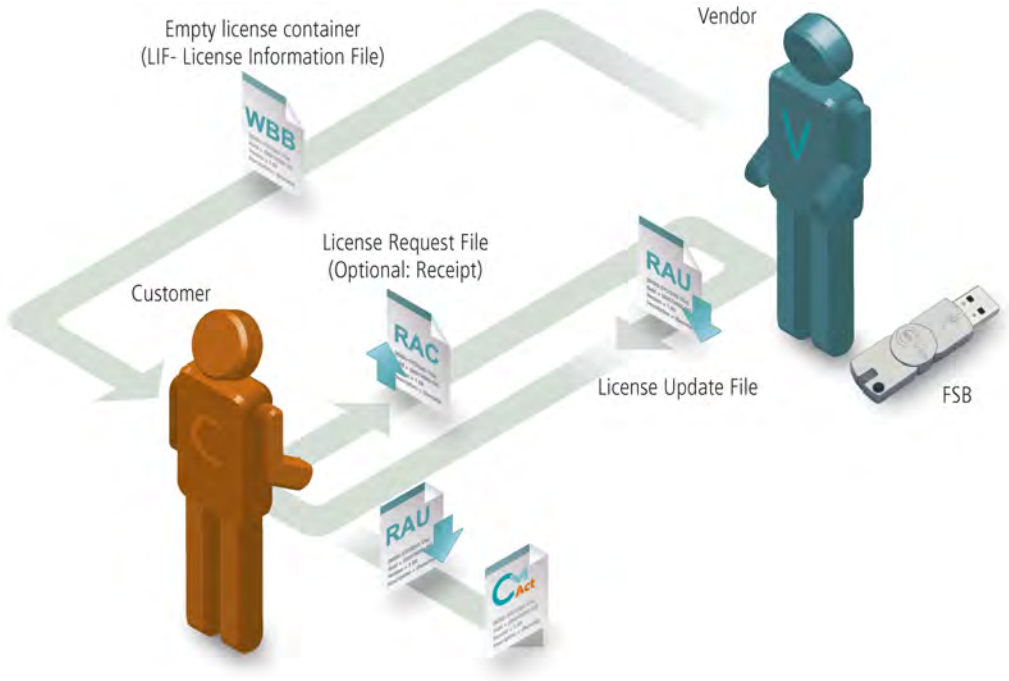


Figure 5: *CmActLicense* - Activation by file-based remote programming (CmFAS, *CodeMeter Field Activation Service*)

Activation by phone

Next to the standard activation of *CmActLicense* licenses also an activation by phone is available. In this case, instead of a license information file (`*.WibuCmLIF`) the customer receives by the vendor a pre-programmed, encrypted license container (`*.lip` file,), s/he then imports. A separate application at the licensor subsequently calculates a unique PC-specific Installation ID. This ID the licensee transfers to the licensor by phone. From this license Installation ID the licensor calculates the Activation Code, and transfers it to the licensee by phone. The license then activates the license container and is able to collect a license using this Activation Code.

3.1.3 CmCloudContainer: credential based licensing

The cloud-based version of *CodeMeter* allows online access to licenses in the cloud.

The *CmCloudContainer* contains the licenses of the end user. The *CmCloudContainer* is bound to an end user and is managed on *CodeMeter Cloud Server*.

The end user can access this *CmCloudContainer* from the local computer with appropriate credentials. This also means that only the credentials are located on the local computer, while the *CmCloudContainer* and the licenses are located in the cloud.

For this reason, licenses are not activated in a local *CmContainer*, but transferred to the *CmCloudContainer* on *CodeMeter Cloud Server*. The end user with the appropriate access data then accesses these licenses in an authenticated manner and can use them as if they were local licenses. *CmCloudContainer* and the contained licenses are integrated into the protection technology and licensing processes of *CodeMeter* as far as creation, delivery, modification and administration are concerned.

With *CodeMeter Cloud Manager*, the software vendor can create and manage these end user credentials. With the access data, end users can identify themselves and access the correct *CmCloudContainer*.

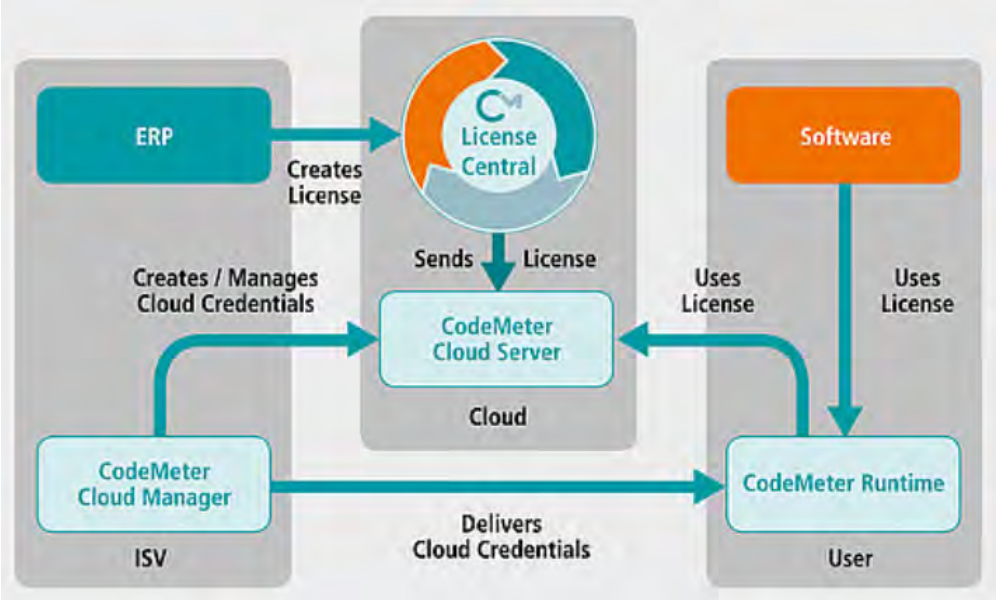


Figure 6: *CodeMeter Cloud*

3.2 Operating Systems supported by CodeMeter

CodeMeter is available for many operating systems and runtime environments, such as, Windows 32-bit/64-bit, macOS, Linux 32-bit/64-bit, Java, .NET.

Operating systems	CodeMeter
Windows XP	√
Windows Vista	√
Windows 7	√
Windows 8, 8.1, 10	√
Windows 2003 Server	√
Windows 2008 Server	√
Windows 2012 Server	√
Windows 2016 Server	√
macOS	√
Linux	√
Win 7 Embedded	√
Windows XP Embedded Service Pack 3	√
Windows CE 5.0	√
Windows CE 6.0	√
VxWorks	√

3.3 Additional Features

Additional Flash Memory and Mobile Applications

- In its version with additional Flash memory, *CmContainer* represents *CmDongle* and memory medium in one go, and involves the direct deployment of the software. The software can start directly without the need for separate installation on the attached system.
- *CmDongle* uses the SLC memory (Single Level Cell) suiting industrial needs. It is faster, more durable, and most robust against data loss compared to the MLC memory (Multi Level Cell) used in the consumer segment.

All Drivers on Board

CodeMeter is usable for many platforms via *CodeMeter License Server*. This background service communicates below with integrated operating system USB or Mass Storage Device driver with *CmDongle/CmActLicense*, and above with the provided *CodeMeter Core API*. No device drivers mean fewer calls to your support center.

License Server Settings

Local configuration options of *CodeMeter License Server* are provided by *CodeMeter Control Center*. *CmContainer* may run locally but also on the network. By default, *CodeMeter License Server* is installed as service or deamon (Linux, macOS) and automatically auto-

starts. When the service runs, other programs are able to access the licenses stored in *CmContainer* and to use protected data areas in a *CmContainer*.

Display of License Entries

Information about connected *CmContainer* and programmed license entries are displayed in *CodeMeter WebAdmin* which provides many configuration and analysis options.

3.4 CodeMeter as Token

CmDongle is used mostly for decrypting protected software and managing licenses. However, *CodeMeter* is also able to store certificates in established formats, such as, X.509. In order to use a cryptographic device as a token, the device has to be able to safely store and use secret keys. *CodeMeter* has always been able to do this using the *Secret Data* field. Moreover, the current firmware versions feature the use of the well-known asymmetric cryptographic algorithm RSA with a key length of 2084 bits. With both features *CodeMeter* fulfills all requirements to be integrated as a token. What has been missing so far was the option to apply these features using standardized system interfaces. The co-operation with charismathics now closes this gap and nothing stands in the way of using *CodeMeter* as a token in many applications.

Asymmetric Encryption

When encrypting asymmetrically, the private key is known only to the owner, while the public key may be widely distributed. The basic feature of asymmetry then is that the public key can be derived from the private key but not vice versa, i.e., the private key cannot be feasibly derived from the public key.

Public Key Infrastructure

Tokens require authenticity, signature verification, and encryption. The critical question here is: whom can I trust, and to what extent? Thus a basic prerequisite is a trusted Public Key Infrastructure (PKI) allowing all participants to verify the authenticity of the partner. This requires that keys are attested by a third party, i.e., a certificate authority. Then partners can verify that a certain public key does indeed belong to whoever partner is certified by the certificate authority. Several service provider offer such an infrastructure and, based on the X.509 standard, *CodeMeter* is able to store and use certificates issued by these providers.

Application Areas

When using *CodeMeter* as a token in PKI, along with some additional data the private key is saved within a X.509 certificate to the *CmDongle*. Using the certificate and the cryptographic procedures involved allow you to perform several tasks, such as, securing VPN access, signing and/or encrypting e-mails, and using strong two-factor-authentication for access control. Also you may use a certificate-based Windows login, authenticate for web-based applications (SaaS, or software as a service) or configure a company-wide single-sign-on for Windows.

Acting as Middleman

Charismathics Smart Security Interface (CSSI) middleware provides all token services for access, identification, and authentication and communicates function calls between the *CodeMeter* token and applications using the Windows proprietary CSP (Crypto Service Provider) and the generic PKCS#11 (Public Key Cryptography Standard) interfaces. The services then are available for Windows, macOS, and Linux.

Token and Dongle without Middleware

For proprietary applications you may simultaneously use *CodeMeter* as a dongle and token also without the CSSI middleware. If you do the key management yourself with the *CodeMeter Core API* you are able sign and encrypt own or existing keys applying the ECIES algorithm.

3.5 CodeMeter on Embedded Systems

Wibu-Systems provides *CodeMeter Embedded* for embedded devices which replaces the *CodeMeter License Server* and allows direct access to the *CmDongle* or *CmActLicense* from within your software.

CodeMeter Embedded is available as ANSI C source code or as a static library and can be compiled for your target system. An important feature of *CodeMeter Embedded* from Wibu-Systems is its modular design which allows you to streamline it into your project. It is the ideal alternative when installed in your own operating system or an embedded operating system.

An integration of *CodeMeter* into the real-time operating system VxWorks of Wind River and into the automation software CODESYS SPS of 3S-Smart Software Solutions GmbH is available.

4 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.

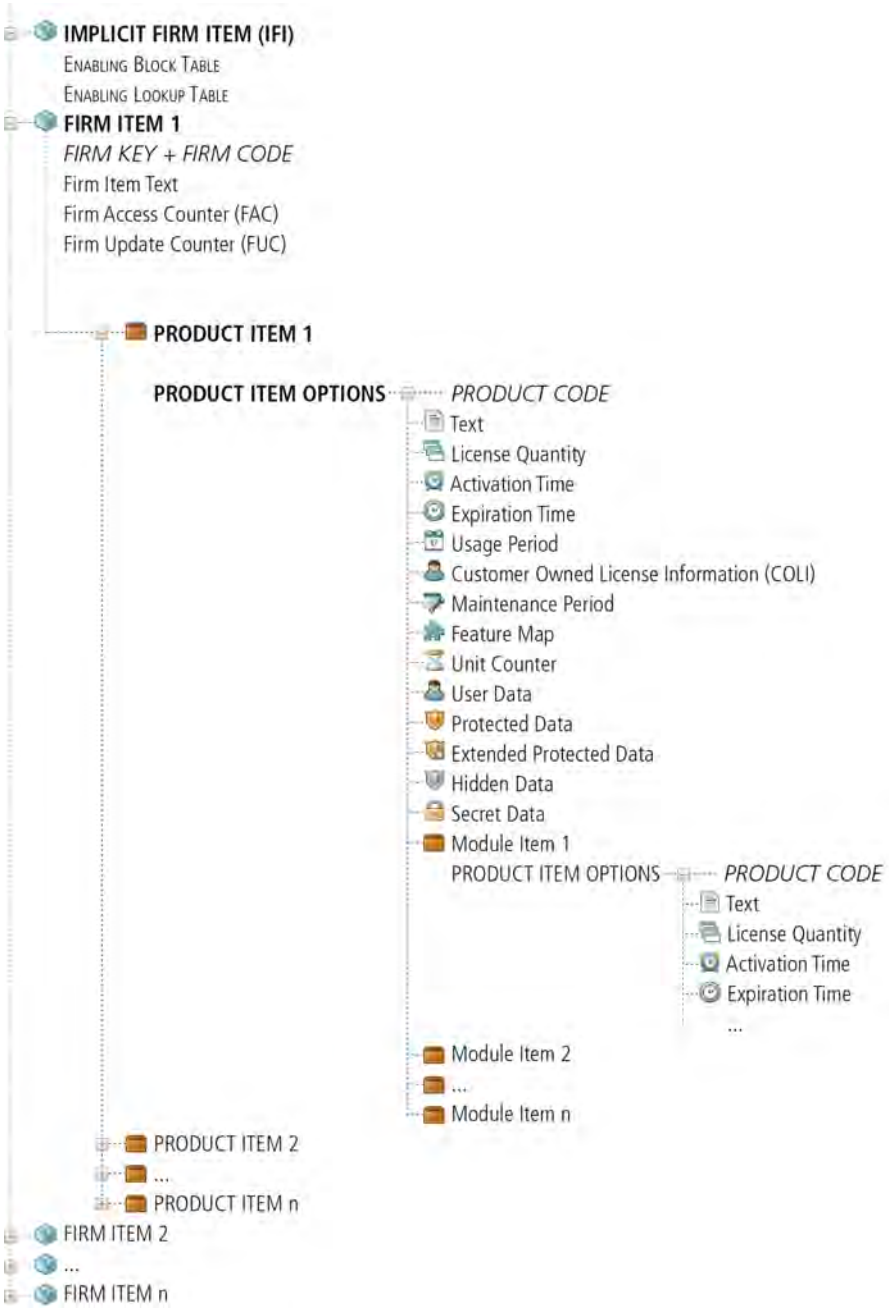


Figure 7: *CmContainer* License Entry Organization

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*.

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) 27.

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 40).

Module Items

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, one can add additional license properties to a *Module Item*. A *Product Item* may hold 0 to n *Module Items*. A *Module Item* itself cannot have *Module Items*.

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).

4.1 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	√

Product Item Option	Description	Read Access	Write Access	Integrated in Encryption
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	√
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 292 and writing 293 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.

4.1.1 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 311
CmBoxPgm	Programming the PIO /p
CodeMeter License Central	Programming 341 the PIO
Query/Check	
AxProtector	*Product Code* has to be defined.
Software Protection API (WUPI)	***WupiCheckLicense*** 291 or ***WupiAllocateLicense*** 291 ***WupiQueryInfold*** 291 Query information about the currently allocated license entry.
Core API	***CmAccess*** 298 and handle editing using ***CmCrypt*** 298

4.1.2 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 312 the PIO *Text*
CmBoxPgm	Programming the *PIO* **/pt** 326
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetProductItemText*** 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents***

4.1.3 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 312 the PIO License Quantity
CmBoxPgm	Programming the *PIO* **/plq** 322
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetAbsoluteLicenseQuantity*** 305 or ***SetRelativeLicenseQuantity*** 305
Query/Check	
AxProtector	License handling and License Options
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents*** 299

4.1.4 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* 30, 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 357.
- The *Activation Time* is part of the key derivation 44. 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 313 the PIO Activation Time
CmBoxPgm	Programming the *PIO* ***/pat*** 320
CodeMeter License Central	Programming 341 the number of days the application allowed to run from the first start
Programming API	Call class ProductItemParamSet 305 and subsequently SetAbsoluteActivationTime 305 or SetRelativeActivationTime
Query/Check	
AxProtector	Select options in advanced runtime settings
Software Protection API (WUPI)	***WupiQueryInfoId*** 291 Query information about the currently allocated license entry.
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents***

4.1.5 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* 30, 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 357.
- The *Expiration Time* is part of the key derivation 44. 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.

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 313 the PIO *Expiration Time*
CmBoxPgm	Programming the *PIO* ***/pet*** 321
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetAbsoluteExpirationTime*** 305 or ***SetRelativeExpirationTime***
Query/Check	
AxProtector	Selecting options in advanced runtime settings.
Software Protection API (WUPI)	***WupiQueryInfoId*** 291

Query/Check	
	Query information about the currently allocated license entry.
Core API	***CmAccess*** [298] and in Managing API ***GetBoxContents*** [299]

4.1.6 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.

4.1.7 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 [44]. 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 [312] the PIO *Unit Counter*
CmBoxPgm	Programming the *PIO /* [326] **puc**
CodeMeter License Central	Programming [341] the PIO
Programming API	Call class ***ProductItemParamSet*** [305] and subsequently ***SetAbsoluteUnitCounter*** [305] or ***SetRelativeUnitCounter*** [305]
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*.
Software Protection API (WUPI)	***WupiDecreaseUnitCounter*** [291] Decrementing of the Unit Counter of a license which in *AxProtector* is defined with the `Id = LicenseId` ***WupiQueryInfoId*** [291] Query information about the currently allocated license entry.
Core API	***CmCrypt***

4.1.8 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*.

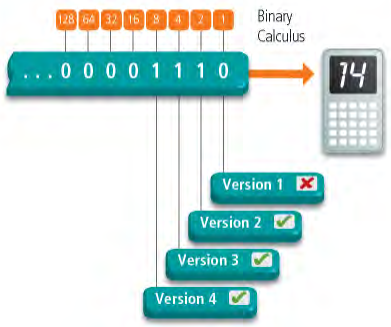


Figure 8: Version Management using PIO *Feature Code*

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 44 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 44. 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 313 the PIO *Feature Map*
CmBoxPgm	Programming the *PIO* **/pfm**
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetFeatureMap***
Query/Check	
AxProtector	Must be defined
Software Protection API (WUPI)	***WupiQueryInfold*** 291 Query information about the currently allocated license entry.
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents*** 299

4.1.9 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 313 the *PIO* *Maintenance Period*

Create/Edit/Delete	
CmBoxPgm	Programming the *PIO* **/pmd**
CodeMeter License Central	Not yet implemented
Programming API	Call class ***MaintenancePeriodParamSet*** 305 and subsequently ***MaintenancePeriodPIO*** 305
Query/Check	
AxProtector	May be activated and checked
Software Protection API (WUPI)	***WupiQueryInfold*** 291 Query information about the currently allocated license entry.
Core API	***CmCrypt2*** 298, ***CmAccess2*** 298 and in Managing API ***GetBoxContents*** 299

4.1.10 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 313 the *PIO* *Linger Time*
CmBoxPgm	Programming the *PIO* **/plt**
CodeMeter License Central	Not yet implemented
Programming API	Call class ***LingerTimeParamSet*** 305 and subsequently ***LingerTimePIO*** 305
Query/Check	
AxProtector	May be ignored
Core API	***CmAccess2***

4.1.11 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 313 the *PIO* *Minimum Runtime Version*
CmBoxPgm	Programming the *PIO* **/pmrt**
CodeMeter License Central	Not yet implemented
Programming API	Call class ***MinVersionParamSet*** 305 and subsequently ***MinVersionPio***.
Query/Check	
AxProtector	Option -D 264 (section licensing systems)
Core API	***CmAccess2*** 298 checks the current runtime with the programmed required *Minimum Runtime Version*.

4.1.12 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**
CodeMeter License Central	Not yet implemented
Programming API	Call class ***NamedUserParamSet*** and subsequently ***NamedUserPIO***
Query/Check	
AxProtector	---
Core API	***CmAccess2*** 298 and in the Managing API ***GetInfo***

4.1.13 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** 320
Programming API	Use classes ***CustomerOwnedLicenseInfoParamSet*** 305 and ***CustomerOwnedLicenseInfoPio***
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents*** 299

4.1.14 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 314 the *PIO* *User Data*
CmBoxPgm	Programming the *PIO* **/pud** 326
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetUserData*** 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents***

4.1.15 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 314 the *PIO* *Protected Data*
CmBoxPgm	Programming the *PIO* **/ppd** 325
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetProtectedData*** 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents*** 299

4.1.16 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 314 the *PIO* *Extended Protected Data*
CmBoxPgm	Programming the *PIO* **/ped** 320
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class ***ProductItemParamSet*** 305 and subsequently ***SetExtendedProtectedData*** 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	***CmAccess*** 298 and in Managing API ***GetBoxContents*** 299

4.1.17 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 292 and writing 293 of data from or into a *CmContainer* is featured also without FSB access at runtime using WUPI functions 290, 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 313 the *PIO* *Hidden Data*
CmBoxPgm	Programming the *PIO* **/phd** 321

CodeMeter License Central	Programming 341 the PIO
Programming API	Call class *ProductItemParamSet* 305 and subsequently *SetHiddenData* 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	*WupiReadData* 292 or *WupiReadDataInteger* 292, *WupiWriteData* 293 or *WupiWriteDataInteger*
Core API	*CmAccess* 298 and in the Managing API *GetBoxContents*

4.1.18 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.

> 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 314 the *PIO* *Secret Data*
CmBoxPgm	Programming the *PIO* **/psd** 325
CodeMeter License Central	Programming 341 the PIO
Programming API	Call class *ProductItemParamSet* 305 and subsequently *SetSecretData* 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	*CmAccess* 298 and in the Managing API *GetBoxContents* 299

4.1.19 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** 319
CodeMeter License Central	Activate this option
Programming API	Call class *ProductItemParamSet* 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	Managing API *GetBoxContents* 299 and *CmGetInfo*

4.1.20 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** 324
CodeMeter License Central	Activate this option
Programming API	Call class *ProductItemParamSet* 305
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	Managing API *GetBoxContents* 299 and *CmGetInfo*

4.1.21 Universal Data

The PIO *Universal Data* allows allows the creation and management of different data types: keys, data and passwords.

> Currently available for *CmDongle* with a minimum Firmware version of 4.30. *CmActLicense* and *CmCloud* are not supported.

- The PIO *Universal Data* comprises 2 x 32769 (65536) field indices of freely selectable length.

> Of the indices, 32769 (index 0-32768) are reserved for the software manufacturer (ISV) and 32769 (index 32769-65536) for Wibu-Systems.

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* **/puvd** 327
CodeMeter License Central	not yet implemented
Programming API	not yet implemented
Query/Check	
AxProtector	---
Software Protection API (WUPI)	---
Core API	not yet implemented

The introduction of Universal Data (*UvD*) PIO removes some of the restrictions that existed when using data fields, while making them more flexible to use. In addition, the changes extend the range of possible fields of application.

The maximum entry length of 256 bytes, for example, restricted the cryptographic use of keys to certain lengths. Also the storage of large amounts of data (Binary Large OBjects, BLOBs) was inflexible and required the management of multiple indices including padding. And finally, there were no procedures for using data fields, for example to protect against certain attack scenarios that, for example, require access to data fields at runtime without having a *Firm Security Box* (FSB). Necessary, for example, when programming a certificate store separately with the option of certificate imports and password management. The introduction takes these changed requirements into account.

Universal Data (*UvD*) is used as part of a highly configurable interaction of several components: three *UvD* data types can be assigned different access permissions for different access types. Since this is possible for accesses with a *Firm Security Box* (FSB) or for accesses at runtime, there are additional security options against unauthorized or unintentional mixing of access types to *UvD* data types.

Basically, a complete *UvD* entry is programmed via the *Firm Security Box* (FSB). The data type 37, all access types 37 and access permissions 37 including security options 37 are set and the possible password indices are linked.

UvD Data types

There are 3 different data types: 'Key', 'Data' and 'Password'.

> A data type change of a *UvD* entry, e.g. from type 'key' to type 'data' or vice versa, is not allowed without *Firm Security Box* (FSB) programming.

Data type	Description
Key	A total of 5 subtypes for the encryption algorithms AES and RSA are supported. For AES the key lengths 128, 256 and for RSA the key lengths 2048, 3072, 4096. A change of the subtype, e.g. from rsa4096 to rsa2048 or from rsa4096 to aes256 is allowed without *Firm Security Box* (FSB) programming, if the write permission for the *UvD* entry 'Key' exists. The keys can be set directly or a random generation can be selected within the *CmContainer*. The keys never leave the *CodeMeter* SmartCard chipo.
Data	The data can be entered directly or as a path to the input file.
Password	The password can be entered directly. The reading, writing and use of *UvD* data types can be controlled by passwords.

Access types and Access permissions

There are three ways to access a *UvD* entry:

Access	Description
(**r**)ead	read access to the *UvD* entry.
(**w**)rite	write access to the *UvD* entry, i.e.,creating and changing the data.
(**u**)se	utilizing access to the *UvD* entry.

The access permissions are ordered as follows:

Access	Description
permission denied [-]	prevents access to the *UvD* entry.
permission granted [+]	allows access to the *UvD* entry if a second password (master password) is set. The access matrix is as follows:
permission always granted [+]	always allows access to the *UvD* entry (neither a password1 nor a password2 are set)

Password1	Password2 (Master)	Status
set	not set	valid √
set	set	valid √
not set	not set	valid √
not set	set	invalid x

This results in the following combinations of access type and access permissions.

	Access permissions		
Access type	**denied**	**always granted**	**granted with password (1 or also 2)**
read	**r-**	**r+**	**r+ (p1/p2)**
write	**w-**	**w+**	**r+ (p1/p2)**
use	**u-**	**u+**	**r+ (p1/p2)**

Programming

The programming of a *UvD* entry should always take into account that access to *UvD* entries can take place via two different authorization systems: on the one hand via sole access by an FSB owner and on the other hand via read and write access at runtime, e.g. via the use of passwords.

If the FSB owner has exclusive access, for example, only the owner of the FSB can update the *UvD* entry and access content via a context file (`*.WibuCmRaC`). At the same time, however, a protected application may be allowed to use a key via an encryption call (*CmCrypt*...) during runtime.

During runtime access, read and write access to the *UvD* entries may be allowed, but no new *UvD* entries may be created. Write accesses, e.g. during an update, are limited to the user data and changing 'key' subtypes.

Security options

UvD data types can be accessed either with a *Firm Security Box* (FSB), or at runtime. With seals set by default, there are additional protection options designed to prevent *UvD* entries that can be accessed at runtime from being read or updated by the FSB owner without permission or intention, or from being accessed via a modified password usage.

The following list shows which protection option is active for which data types and accesses (x).

Seal	Description	FSB owner access		Runtime access	
`NO_READ`	Prevents the *UvD* entry from being read using a *Firm Security Box* (FSB).	Key	x	Key	x
		Data	√	Data	x
		Password	x	Password	x
`NO_UPDATE`	Prevents the subsequent modification of the *UvD* entry via a *Firm Security Box* (FSB) programming..	Key	x	Key	x
		Data	√	Data	x
		Password	√	Password	x
`INTEGRITY`	Invalidates a *UvD* entry as soon as a password *UvD* entry referenced to it has been deleted and integrity can no longer be guaranteed.	Key	√	Key	x
		Data	√	Data	x
		Password	√	Password	x

With the new PIO *UvD*, the mapping of previous data fields could look like this:

To set the parameters in *CmBoxPgm* see /puvd 327 ff.

- *Secret Data* as *UvD* data type 'Key':

  Useful, (**u+**) but not reading (**r-**) and writing (**w-**) access. As protection options for sole programming by the FSB owner `NO_READ` and `NO_UPDATE`.

- *Extended Protected Data* as *UvD* data type would correspond to 'Data':

  Read (**r+**), but not write (**w-**) access. Requires no protection options for sole programming by the FSB owner.

- *Hidden Data* as *UvD* data type 'Data' with access via a password:

  Reading (**r+**) access only with password (would correspond to *Hidden Data Access Code*), but not writing (**w-**) access. Requires no protection options for sole programming by the FSB owner.

4.2 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 has a status `"disabled"`
- 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

4.3 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 324
CodeMeter License Central	---
Programming-API	Calling class *ProductItemParamSet* 305 and then *ProductItemParamSet* 305
Query/Check	
AxProtector	----
Software Protection API (WUPI)	---
Core API	*GetBoxContents* 299 reads the original properties of the *Product Items* used as *Module Item*; *CmGetInfo* 299 reads the actual effective properties.

4.4 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.

Advantage	Description
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 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

4.5 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.
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.

License model	Description
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*

4.5.1 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 [305].

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.

4.5.1.1 Local Single User Licenses

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

e.g. Define a freely chosen *Product Code* at the Product Item level. Set the Product Item Option *License Quantity* for example in *CmBoxPgm* [322], to the value `local`.

4.5.1.2 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.

e.g. 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* [425]. 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.

4.5.1.3 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.

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* [357]).

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 [41]).

4.5.1.4 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.

4.5.1.5 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 357).

4.5.1.6 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.

4.5.1.7 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.

4.5.1.8 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.

4.5.1.9 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 417 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.

4.5.1.10 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.

4.5.1.11 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.

4.5.1.12 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 411 is displayed.

4.6 License Transfer

The license transfer features the transfer of licenses from one *CmContainer* to another *CmContainer*. Here a *Firm Security Box* (FSB) is not mandatory.

From a technical point of view, licenses are no longer transferred symmetrically 46 safeguarded but are stored as certificates in the context of asymmetrical cryptography 47 .

This also includes the standardization of the license formats for *CmDongle* and *CmActLicense* which become separate *CmContainer Types* identified by different CmAct Ids.

CmAct Id Range	*CmContainer Type*
1****	*CmDongle*
2****	*CmActLicense*
3****	*CmCloud* (not yet implemented)

Wibu-Systems grants Independent Software Vendors (ISV) the right for a license transfer. Subsequently. the ISV is able to decide whether or not a license transfer is part of its licensing strategy.

The license transfer is authorized by a private key which is required to create the required certificates.
If you decide for which *CmContainer Types* 323 you desire to authorize the license transfer, it is essential that you note that transferring this private key will allow the use of a protected application with such a transferred license.

The transfer of licenses from a sending to a receiving *CmContainer* currently involves the following transfer modes:

- ***Push*** transfer mode
  A license is transferred from a sending *CmContainer* to a receiving *CmContainer*.
- ***Return*** transfer mode
  A previously transferred license is returned from the receiving *CmContainer* to the sending *CmContainer*.

Requirements

1. The *Firm Codes* used must be greater than `6,000,000` (*Universal Firm Codes*).
2. The installed *CodeMeter* minimum version must be `6.0`.
3. If using *CmDongles* the mask number must have a value of `3`.
4. A *Firm Security Box* (FSB) must be available to program the license transfer.

Transfer options

Option	Description
Returning allowed	This option specifies whether the transfer mode Return is allowed.

Option	Description
Firm Item at target required	This option specifies whether on the target *CmContainer* a *Firm Item* must exist before the transfer takes place (*CmActLicense*).
Transfer Type	This option specifies the following types. **Transfer Type** / **Description** *Move licenses* — An existing *Product Item* with *License Quantity* is duplicated in two *Product Items* while the *License Quantity* is split over the two *CmContainer*. On return the licenses are merged. Multi-stage license transfers must be returned in reversed order. *Borrow local license* — A license is borrowed for local use only (without connection to a license server) for a definable period from one *CmContainer* to another. After the period has expired, the licenses automatically reallocates to the server's license pool. A locally borrowed license can not be further transferred. The license transfer depth has a value of 1.
Transfer Depth	This option specifies the number of steps the transfer is allowed to differ from the original license, if a license is transferred to other *CmContainer*.

License entries for the transfer are programmed using the commandline program *CmBoxPgm* 323 and are transferred file-based between sending and receiving *CmContainer* using the commandline program *cmu* 454 .

The interface to communicate with *CmContainer* at runtime of *CodeMeter License Server* provides *CodeMeter Core API* 297 .

The display of license transfers is part of *CodeMeter WebAdmin* 411 .

The license transfer feature for *CodeMeter License Central* is available with version 2.20 (only transfer type **Borrow local license**).

4.7 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.

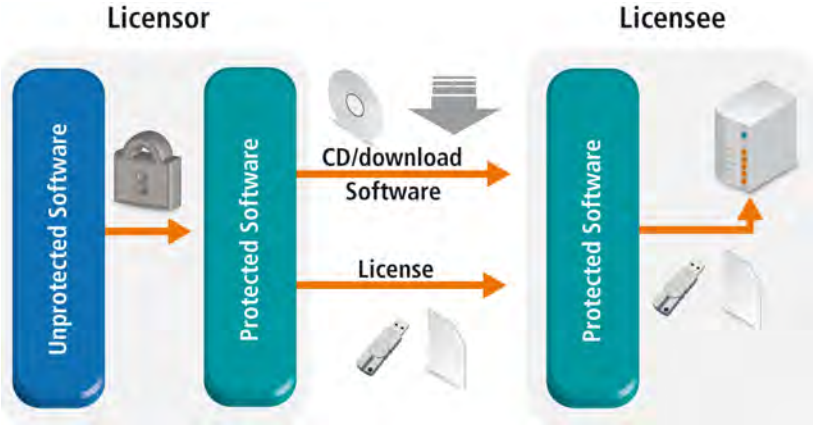


Figure 9: Security by Encryption

4.7.1 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". This process of the key derivation takes place within the *CmContainer* but the derived keys never leave the *CmContainer*.

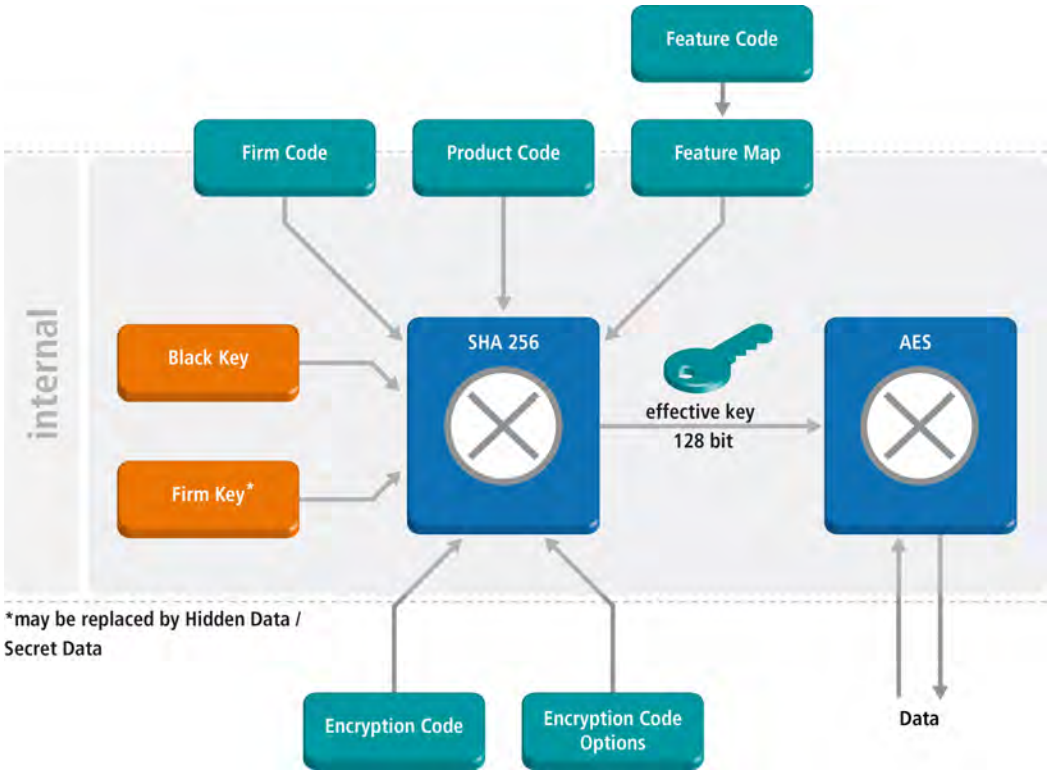


Figure 10: Key Derivation *CmDongle*

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*.

Alternatively, also a *Secret Data* or *Hidden Data* field may be used instead of the *Firm Key*.

Firm Code, Product Code, Feature Code, Release Date

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
- 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 but never leave the CmContainer.

4.8 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.

4.8.1 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.

4.8.2 Symmetric Enryption

In the case of symmetric cryptographic methods for encryption operations the same key is used (see *Encryption API* 298).

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.

4.8.2.1 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).

4.8.3 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 298 , Encryption API 298 , Blocks

4.8.3.1 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).

4.8.3.2 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`.

4.8.3.3 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* 298 needed for performing the authentication procedures.

4.8.3.4 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.

4.8.4 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.

5 CodeMeter Start Center

CodeMeter Start Center serves as communication center. It allows the access to basic *CodeMeter*® tools, applications, and interfaces.

5.1 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 11: *CodeMeter Start Center*

5.1.1 Menu Bar

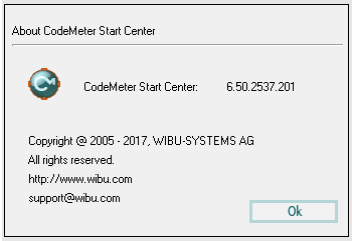
File Menu

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* 53 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* 300 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* 290 (WUPI) 290 .
CodeMeter License Editor	*CodeMeter License Editor* 306 is an intuitive graphical tool for creating, editing and deleting licenses in *CmDongles*. It supports locally connected *CmDongles* and also file-based remote programming 342 .
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*.

6 CodeMeter License Server

The central component of *CodeMeter* is *CodeMeter License Server* (`CodeMeter.exe`). It will run as a service on each computer where software protected with *CodeMeter* has been installed. *CodeMeter License Server* provides the interface between your software and the licenses stored in a *CmContainer*.

Many dongle manufacturers provide separate dynamically linked libraries (DLLs) for directly accessing the dongle. Wibu-Systems takes another path. Instead of DLLs we rely on our proprietary *CodeMeter License Server* to act as a central turntable providing all communication tasks for *CmContainer*. *CodeMeter License Server* communicates between the *CmContainer* using USB driver (as Mass Storage or Human Interface Device, HID [461]) provided by the operating system and the interface to your *CodeMeter* protected software.

Access Management - seamless, integrated and secure

Running as a background service, *CodeMeter License Server* manages all access from protected applications to *CmContainer*. It does not matter if several applications try to simultaneously access a *CmContainer* or if applications need license information stored in several *CmContainer.* And, of course, all communication both to and from *CodeMeter License Server* is secured using strong encryption.

Meeting Future Standards

Future hardware form factors will pose no problem for *CodeMeter License Server*. For example, software encrypted today, will run in the future on a SD card or CF Card. You will not need to adjust your software by programming a new interface. *CodeMeter License Server* automatically guarantees that your application will be executable. Moreover, backwards compatibility is also guaranteed. Even with future versions of *CodeMeter License Server* all delivered versions of your protected software will be executable; and this without recompiling your software.

Automatic License Allocation - local and network-based

CodeMeter License Server not only provides for automatic management of licenses on the local PC. Installed as a network server [425], it is also capable of managing all available licenses installed throughout the network. This means that once the maximum licenses have been allocated, a further instance of the protected application will not start. Different operation modes exist for issuing licenses. In the normal case, each instance of the application started by a different user reserves a license. However, selecting the option "station share" allows you to specify that the application can start any number of times by any number of separate users but only reserve one license per PC. In this mode, each terminal server session and each virtual machine is counted as a separate PC.

Since *CodeMeter* Version 5.0 the network communication includes also Wide Area Networks, WAN. For more details see here [361].

On installing [370] it can be decided whether *CodeMeter License Server* is set up as a server in a network environment and the related TCP port [52] 22350 is registered with the Windows firewall. By default, *CodeMeter License Server* is only available locally (local host). During the installation an automatic search of network servers is set as default. This is implemented by a broadcast via UDP (User Datagram Protocol) (it is listened only at server search time and only until the end of the UDP Waiting Time) and for communication the related UDP port [52] 22350 is registered with the Windows firewall.

Automatic and manual License Sharing

If, in rare cases, your application should unintentionally crash, *CodeMeter License Server* through constant checking of registered applications ensures that licenses are automatically shared again. In addition, an option exists allowing the administrator to manually share [412] licenses again.

CodeMeter Control Center and *CodeMeter WebAdmin*

Set local configurations for *CodeMeter License Server* in *CodeMeter Control Center* [384]. And *CodeMeter WebAdmin* [411] allows you to view and manage additional information on allocated licenses on the network. All communication between all components is based on the TCP/IP network protocol.

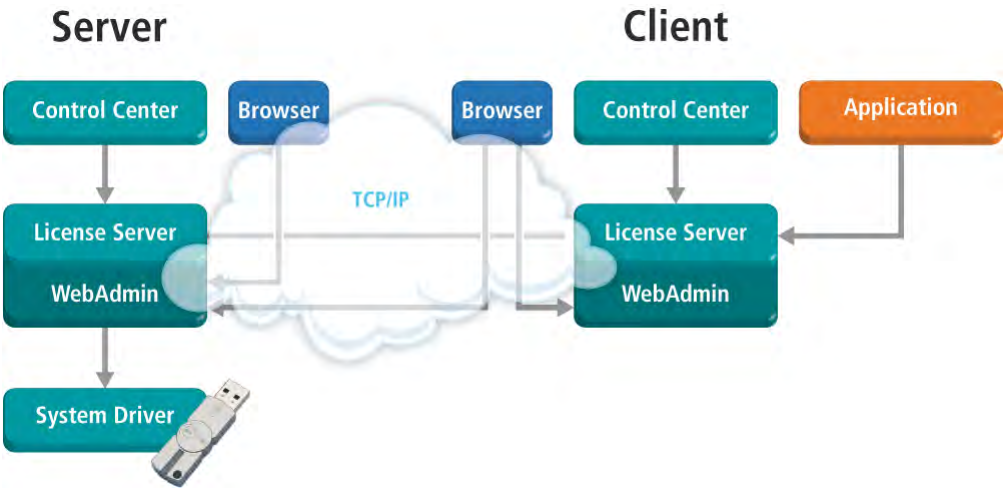


Figure 12: *CodeMeter License Server*

Diversity of Operating Systems

CodeMeter License Server is available for these operating systems: Windows 7, 8, 8.1, 10, Vista, Windows XP, Windows CE, Windows XP Embedded Service Pack 3, Win 7 Embedded, macOS, Linux (different 32 and 64-bit derivates), and VxWorks.

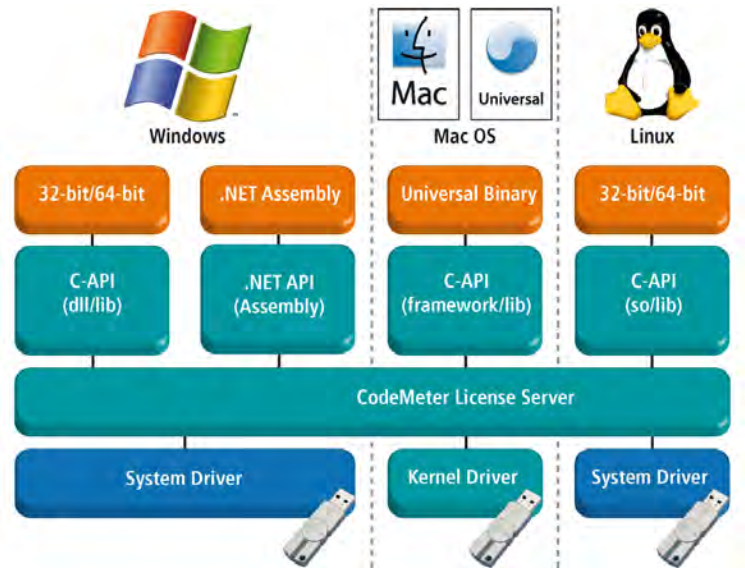


Figure 13: *CodeMeter License Server* and Operating Systems

CodeMeter License Server also feels right at home in heterogeneous system environments. For example, *CodeMeter License Server* may run as a network service on Linux, while your software runs on Windows and macOS in the same network.

For the various operating systems references to libraries are used which include all *CodeMeter* API functions.

Operating system	Library
Windows 32 bit, 64 bit	`WibuCm32/64.dll`
.NET	`WibuCmNet`
macOS	`WibuCmMacX Framework`
Linux	`libwibucm.so`
Java	Native library Java requires JNI (Java Native Interface): Windows: `WibuCmJNI[64}.dll`, macOS: `libwibucmjni.dylib` Linux: `libwibucmjni.so` For pure Java (*CodeMeter* Version 4.40 and higher) no native DLL is required.

.NET characteristics

For referencing to *WibuCmNet* depending on different .NET frameworks and policies the following characteristics exist.

Required *WibuCmNet* versions depend on:

- referenced *WibuCmNet* version and .NET framework at building time at the independent software vendor (ISV) and
- currently installed runtime on the end user system.

referenced *WibuCmNet* version and .NET framework at building time at the ISV	.NET framework used by the ISV at building:					
	< 4.0	≥ 4.0	< 4.0	≥ 4.0	< 4.0	≥ 4.0
	Currently installed runtime version on the end user system:					
	6.40		6.50		6.60	
CodeMeter 3.x	3.34	3.34	3.34	3.34	3.34	3.34
CodeMeter 4.x	6.40	6.40	6.40	6.50	6.40	6.60
CodeMeter 5.x	6.40	6.40	6.40	6.50	6.40	6.60
CodeMeter 6.0 - 6.40	6.40	6.40	6.40	6.50	6.40	6.60
CodeMeter 6.50	*	-	*	6.50	*	6.60

referenced *WibuCmNet* version and .NET framework at building time at the ISV	.NET framework used by the ISV at building:					
	< 4.0	≥ 4.0	< 4.0	≥ 4.0	< 4.0	≥ 4.0
	Currently installed runtime version on the end user system:					
	6.40		6.50		6.60	
CodeMeter 6.60	*	-	*	-	*	6.60

***** does not translate / application is neither build nor translated

- referenced *WibuCmNet* version is too new for end user system

The use of TCP/IP in *CodeMeter*

The communication between protected applications and *CodeMeter License Server* bases on the Transmission Control Protocol/Internet Protocol (TCP/IP). This is valid not only for locally existing licenses, but also for licenses which are provided via a network.

By default, *CodeMeter* uses the port 22350 registered by Wibu-Systems at IANA (Internet Assigned Numbers Authority) and uniquely assigned for the *CodeMeter* communication. The list of assigned ports can be viewed at www.iana.org/assignments/port-numbers.

In order to make available a client access to a *CodeMeter License Server* on the network 370, a communication using the *CodeMeter* port must be supported. If the server should locate in another network area, eventually the port must be made known and accessible as part of the infrastructure (router, firewall, etc.).

For the direct access to *CodeMeter License Server* on the network, the communication bases on TCP. For an automatic search of servers 370 on the network, additionally a broadcast via UDP (User Datagram Protocol) is performed (it is listened only at server search time and only until the end of the UDP Waiting Time).

> The access using the *CodeMeter* port is performed only for the access to *CodeMeter License Servers* and this only within the organization which runs the network server.
>
> In particular, using this port **no** communication into the internet is performed.

In *CodeMeter* settings of *CodeMeter WebAdmin* an option exist to configure 425 the *CodeMeter* port to a value other than the default of 22350. However, such a change should have plausible reasons, e.g. in the case of parallel test environments on the same network. In addition, such a change requires the same configuration of all affected *CodeMeter License Servers*.

> If another, different port is used, settings for operating mode 370 and port communication 370 must be made manually.

7 Automatic Software Protection using AxProtector (Tool of CodeMeter Protection Suite)

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 56

AxProtector is also available as a commandline variant 263 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.

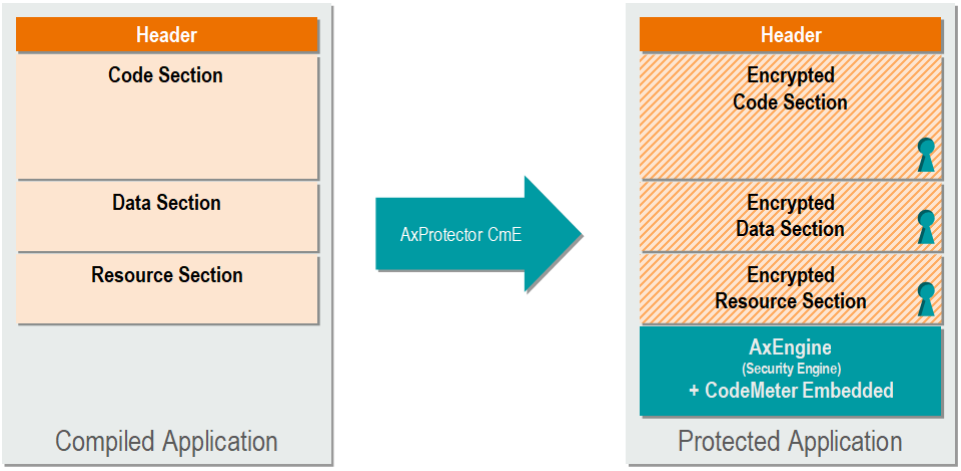


Figure 14: *AxProtector CmE*

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
Windows Application or DLL	AxProtector Windows 57 IxProtector Windows	√	Windows commandline
NET Assembly	AxProtector .NET 82 IxProtector .NET	√	.NET commandline
.NET Standard 2.0 Assembly	AxProtector .NET Standard 2.0 105 Ix Protector .NET Standard 2.0	√	.NET commandline
macOS Application or Dylib	AxProtector macOS 128 IxProtector macOS	√	Windows commandline Commandline available for macOS (runs on macOS operating systems)
Java Application (Archive Format `*.jar`, Webarchive Format `*.war`)	AxProtector Java	√	Windows commandline Commandline available for Java (runs on Windows, macOS, and Linux operating systems)
Linux Application or Shared Object	AxProtector Linux 174 IxProtector Linux	√	Windows commandline Commandline available for Linux (runs on Linux operating systems)
Files your protected application uses	AxProtector File Encryption	√	Windows commandline
Linux, Android, Windows Embedded			Windows commandline

Table 5: *AxProtector* – Applications to be protected, Project Types, and Encryption Tools

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.

7.1 Structure and Navigation

You access *AxProtector* by using *CodeMeter Start Center* 48 or, alternatively, by the **"Start | All Programs | AxProtector"** start menu item (Press "Windows" key to open Start screen | Type "AxProtector" | Press "Enter" key).

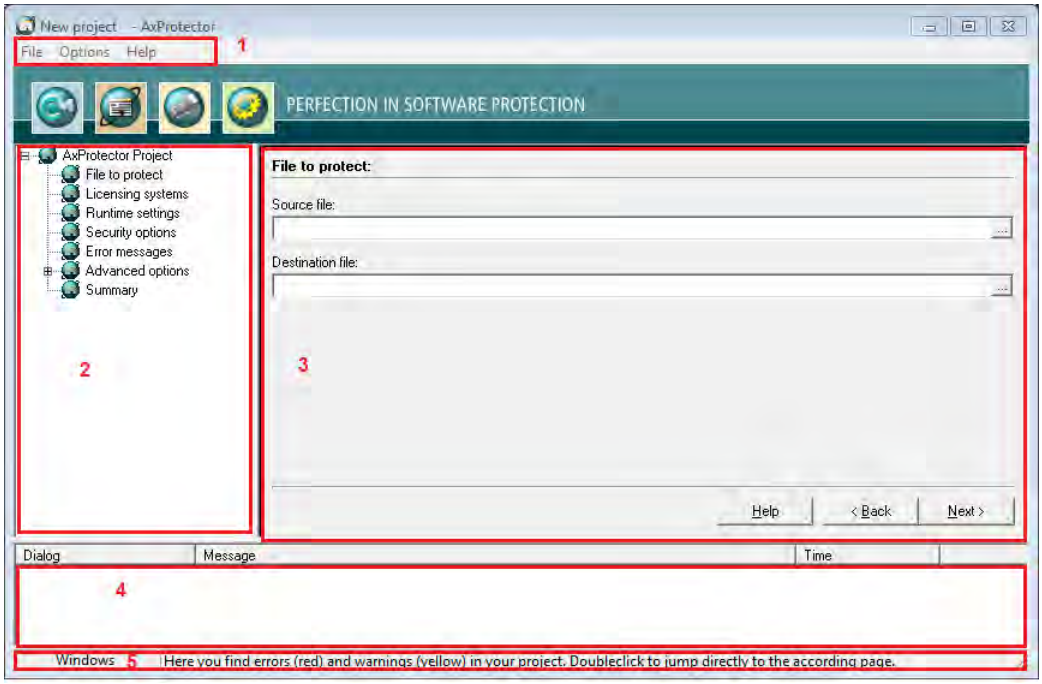


Figure 15: *AxProtector* – GUI and Navigation

The *AxProtector GUI* consists of five separate areas:

- Menu bar 54 **(1)**
- Navigation window 56 **(2)**
- Input window 56 **(3)**
- Note and error window 56 **(4)**
- Project type area 56 **(5)**

7.1.1 Menu Bar

File menu

Element	Description
Project	**New Project** To create a new project, please proceed as follows: **1.** Select the **"File | New Project"** menu item. Alternatively, press the <CTRL+N> key combination. The **"New Project"** dialog opens for selecting the project type. **Open Project** To open an existing project, please proceed as follows: **1.** Select the **"File | Open Project"** menu item. Alternatively, press the <CTRL+O> key combination. The **"Open"** system dialog opens from which you can select the desired project file. **2.** Select the project file name to be opened, and click the **"Open"** button. **Save Project** To save a created or edited project, please proceed as follows: **1.** Select the **"File | Save Project"** menu item. Alternatively, press the <CTRL+S> key combination. **Save Project as** To save an opened project using another project name, please proceed as follows:

Element	Description
	1. Select the **"File | Save Project as"** menu item. **2.** Select a destination folder in the **"Save as"** window and specify the new name of the project file. If this file already exists, *AxProtector* prompts with an overwrite confirmation dialog. Click on the **"No"** button and save the project using a different name, to keep the existing project file.
Export	Selecting this menu item exports the protection settings into a `*.wbc` file you are free to name and save. Later you may use this file in the *AxProtector commandline tool* 263. This menu item is active only after the project has passed all necessary checks.
Exit	Select the **"File | Exit"** menu item to close *AxProtector*. Alternatively, close the AxProtector by the **"x"** control or the `<ALT+F4>` key combination. Before exiting AxProtector you are prompted to save the changes you have made to a project.

Options menu

Element	Description
Language	*AxProtector* provides you with different language version for the graphical interface. Select from eight different language settings: Chinese, German, English, Spanish, French, Japanese, Italian, and Russian.
Show Advanced Llcensing Options	Advanced licensing settings display on dialogs, e.g. Linger Time for Licensing Systems.

? menu

Element	Description
Content	Select this menu item to open the *AxProtector* online help.
About	Select this menu item to open a window holding *AxProtector* version information.

7.1.2 Navigation Window

For every project type, the navigation window displays the single protection steps in a tree view. The navigation allows you to access each single step.

7.1.3 Input Window

For each protection step, the input window provides for specifying protection options using corresponding fields and controls. You navigate through the single steps by using the **"Next >"** or **"< Back"** buttons at the bottom of each window.

This symbol informs you that you have set additional protection options using the **"Advanced"** button.

7.1.4 Note and Error Window

This window displays information, errors or warnings using symbols. You also see the symbols in front of each protection step within the tree view.

Symbol	Description
	When setting an option an error occurred. The protection step involved is not executed. A text informs you about what the error might be. Then you have the option to check your input.
	Please note a warning related to the options you set when protecting your application.
	All settings are correct. This protection step is will be executed.

With a double-click on the and symbols you will automatically access the protection step to which the information relates.

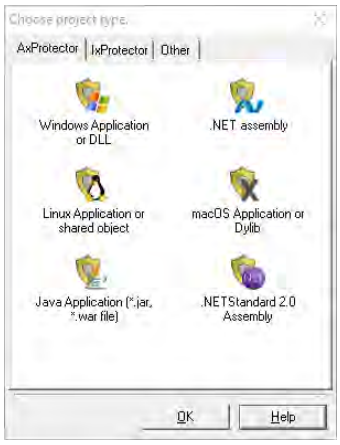
7.1.5 Project type area

This area displays which project type you currently working with and shows the content of existing tooltip texts when you move your mouse over dialog elements.

7.2 Project Dialog

When you open *AxProtector* or create a new project in *AxProtector* a project dialog opens where you make the selection from different project types.

The tabs "**AxProtector**", "**IxProtector**" and "**Other**" show all available project types.



You receive help by clicking on the **"Help"** button.

7.3 Project Types

AxProtector features the following project types:

Icon	Project type
	AxProtector
	Windows Application or DLL

Icon	Project type
	.NET Assembly
	.NET Standard 2.0 Assembly
	macOS Application or Dylib
	Java Application (jar file)
	Linux Application or Shared Object
	IxProtector
	Windows Application or DLL
	.NET Assembly
	.NET Standard 2.0 Assembly
	Linux Application or Shared Object
	macOS Application or Dylib
	Other
	File encryption

7.4 AxProtector Tab

This tab offers you the selection of the following project types:

	Windows Application or DLL
	.NET Assembly
	.NET Standard 2.0 Assembly
	macOS Application or Dylib
	Java Application (jar file)
	Linux Application or Shared Object

7.4.1 Windows Application or DLL

AxProtector protects executable files (applications `*.exe` and libraries `*.dll`) in PE format (Portable Executable). The executable files may be created by established compilers, for example, (C, C++; Delphi, VB 6.0, FORTRAN, ...), or by authoring tools (Adobe Flash, etc.).

The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
Windows Application or DLL	AxProtector Windows	√	Windows commandline

The following menu items are available in the navigation windows:

- File to protect 58
- Licensing systems 58
- Runtime settings 64
- Security options 67
- Error messages 71
- Advanced options 72
    - License lists 72
    - IxProtector 77
    - File encryption 79

- Summary

7.4.1.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

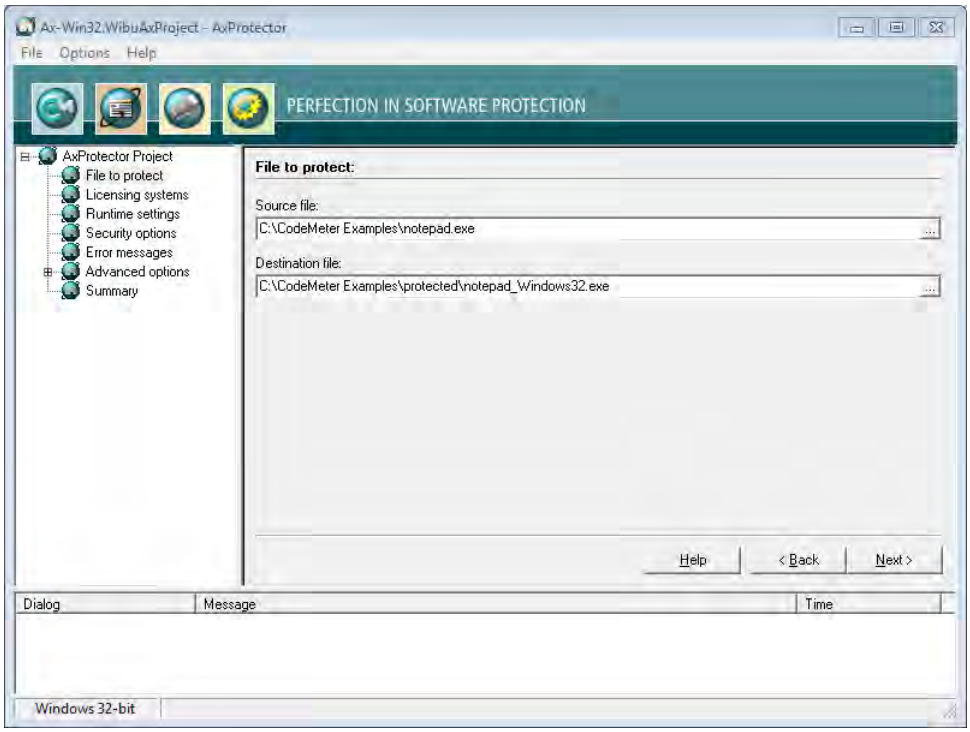


Figure 16: *AxProtector* - Windows "File to protect"

File to Protect

Element	Description
Source file	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination file	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.4.1.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

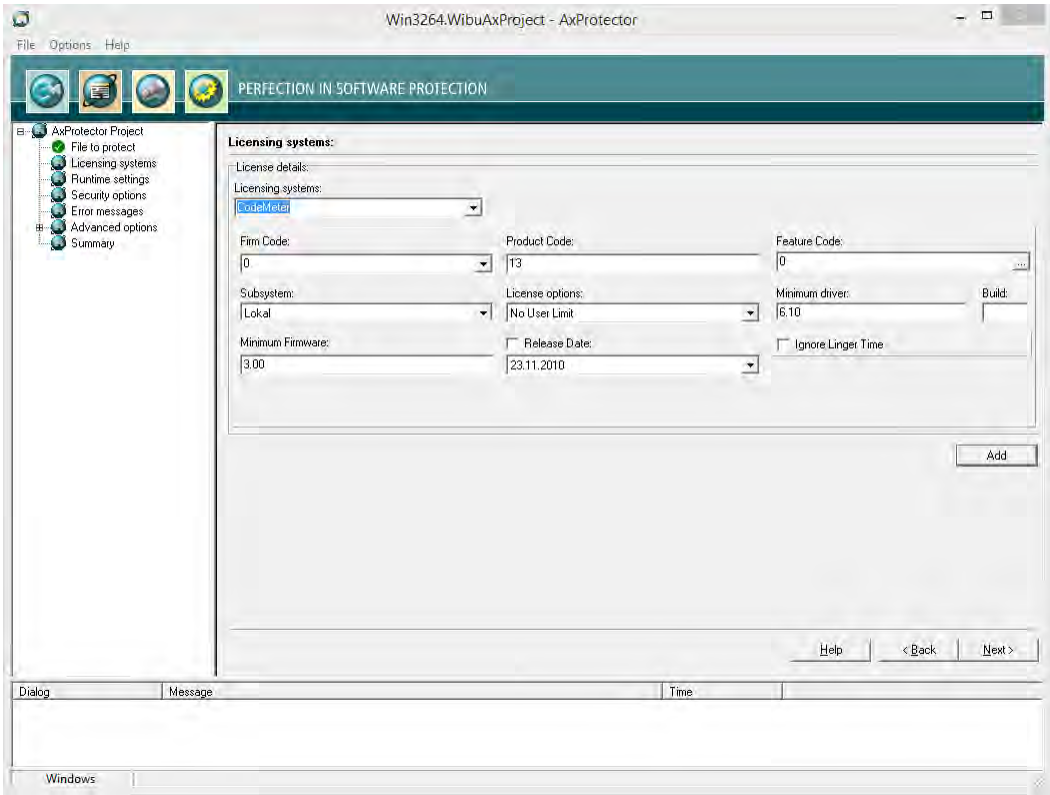
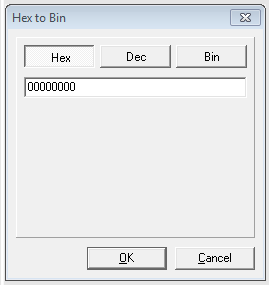


Figure 17: *AxProtector* - Windows "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: *Firm Code CodeMeter* Software Development Kit (SDK) / Licensing system `6000010` Evaluation *Universal Firm Code* / *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* / *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* / *CmActLicense* Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.

Element	Description
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 18: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local — This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network — This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network — This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local — This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit — Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share — Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode — Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode — Here a protected application can be started only once on a PC. No user limit — Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*) — 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal

Element	Description
	Firm Codes (licensing system) / Version (*CmActLicense Firm Code*) servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) / 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) / 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

7.4.1.2.1 Licensing system - Add licenses

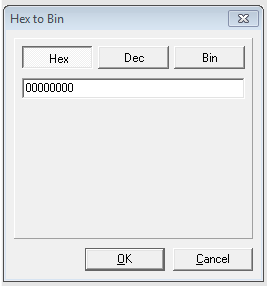
Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description

Licensing systems	Select from the dropdown control the desired licensing system. Available are the following entries:
	`CodeMeter`
	`WibuKey`
	For setting *WibuKey* options, see the separate "WibuKey Developer Guide".
	If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.
	In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: *Firm Code CodeMeter* Software Development Kit (SDK) / Licensing system `6000010` Evaluation *Universal Firm Code* / *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* / *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* / *CmActLicense*

Element	Description
	Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 19: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Firm Codes (licensing system)	Version
`6000010; >= 6.000.000` (*Universal Firm Code*)	6.10 This supports the License Transfer feature.
`10, 100.000-4.999.999` (*CmDongle Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses.

Element	Description
	Firm Codes (licensing system) / Version Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*): 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*): 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.1.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

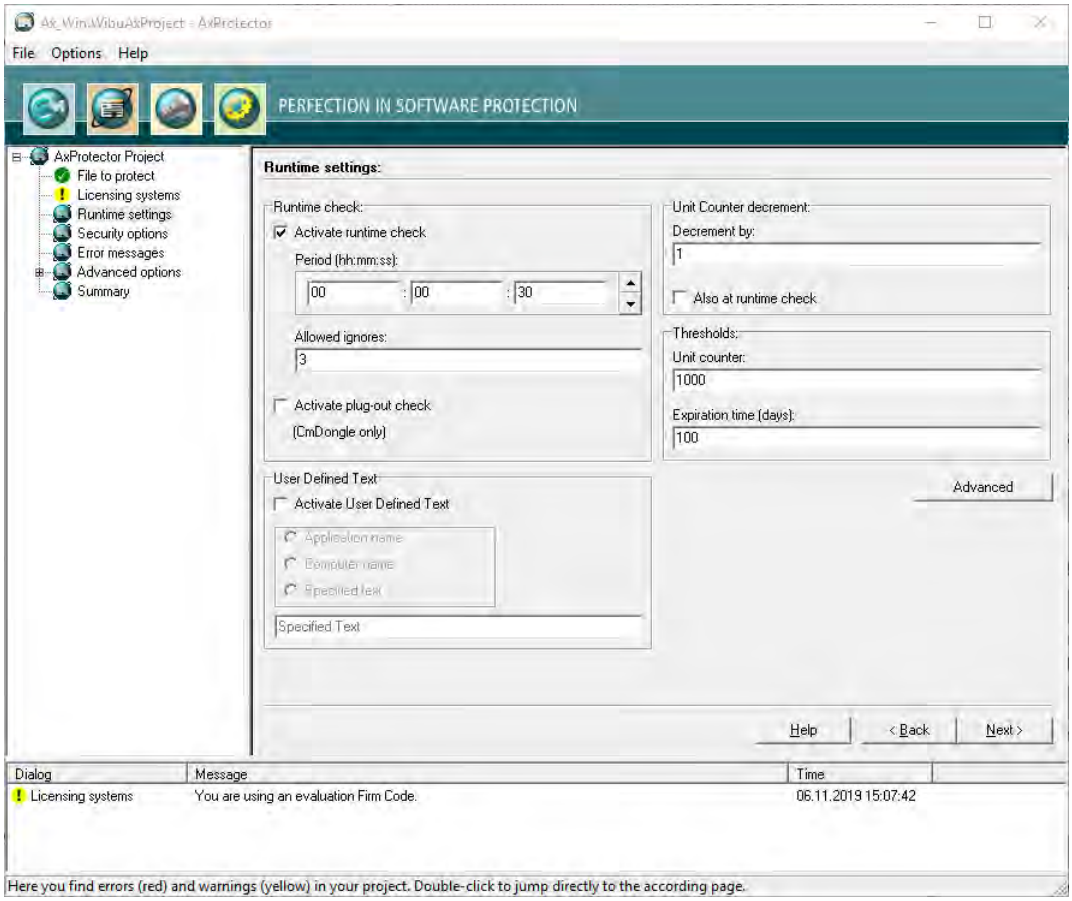


Figure 20: *AxProtector* - Windows "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: `hours: minutes: seconds`.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.
Activate Plug-out Check (only *CmDongle*)	This option closes the protected application when the *CmDongle* is removed while the application is running. Immediately, an error message is issued. This option is valid for *CmDongle* only. Commandline option see here 267 .

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the *Unit Counter* is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above every 30 seconds (see the defined period) a set *Unit Counter* is decremented by a value of 1.
Also at Runtime Check	Decrements the *Unit Counter* also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime Check 64" group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here[71].

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here[277].
Expiration Time (days)	If the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here[277].

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here[279].

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. Element / Description: Application name — uses the application name. Computer name — uses the computer name. Specified text — uses the specified text in the field of the same name.

7.4.1.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

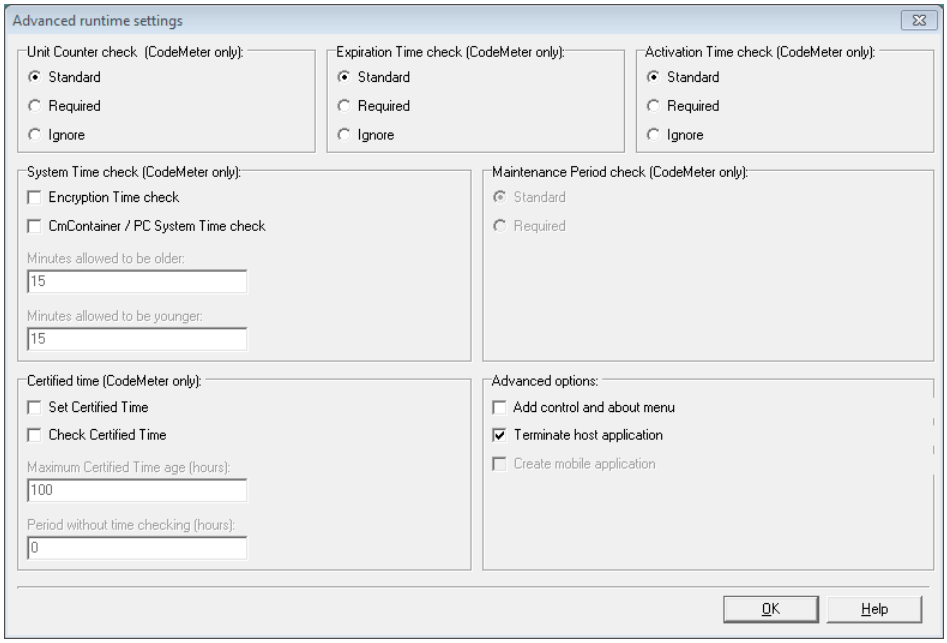


Figure 21: *AxProtector* - Windows "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignore
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here[276]).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches a value of 0 (null) the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts if no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, if the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time 357 is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, if the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 (commandline option see here 276).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems 61.

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only in the case a *Maintenance Period* exists. This corresponds to the default setting, even when on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmContainer* is connected with the computer or activated. When the *CmContainer* is connected or activated, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *Certified Time*. On updating the *Certified Time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here 270).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here 357 ..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmDongle*. The *Certified Time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check", you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time.*
Period without time checking (hours)	Specifies the period (in hours) when no check of the *Certified Time* certificate is performed. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here 267).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only when the *CmContainer* System Time is newer than the encryption time.

	Requires at least *CodeMeter*® 4.10.
CmContainer / PC System Time check	If activated, these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor, the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

Advanced options

This group allows to set further options.

Element	Description
Add control and about menu	Adds the "About" and "Control" menu items to your application (commandline option see here 270).
Terminate host application	When no valid license is found, in the case of protected DLL application files the calling `*.exe` is terminated (commandline option see here 277).
Create mobile application	[not yet implemented]

7.4.1.4 Security Options

This input window lets you select from different mechanisms and methods for protecting your application. You are able to scale the degree of security for yourself, for example, how intensive the search for debugger is to be, or whether a *CmContainer* is locked.

If the options you set here turn out to be incompatible with your protected application, you are also able to separately deactivate single security options.

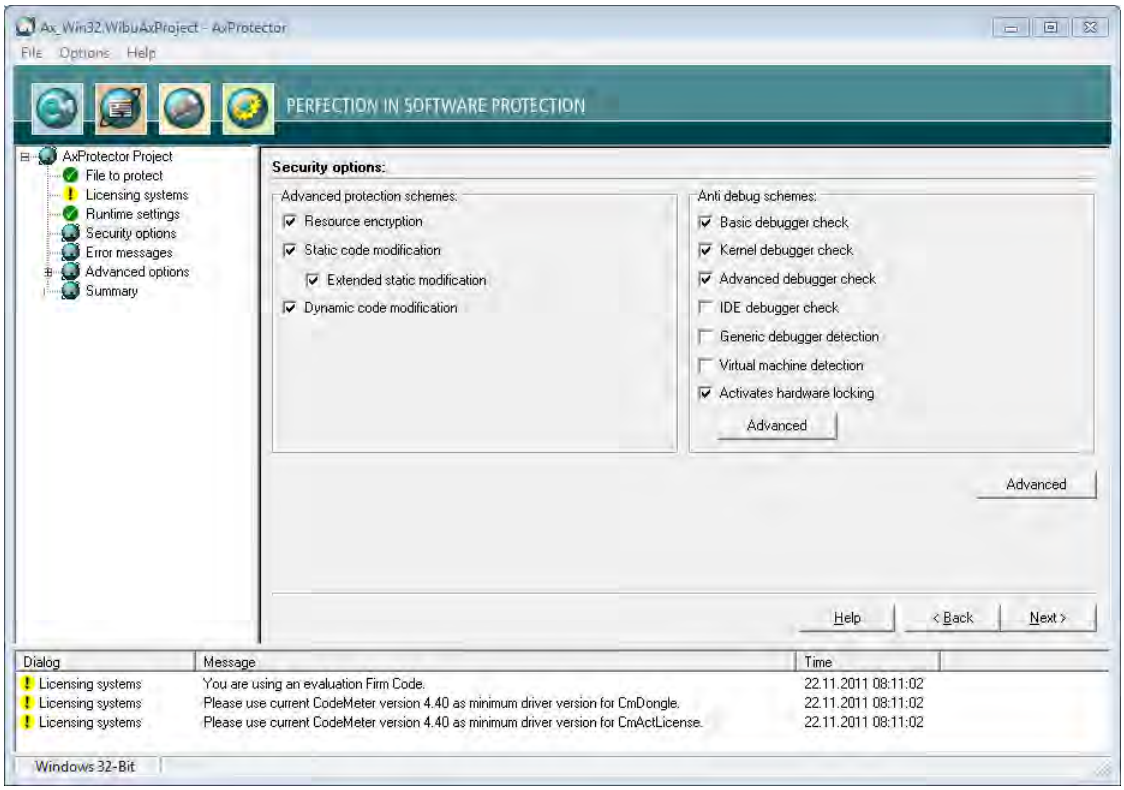


Figure 22: *AxProtector* - Windows "Security Options"

Advanced Protection Schemes

The advanced protection schemes deeply intervene into your application. In some cases, this may mean that some single mechanisms will not work due to compatibility reasons (commandline options see here 266).

Element	Description
Resource Encryption	Also encrypts the resources of your protected application. After the start of your application, the resources located in the PC memory and are decrypted "on demand".
Static Code Modification	Your software is modified in a way so that it is protected against debugging, dumps and reverse engineering. These modifications are added to your application when encrypted.
Extended Static Modification	This option adds extended multi-nested security mechanisms to the static code modification.

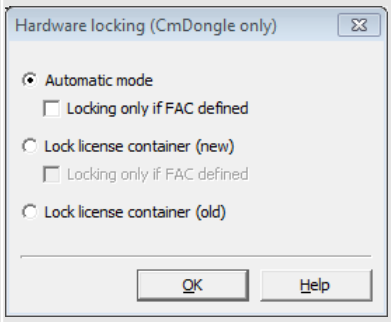
Element	Description
Dynamic Code Modification	The source code of the application to be protected is modified dynamically at runtime of the application.

The options "Static Code Modiification" and "Extended Static Modification" conflict with an activated option "Activate Automatic File Encryption 72 " on page "Advanced Options".

Anti-Debug Schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here 267).

Element	Description
Basic Debugger Check	Checks if a debugger is attached to your application. If a debugger is found, your application will not be started or exited.
Kernel Debugger Check	Additionally checks for Kernel debugger programs, such as, SoftICE. If a debugger is found, your application will not be started. The next two mechanisms comprise methods for detecting specific debugger programs and tools.
Advanced Debugger Check	Checks in an advanced search for debugger programs which may run parallel to your application, also cracker tools, such as, ImpREC, are detected. If a debugger is found, your application will not be started.
IDE Debugger Check	Checks for all debugger programs. With this option, debugger programs are not allowed at all, i.e. even within developer environments, e.g. Visual Studio, Delphi. If a debugger is found, your application will not be started.
Generic Debugger Detection	Adds a mechanism to the application preventing the attachment of a debugger program to the application at runtime.
Virtual Machine Detection	Detects if the application is to be started on a virtual machine, and prevents this.
Activates license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the **"Configuration"** button. This button is activated only for *CodeMeter.*
Configuration	If the option **"Activates license access lock"** is activated, you are able to define further settings in the dialog which opens by clicking the **"Configuration"** button: Depending on the Firmware used this dialog allows to define separate locking scenarios (for more detailed information see separate CodeMeter Developer Guide, section "Advanced CodeMeter Features | Locking a CmContainer").

Locking Scenario	Description
immediate locking	is performed starting with Firmware Version 1.14, as soon as a debugger is detected.
prepared locking	is performed by checking the *Firm Access Counter* (FAC). The *Firm Access Counter* locates 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`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked. The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.  Figure 23: *AxProtector* - Windows "Security Options - Hardware Locking"

The following settings are available:

Option	Description
"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher, the *Firm Item* is immediately locked. For compatibility reasons this represents the default setting.

Element	Description	
	Option	**Description**
	"Automatic Mode" activated and "Locking only if FAC defined" activated	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher and a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
	"Lock License Container (new)" activated and "Locking only if FAC defined" not activated	This option requires a Firmware version 1.14 or higher. The *Firm Item* is immediately locked. Seen from a security point of view this is the recommended option. This, however, requires that all *CmContainer* in the field must have a Firmware Version 1.14 an higher.
	"Lock License Container (new)" and "Locking only if FAC defined" activated	This option requires a Firmware version 1.14 or higher. If a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
	"Lock License Container (old)" activated	Independent from the Firmware version, if a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. This holds for all Firmware versions. If 'prepared locking' is programmed, the *Firm Access Counter* is decremented by a value of `1`.

7.4.1.4.1 Advanced Security Options

This input window lets you define further settings.

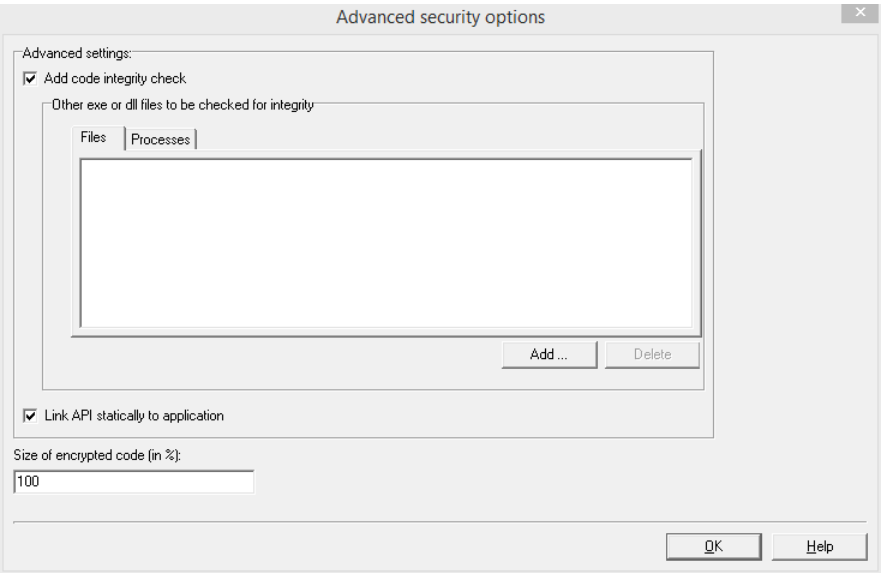
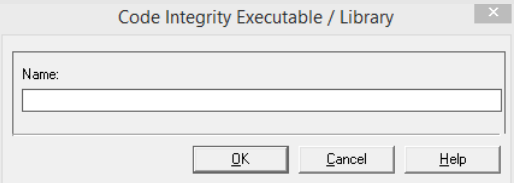
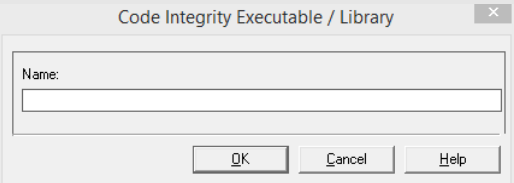


Figure 24: *AxProtector* - Windows "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

Element	Description
Add code integrity check	The protected application is checked for code integrity using asymmetric authentication asymmetric authentication 47 mechanisms, if you check this box (commandline options see here 270). On code integrity check first a check sum (hash value) of the application is created and signed with the private key of the Individual Software Vendor (ISV). The hash value and the signature are added to the application. The recalculation and the integrity check of the hash value and thus of the application is performed at runtime check using the public key located in the software (AxEngine). Alternatively to the default private key you can also apply the commandline option -sig 266 to use an entry of a *Hidden* or *Secret Data* field to define another private key. Moreover, the code integrity check may also cover several executable files / libraries. Then each file is able to check all other files for integrity. Each file then requires the public key of the ISV: The hash value of the files to be checked then is recalculated and compared to the hash value signed with the private key. To add other files for performing an integrity check, please proceed as follows. **1.** Set focus to tab "**Files**". **2.** Click the "**Add**" button. The dialog for adding displays.

Element	Description
	 **2.** Add a single or several executable files / libraries by completing the "**Name**" field. The sequence of the specified files does not matter. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button. Moreover, on encrypting a DLL also a list of applications can be transferred allowed to load these libraries. On loading the DLL then it is checked whether the process name includes one of the names specified in tab "**Processes**". If not, an error message displays and subsequently the application closes. To add processes please proceed as follows: **1.** Set focus to tab "**Processes**". **2.** Click the "**Add**" button. The dialog for adding displays.  **3.** Add one or more processes which include one or more application names listed in tab "Files" by completing the field "**Name**". The sequence of the specified files does not matter. If the same application names are also specified in the list of tab "Files" also their code integrity is checked. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button.
Link API statically to Application	The *CodeMeter Core API* is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here 263).
Size of encrypted Code (in %)	Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here 270).

7.4.1.5 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

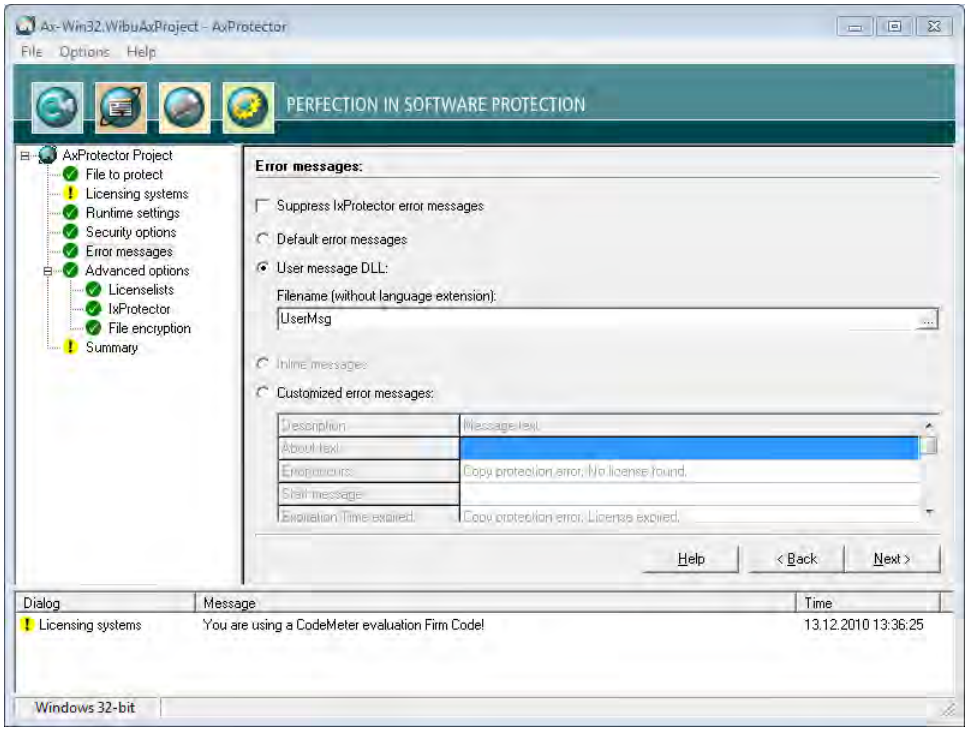
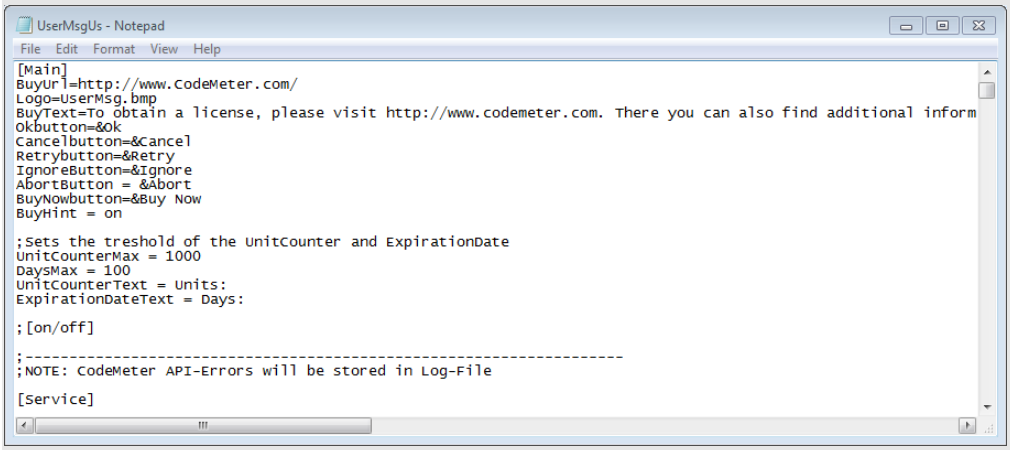


Figure 25: *AxProtector* - Windows "Error Messages"

Error Messages

Element	Description
Suppress IxProtector Error Messages	The output of *IxProtector* error messages is suppressed (commandline option see here[273]). If you do not activate this option, when using *IxProtector* errors, additional message windows are displayed along with the messages you program in the project.
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here[277]).
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text (commandline option see here[278]). The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.  Figure 26: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension. The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.
Inline Messages	Links for .NET projects, with an inline assembly which can also be configured by `*.ini` files.

```
[Main]
BuyUrl=http://www.CodeMeter.com/
Logo=UserMsg.bmp
BuyText=To obtain a license, please visit http://www.codemeter.com. There you can also find additional inform
Okbutton=&Ok
Cancelbutton=&Cancel
Retrybutton=&Retry
IgnoreButton=&Ignore
AbortButton = &Abort
BuyNowbutton=&Buy Now
BuyHint = on

;Sets the treshold of the UnitCounter and ExpirationDate
UnitCounterMax = 1000
DaysMax = 100
UnitCounterText = Units:
ExpirationDateText = Days:

;[on/off]

;-----------------------------------------------------------------
;NOTE: CodeMeter API-Errors will be stored in Log-File

[Service]
```

Element	Description
	This option is available for the encryption of .NET applications only.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.1.6 Advanced Options

This input window lets you set further options for the encryption using *IxProtector* and for the project type file encryption.

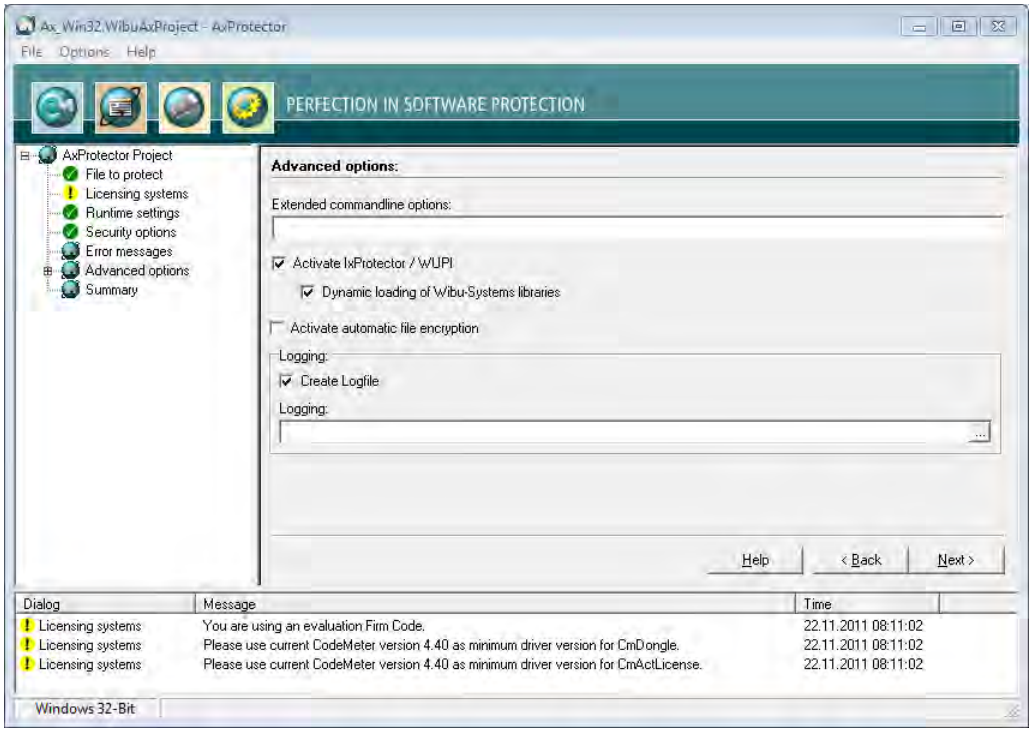


Figure 27: *AxProtector* - Windows "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (command option see here 274).
Dynamic loading of Wibu-Systems libraries	If activated, this checkbox results in a special, more time-intensive process. This when VB6 applications or dynamic loading of Wibu-Systems libraries are involved (command option see here 273)
Activate Automatic File Encryption	Activate this checkbox to trigger the automatic decryption of files by the *AxProtector* engine (command option see here 267). This option must be set if your encrypted application is later to be able to access the encrypted files.
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.4.1.6.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

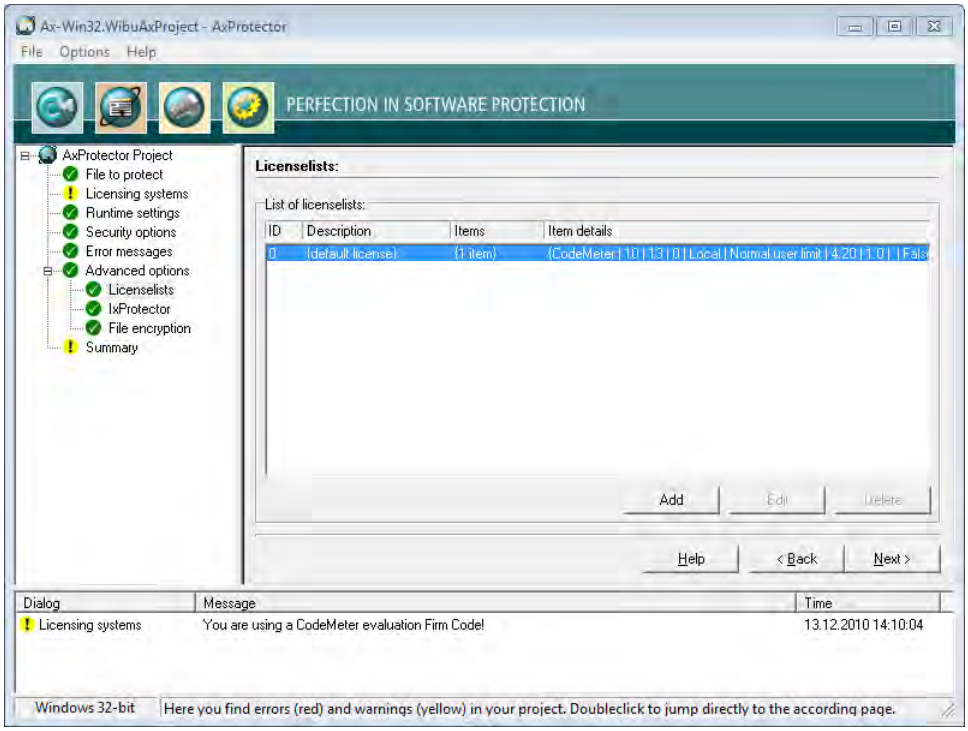


Figure 28: *AxProtector* - Windows "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **ID**s starting from **1**.
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.

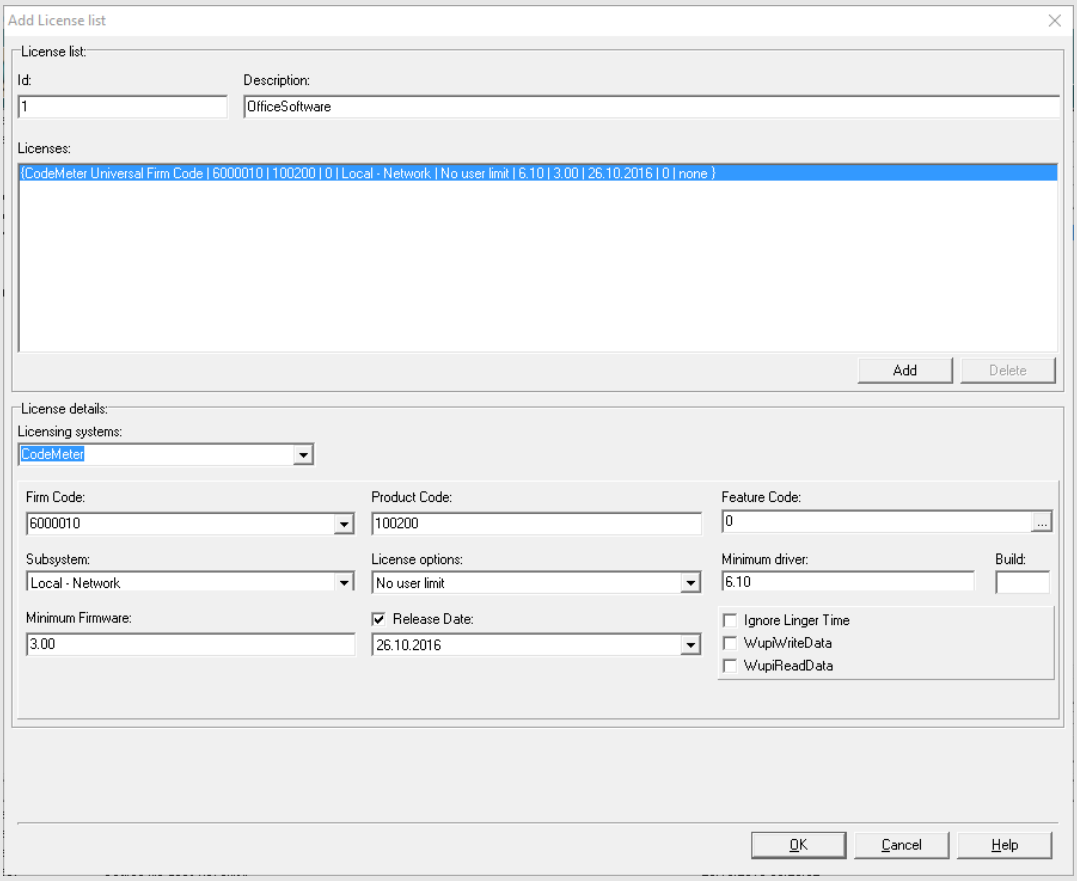
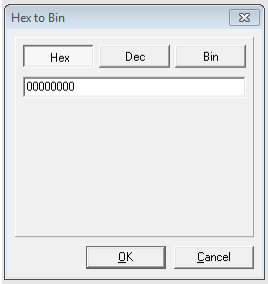
<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td></td><td>


Figure 29: AxProtector - Windows "Add License Lists"
</td></tr>
<tr><td>Licensing Systems</td><td>Selecting the desired licensing system to be applied:
<table>
<tr><th>Entry</th><th>Description</th></tr>
<tr><td>CodeMeter</td><td>Applying the licensing system CodeMeter.</td></tr>
<tr><td>IP Protection</td><td>Applying the licensing system IP Protection.
Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required.
Depending on the input file and the selected encryption options, AxProtector creates a key with which the application to be protected is encrypted.
With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption.
Commandline option see here 264 .
• Please note that after a decision for exclusive protection (IP Protection) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.</td></tr>
<tr><td>WibuKey</td><td>Applying the licensing system WibuKey.
For setting WibuKey options, see the separate "WibuKey Developer Guide".
• If you are switching from WibuKey to CodeMeter, please activate both licensing systems.
In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a CmDongle or a CmActLicense together with the protected application.</td></tr>
</table>
</td></tr>
<tr><td>Firm Code</td><td>Specify the Firm Code to be used for encrypting the software.
As a registered licensor, you will be issued your own unique Firm Code(s).
The following default settings exist:
<table>
<tr><th>Firm Code CodeMeter Software Development Kit (SDK)</th><th>Licensing system</th></tr>
<tr><td>6000010 Evaluation Universal Firm Code</td><td>CodeMeter Universal Firm Code</td></tr>
<tr><td>10 CmDongle Evaluation Firm Code</td><td>CmDongle</td></tr>
<tr><td>5010 CmActLicense Evaluation Firm Code</td><td>CmActLicense</td></tr>
</table>
Commandline option see here 264 .</td></tr>
<tr><td>Product Code</td><td>Enter the Product Code which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application.
Commandline option see here 264 .</td></tr>
</table>

Element	Description
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 30: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Firm Codes (licensing system)	Version
`6000010; >= 6.000.000` (*Universal Firm Code*)	6.10 This supports the License Transfer feature.
`10, 100.000-4.999.999` (*CmDongle*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.
`5010, 5.000.000-5.999.999` (*CmActLicense*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal

Element	Description
	Firm Codes (licensing system) / Version servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

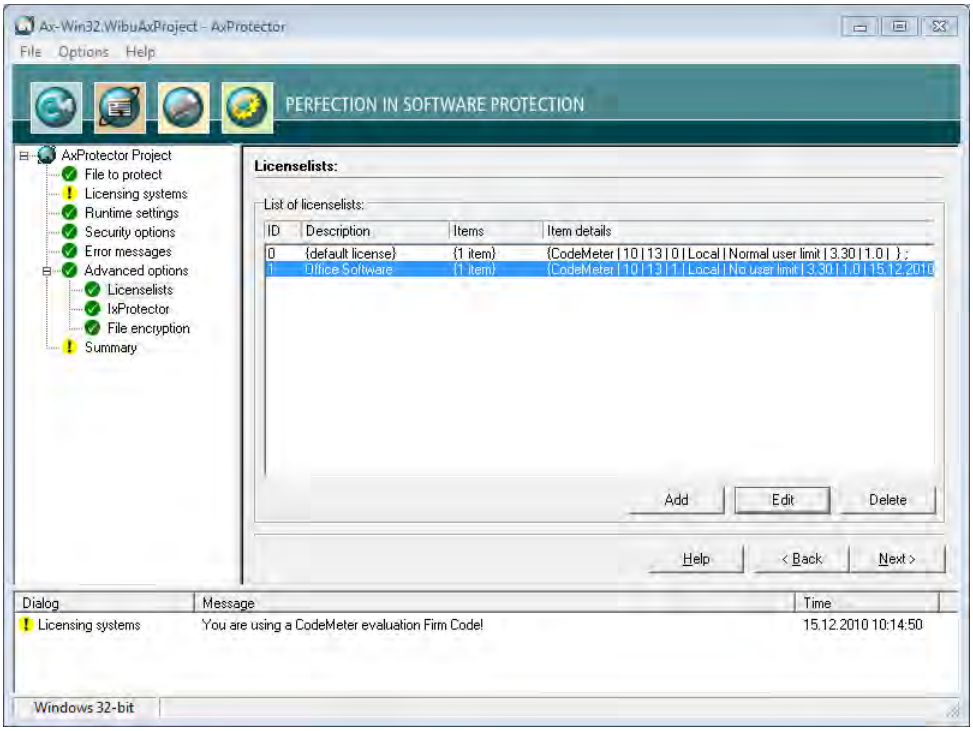


Figure 31: *AxProtector* - Windows "Completed License List"

7.4.1.6.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

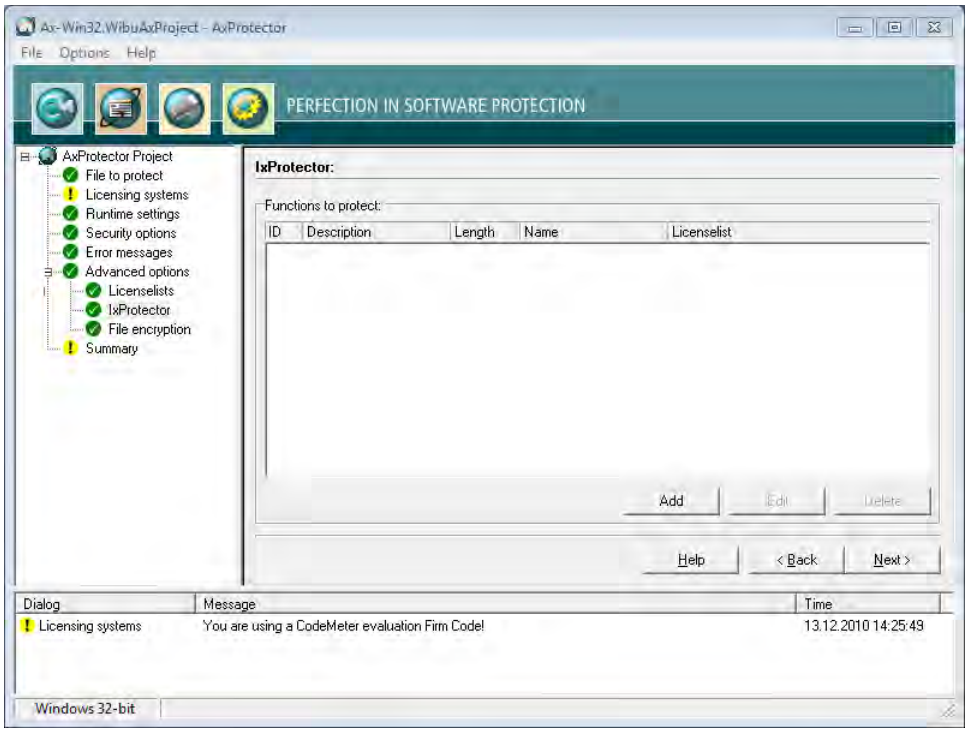


Figure 32: *AxProtector* - Windows *IxProtector* - "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: **1.** Click the **"Add"** button in the group "IxProtector Options".

Element	Description
	2. Define the function by completing the fields in the "Function" group.

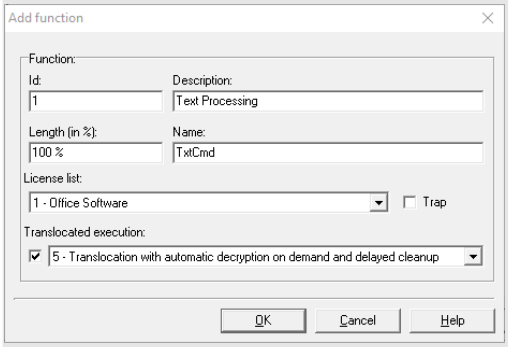


Figure 33: *AxProtector* - Windows *IxProtector* - "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291 .
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.
License List	Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.
Trap	Activates the trap function for the function.
Translocated execution	Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.

Option	Description
1	Translocation with automatic decrpytion on demand and cleanup.
2	Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).
5	Translocation with automatic decrpytion on demand and delayed cleanup. (Default)

Command line option see here 286 .

3. Click the **"OK"** button. The new functions are added to the function list.

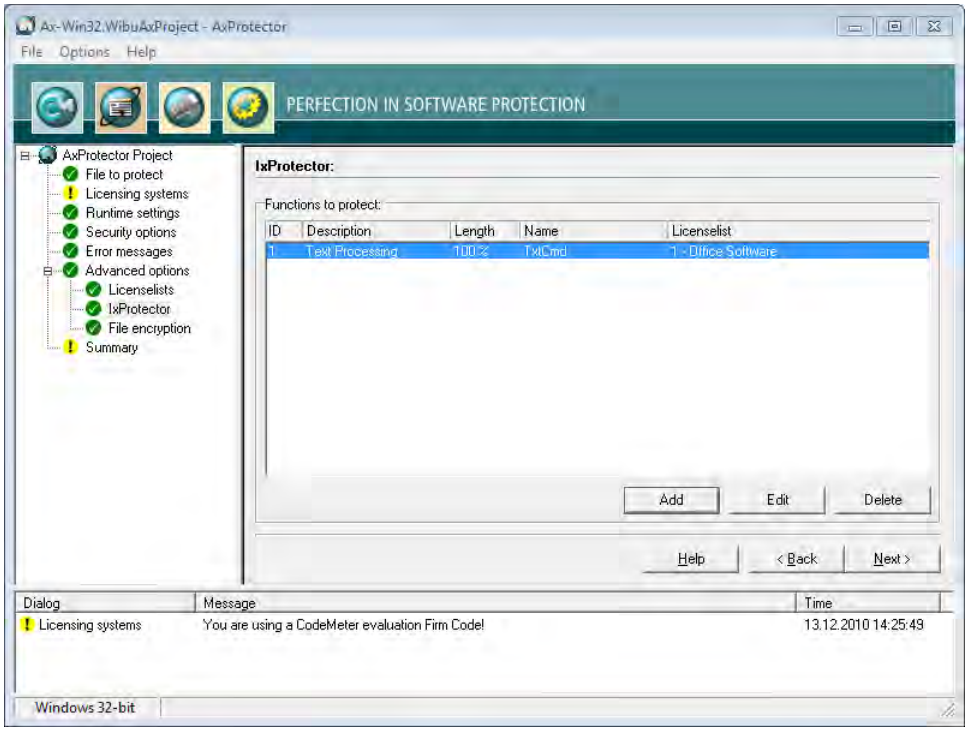


Figure 34: *AxProtector* - Windows *IxProtector* - "Completed Function List"

7.4.1.6.3 File Enryption

This menu item lets you define the rules on how an application accesses the encrypted files. In addition, you have the option to define those rules in a list for different file types. You can add as many file types as possible. For a file only one file type is required.

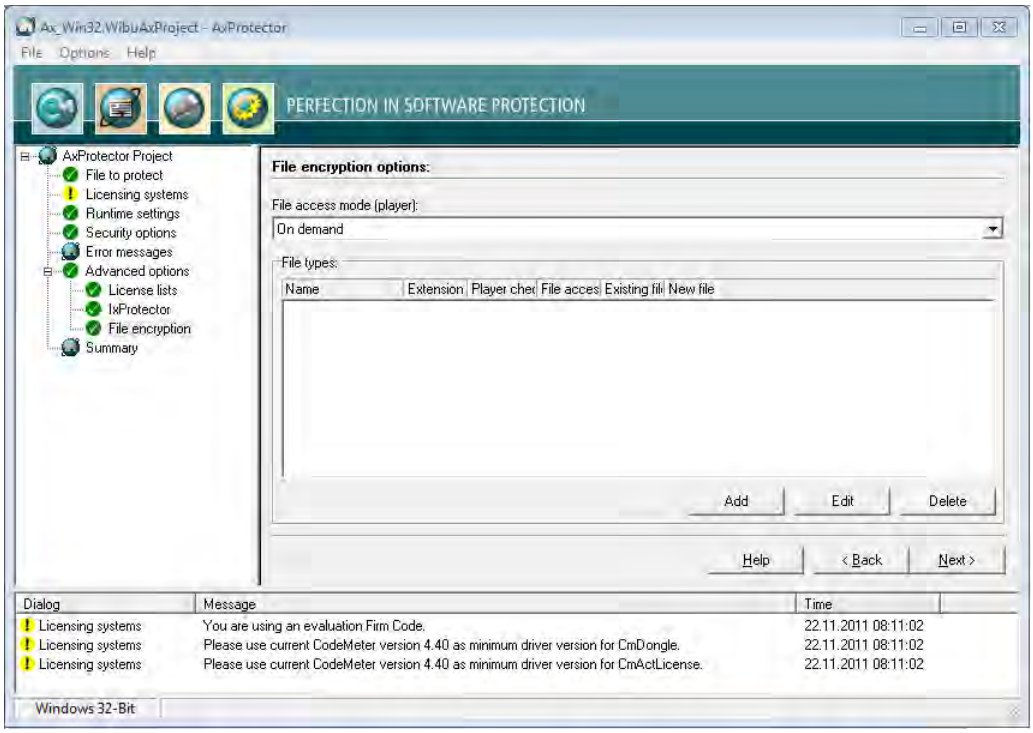


Figure 35: *AxProtector* - Windows "File Encryption"

Element	Description
Add File Type	**1.** Click on the **"Add"** button to add a new file type.

Element	Description

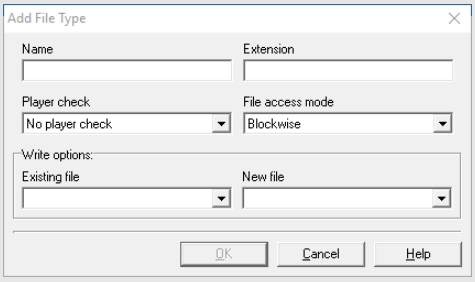


Figure 36: *AxProtector* - File Encryption "Add File Type"

2. Enter in the **"Name"** field a describing descriptive name for the file type. This name has no impact on the encryption.
3. Enter in the **"Extension"** field the file extension of the file type you create, e.g. txt for text files.
4. In the **"Player Check"** dropdown you define whether the license options of the accessing application (player) are checked when the encryption takes place.

License list	The player (accessing application) has to be encrypted using a license from this license list. For example, this allows you to define that a specific file type is accessed exclusively by the application you encrypted.
No player check	No check of the accessing application is performed.

5. In the **"File Access Mode"** dropdown define how the player is prepared for the access of protected files. This mode allows you to configure the memory required and the runtime behavior.

The selection of a suitable mode depends on the type of the player and the size of the file. For example, when working with video files you should select "Huge file mode (read only)". In the case of smaller files (configuration files) you may access several times, the mode "At once" is preferable.
Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.

On demand	The player reserves RAM space for the complete file to be read; but reads only the required part – strictly speaking all 4 Kbyte blocks are holding this part – and decrypts these blocks. For further accesses to the protected file, more required blocks are loaded (on demand) and decrypted. When the required part is located in blocks already loaded, the decrypted image in the memory is used. In this way, step-by-step the player builds up a complete memory image of the required file. This mode requires a lot of memory (the same size as the file to be loaded). However caching the decrypted data provides for good performance at runtime when accessing already decrypted blocks. This mode is available for read and write access.
At once	The player reserves RAM space for the complete file to be read; completely reads it, and completely decrypts it. Further accesses to the protected files, use the decrypted memory image. his mode requires a lot of memory (the same size as the file to be loaded). However, caching the decrypted data provides a good performance at runtime. Compared to the "on demand" mode, this mode requires more time for first access (the file is completely loaded and decrypted). The performance of each additional access is increased because the file resides completely in memory, in a decrypted form. This mode is available for read and write access.
Huge file mode	The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory. This mode requires no additional memory. Multiple accesses to the same data means that the data has to be read and decrypted each time. This mode is available for read access only.

6. In the group **"Write Options"** define the settings on how changes are saved.
Existing File
In this group you define the settings on how changes to an existing file are saved.

Original	Changes are allowed. Where the file was encrypted, it is re-encrypted. Unencrypted files are saved with no decryption.
No writing	Write actions are not allowed. Just read-only access is allowed.
License list	Changes are only encrypted using the license options defined in the selected license list.

New File
In this group you will define the settings on how new files are saved.

Plain	New files are only saved unencrypted.
No writing	New files cannot be saved. A new file is saved, however no data is saved to this file.
License List	New files are only encrypted using the license options defined in the selected license list.

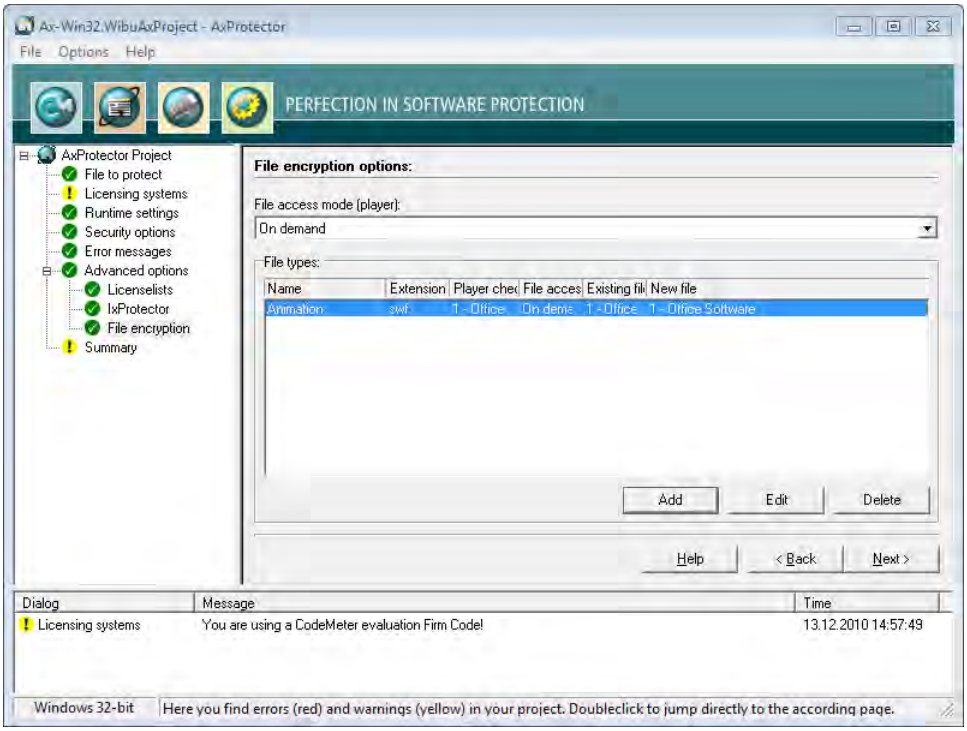


Figure 37: *AxProtector* - File Encryption "Completed Option list"

7.4.1.7 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

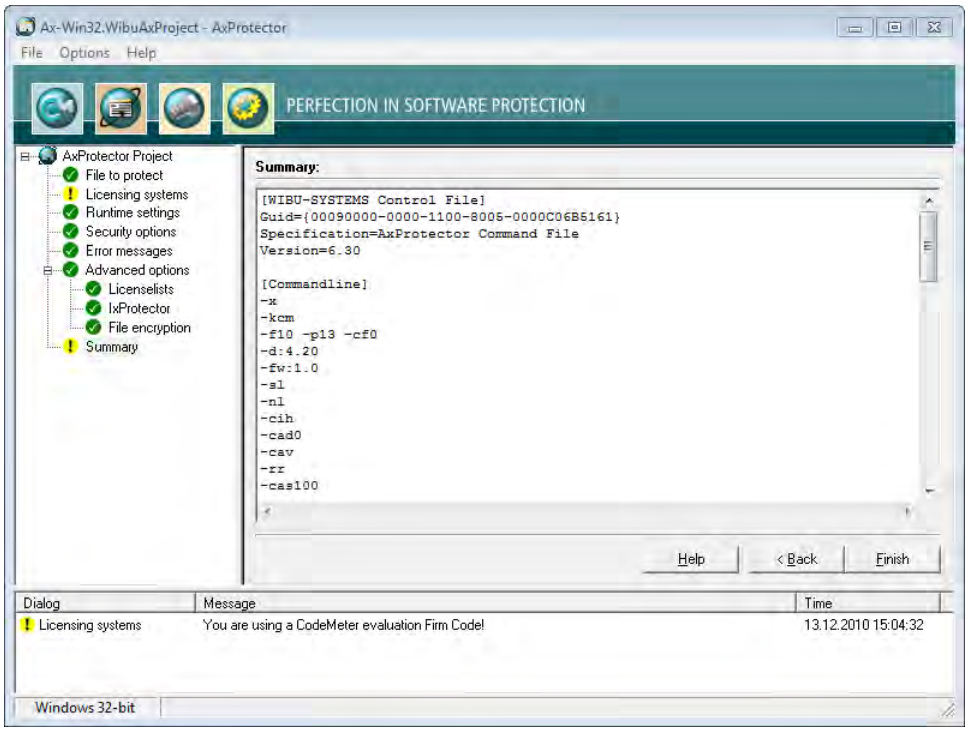


Figure 38: *AxProtector* - Windows "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.

Element	Description
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

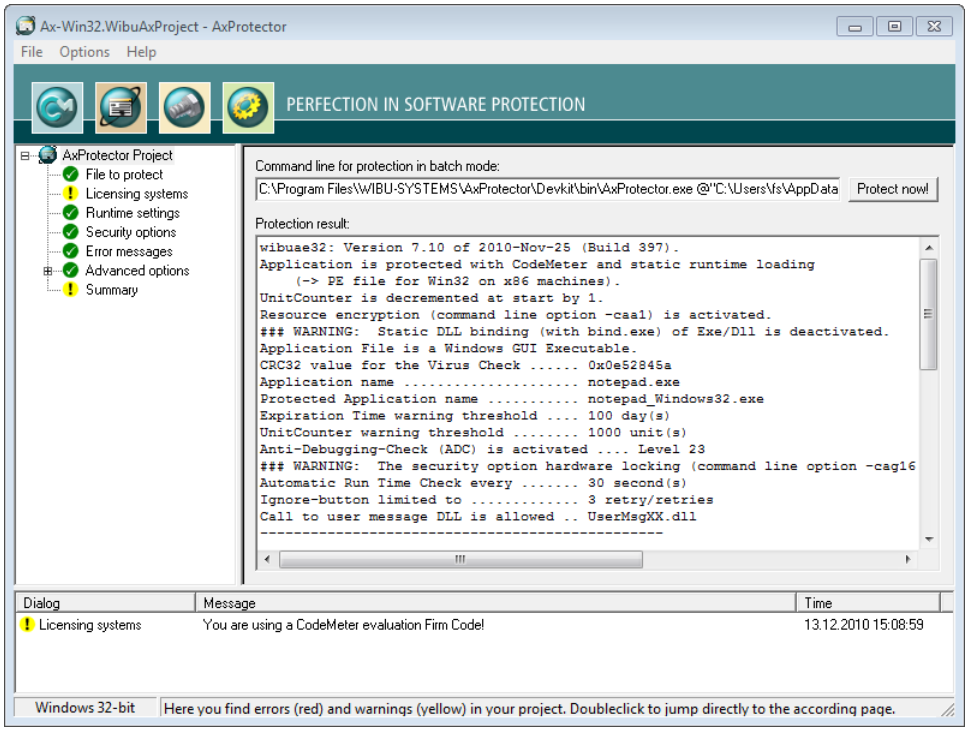


Figure 39: *AxProtector* - Windows "Encryption Result"

Element	Description
Protect now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.4.2 .NET Assembly

In principle, a .NET assembly is an open book to hackers: using capable tools, e.g. Reflector, disassembling of your code and thus reverse engineering is quite simple. In order to prevent unauthorized analysis or modification, your executable code should always be encrypted before delivery.

In the case, you wish to encrypt an already obfuscated program, please note that only a pure name obfuscation has been used.
If other changes have been applied to the assembly, eventually an executable but not editable assembly may the result. Then *AxProtector* is not able to interpret and encrypt this assembly.
Wibu-Systems recommends, first to encrypt the original assembly using *AxProtector* and following to apply the obfuscator.

Please note that after encryption with *AxProtector .NET* the output directory does not only contain the encrypted assembly. In addition, there are other files that are needed to use a protected assembly, e.g. the `WibuCmNet` resource libraries or the `wupi.net.dll`.

The *CodeMeter* API for .NET has been adapted to state-of-the-art cross-platform status, i.e. the *WibuCmNET.DLL* has been cross-platform-implemented in .NET Standard 2.0. This new .NET Standard library is to become the basis for all versions of .NET and provide required APIs to share .NET code across several platforms.
In order to implement this switch the package *NuGet* Package has been created integrating the newly created *WibuCmNET.DLL* file and language satellites.
A respective document how the Wibu-System ISV customer is able to use applications with the *WibuCmNET.DLL* for cross-platform support is available as knowledge base item in the separate CodeMeter Core API 298 help.

Encryption of C# code for Unity 3D and Mono

Unity 3D is a runtime and development environment for games based on Mono. Mono is an open source implementation of the Microsoft .NET framework.

A minimum runtime version of 5.16 is required to run an encrypted assembly under Mono.

A minimum version of the Unity Software Development Kit (SDK) of version 2018.3 is required.

Support is provided for:

a) .NET Framework 4.6.1 and higher: Windows

b) .NET Standard 2.0: Windows, Linux

For Mono and Unity, currently only one assembly can be loaded encrypted within a process.

Protected Unity / Mono applications on Linux currently only use the default port 22350 for *CodeMeter* license accesses. The use of other ports is not supported.

The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
.NET Assembly	AxProtector .NET	√	.NET commandline 263

Starting with Version 4.20c also the .NET 4.0 Framework is supported. The new commandline variant `AxProtectorNet4.exe` is able to handle .NET 4.0 assemblies. *AxProtector .NET 2.0* automatically starts *AxProtector .NET 4.0* on the attempt to encrypt an .NET 4.0 assembly.

How does it work?

AxProtector works as follows:

- Your assembly is disassembled by *AxProtector .NET.*
- Classes, methods and fields are extracted from the original assembly.
- A new assembly is created.
- Classes are created with the same names, methods and fields.
- The newly created methods, however, do not hold the original code but instead make calls to the *AxEngine*.
- The original code is encrypted by the license you select, and is appended to the data section.

At the first call of the encrypted method, the code inserted by *AxProtector .NET* calls the *AxEngine*. The *AxEngine* decrypts the original code stored in the data section, and calls the encrypted code. Because the original methods keep their original names, you are still able to call them from outside. Even the parameters (type and description) stay the same.

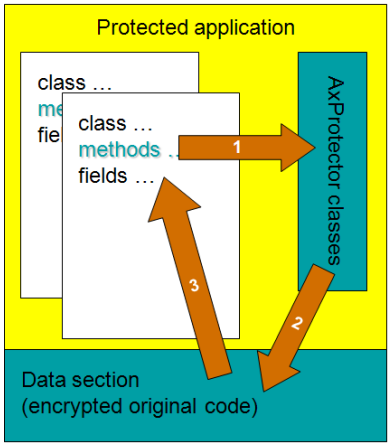


Figure 40: .NET encryption

However, disassembling the encrypted code is not possible.

You can define for yourself which methods are encrypted, and which locate unencrypted in the assembly. This you define optionally for a complete name space, a complete class, or a single method.

A definition at the method level overrules definitions at the class level. The same holds for the class and name space level.
At the same time, you determine whether encryption takes place using the default license, not at all, or separate license lists are used.

With the latter option you automatically implement modular software protection.

The following menu items are available in the navigation windows:

- File to protect 84
- Licensing Systems 84
- Runtime Settings 90
- Security Options 93
- Error Messages 95
- Advanced Options 97

- License Lists 97
- IxProtector 102
- Summary

7.4.2.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

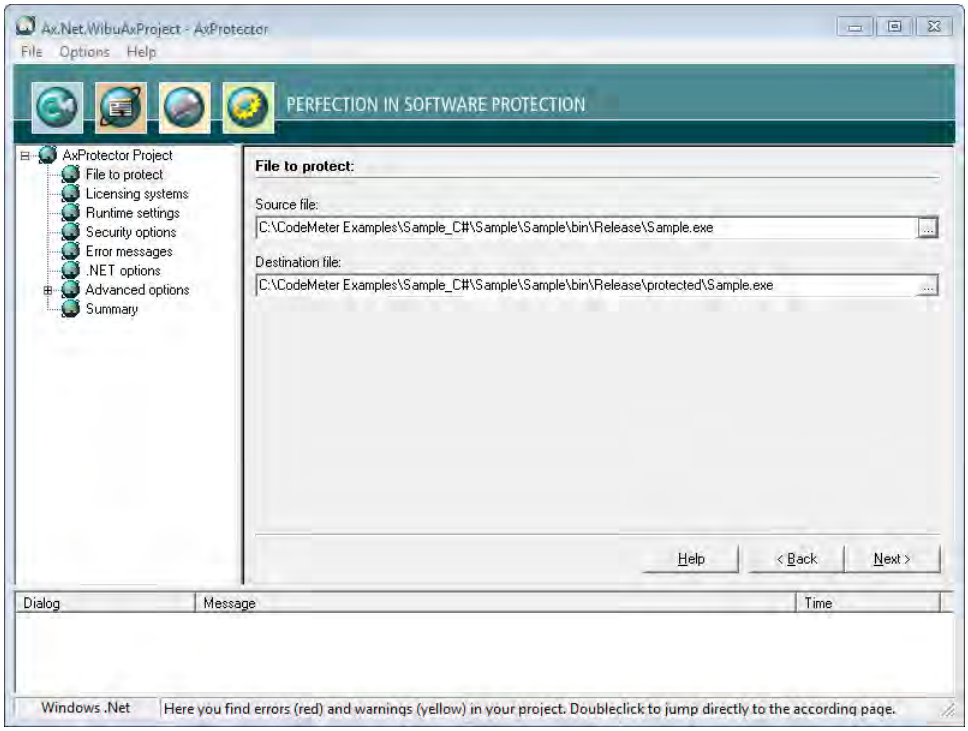


Figure 41: *AxProtector* .NET - "File to Protect"

File to Protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279 .

7.4.2.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

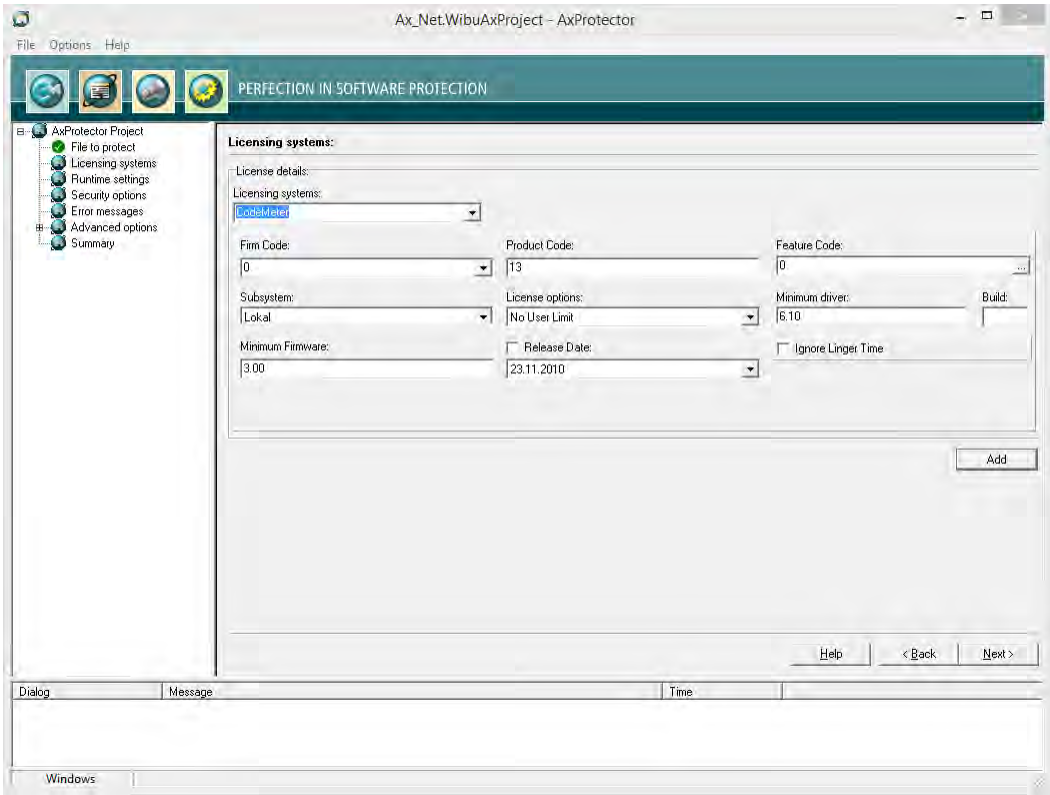
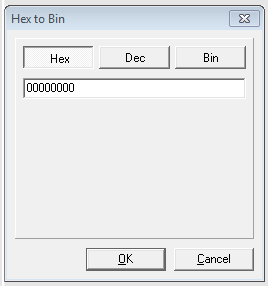


Figure 42: *AxProtector* .NET - "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	**Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: *Firm Code CodeMeter* Software Development Kit (SDK) — Licensing system `6000010` Evaluation *Universal Firm Code* — *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* — *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* — *CmActLicense* Commandline option see here 264 .
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .

Element	Description
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 43: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Firm Codes (licensing system)	Version
`6000010; >= 6.000.000` (*Universal Firm Code*)	6.10 This supports the License Transfer feature.
`10, 100.000-4.999.999` (*CmDongle Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description	
	Firm Codes (licensing system)	Version
	`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.
	Commandline option see here 264 .	
Build	Enter the Build number of the minimum driver version.	
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*	
Minimum Firmware	The following default settings exist:	
	Firm Codes (licensing system)	Version
	`6000010; >= 6.000.000` (*Universal Firm Code*)	3.00 This supports the License Transfer feature.
	`10, 100.000-4.999.999` (*CmDongle Firm Code*)	1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)	1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	Commandline option see here 265 .	
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55".	
	Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265 .	

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

7.4.2.2.1 Licensing Systems - Add licenses

Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description	
Licensing systems	Selecting the desired licensing system to be applied:	
	Entry	Description
	`CodeMeter`	Applying the licensing system *CodeMeter*.
	`IP Protection`	Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.
	`WibuKey`	Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.

Element	Description
	Entry: Description In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the ***Firm Code*** to be used for encrypting the software. As a registered licensor, you will be issued your own unique ***Firm Code***(s). The following default settings exist:

Firm Code CodeMeter Software Development Kit (SDK)	Licensing system
`6000010` Evaluation ***Universal Firm Code***	*CodeMeter*
`10` *CmDongle* Evaluation ***Firm Code***	*CmDongle*
`5010` *CmActLicense* Evaluation ***Firm Code***	*CmActLicense*

Commandline option see here 264 .

Product Code	Enter the ***Product Code*** which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the ***Feature Code*** which defines, for example, the encryption of different software versions.

By default, a ***Feature Code*** of 0 is set. This deactivates the use of the Product Item Option ***Feature Map***.Enter a 32-bit value to use the option.

Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.

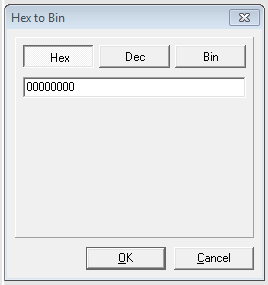


Figure 44: ***Feature Map*** Input

Commandline option see here 264 .

Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.

Element	Description
	Element / **Description** No user limit — Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264 .
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265 .
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265 .

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.2.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

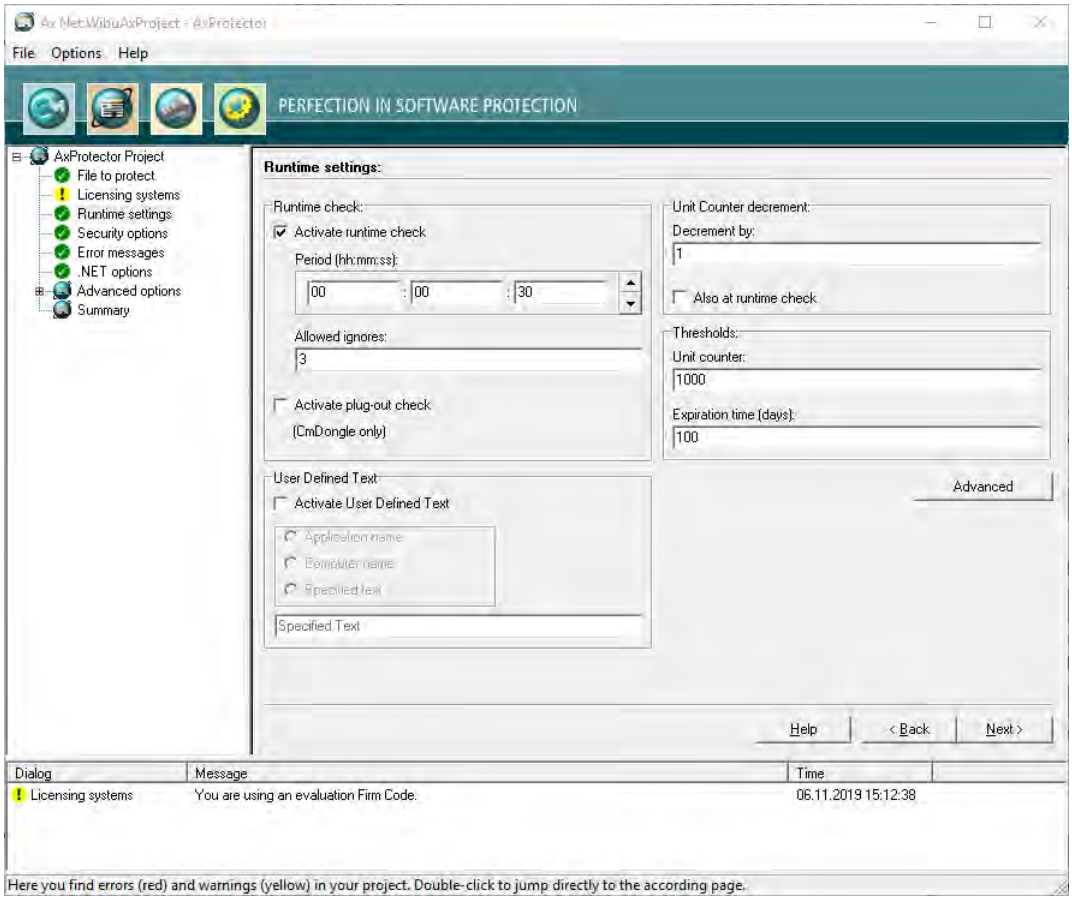


Figure 45: *AxProtector* .NET - "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: hours: minutes: seconds.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.
Activate Plug-out Check (only CmDongle)	This option closes the protected application when the *CmDongle* is removed while the application is running. Immediately, an error message is issued. This option is valid for *CmDongle* only. Commandline option see here 267 .

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the *Unit Counter* is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above, every 30 seconds (see the defined period) a set *Unit Counter* is decremented by a value of 1.
Also at Runtime Check	Decrements the *Unit Counter* also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime Check 90" group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here[95].

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here[277].
Expiration Time (days)	If the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here[277].

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here[279].

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. Element: Application name — uses the application name. Element: Computer name — uses the computer name. Element: Specified text — uses the specified text in the field of the same name.

7.4.2.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

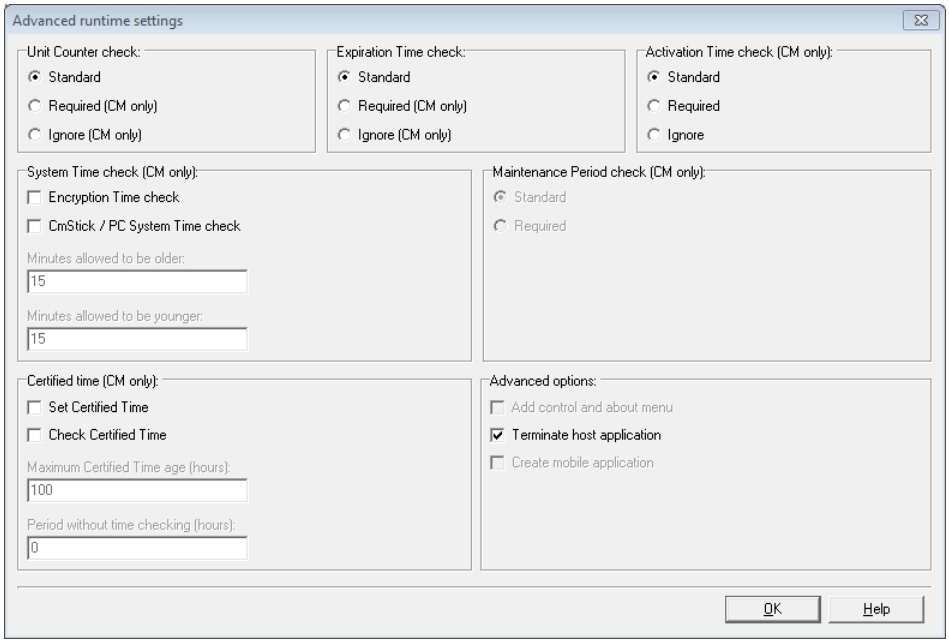


Figure 46: *AxProtector* .NET - "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignore
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here[276]).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches 0 (null) the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts when no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, when the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time 357 is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, when the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 (commandline option see here 276).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems 84.

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only if a *Maintenance Period* exists. This corresponds to the default setting, even if on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmContainer* is connected with the computer or activated. When the *CmContainer* is connected or activated, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *Certified Time*. On updating the *certified time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here 270).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here 357 ..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmContainer*. The *certified time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check" you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time.*
Period without time checking (hours)	Specifies the period (in hours) if no check of the *Certified Time* certificate is taking place. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here 267).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only if the *CmContainer* System Time is newer than the encryption time.

Element	Description
	Requires at least *CodeMeter*® 4.10.
CmContainer / PC System Time check	When activated these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor, the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

Advanced options

This group allows to set further options.

Element	Description
Terminate host application	When no valid license is found, in the case of protected DLL application files the calling `*.exe` is terminated (commandline option see here 277).
Create mobile application	[not yet implemented]

7.4.2.4 Security Options

This input window lets you select from different mechanisms and methods for protecting your application. You are able to scale the degree of security for yourself, for example, the search intensity for debugger or if a *CmContainer* is locked.

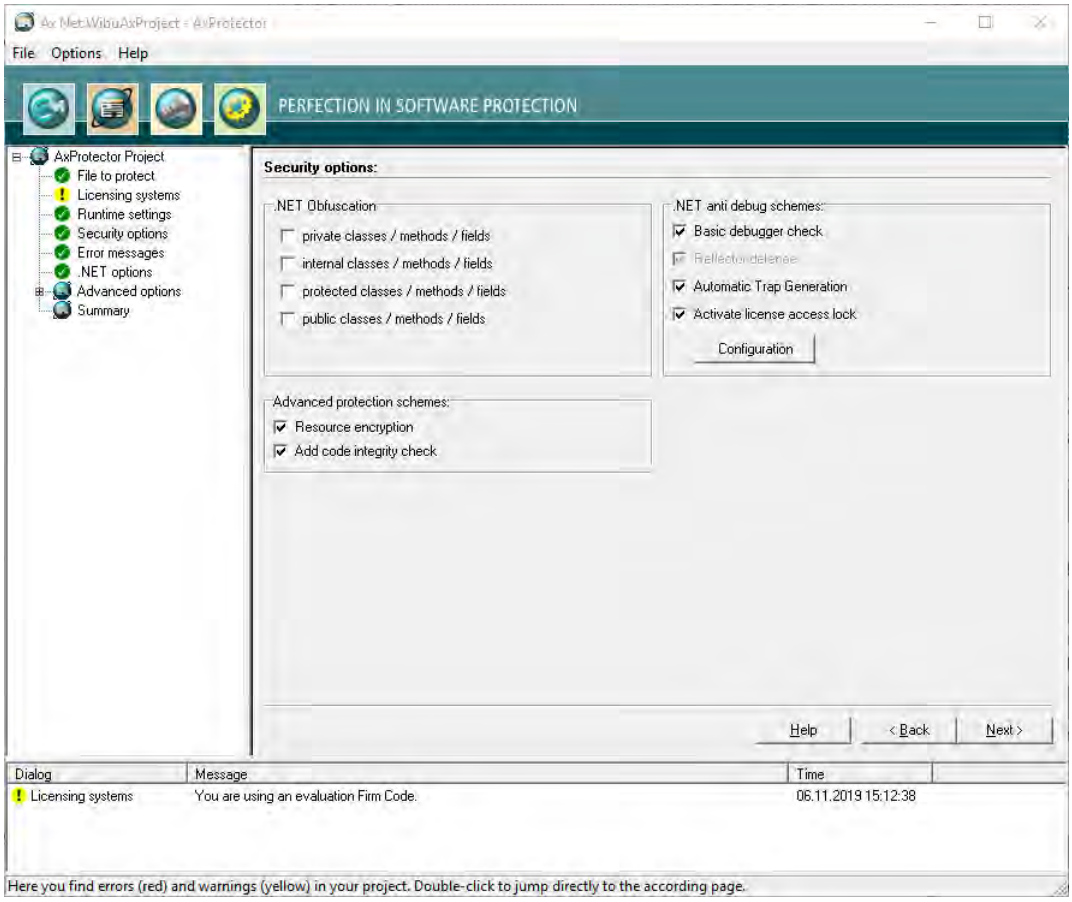


Figure 47: *AxProtector* .NET - "Security Options"

.NET Obfuscation

The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information (commandline option see here 274). Elements comprise classes, methods, and fields.

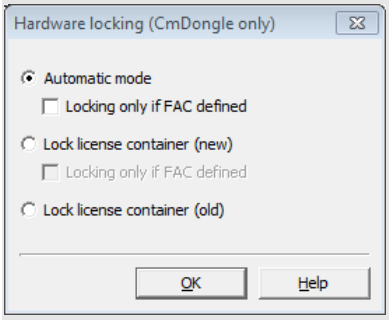
Element	Description
private classes / methods / fields	obfuscates private elements
internal classes / methods / fields	obfuscates internal elements
protected classes / methods / fields	obfuscates protected elements

Element	Description
public classes / methods / fields	obfuscates public elements

Anti-Debugging Schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here 267).

Element	Description
Basic Debugger Check	The 'Basic Debugger Check', checks to see if a debugger is attached to your application. If a debugger is found, your application will not be started or exited.
Reflector defence	For protected .NET assemblies automatically a reflector defence is activated preventing decompiling.
Automatic Trap Generation	Automatically inserts hacker traps into the protected assembly (commandline option see here 279).
Activate license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the **"Configuration"** button. This button is activated only for *CodeMeter*.
Configuration	If the option **"Activate license access lock"** is activated, you are able to define further settings in the dialog which opens by clicking the **"Configuration"** button: Depending on the Firmware used this dialog allows to define separate locking scenarios (for more detailed information see separate CodeMeter Developer Guide, section "Advanced CodeMeter Features | Locking a CmContainer").

Locking Scenario	Description
immediate locking	is performed starting with Firmware Version 1.14 as soon as a debugger is detected.
prepared locking	is performed by checking the *Firm Access Counter* (FAC). The *Firm Access Counter* locates 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`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked. The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.  Figure 48: *AxProtector* -.NET "Security Options - Hardware Locking"

The following settings are available:

Option	Description
"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher, the *Firm Item* is immediately locked. For compatibility reasons this represents the default setting.
"Automatic Mode" activated and "Locking only if FAC defined" activated	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher and a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (new)" activated and "Locking only if FAC defined" not activated	This option requires a Firmware version 1.14 or higher. The *Firm Item* is immediately locked. Seen from a security point of view this is the recommended option. This, however, requires that all *CmContainer* in the field must have a Firmware Version 1.14 an higher.
"Lock License Container (new)" and "Locking only if FAC defined" activated	This option requires a Firmware version 1.14 or higher. If a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.

Element	Description	
	Option	Description
	"Lock License Container (old)" activated	Independent from the Firmware version, if a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. This holds for all Firmware versions. If 'prepared locking' is programmed, the *Firm Access Counter* is decremented by a value of `1`.

Advanced protection schemes

The advanced protection schemes deeply intervene into your application. In some cases, this may mean that some single mechanisms will not work due to compatibility reasons (commandline options see here 266).

Element	Description
Resource encryption	Also encrypts the .NET resources of your protected application. After the start of your application, the resources located in the PC memory and are decrypted "on demand".
Add code integrity check	The protected application is checked for code integrity using asymmetric authentication asymmetric authentication 47 mechanisms, if you check this box (commandline options see here 270). On code integrity check first a check sum (hash value) of the application is created and signed with the private key of the Individual Software Vendor (ISV). The hash value and the signature are added to the application. The recalculation and the integrity check of the hash value and thus of the application is performed at runtime check using the public key located in the software (AxEngine). Alternatively to the default private key you can also apply the commandline option -sig 266 to use an entry of a *Hidden* or *Secret Data* field to define another private key. Moreover, the code integrity check may also cover several executable files / libraries. Then each file is able to check all other files for integrity. Each file then requires the public key of the ISV: The hash value of the files to be checked then is recalculated and compared to the hash value signed with the private key.

7.4.2.5 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used or whether you use default error message windows.

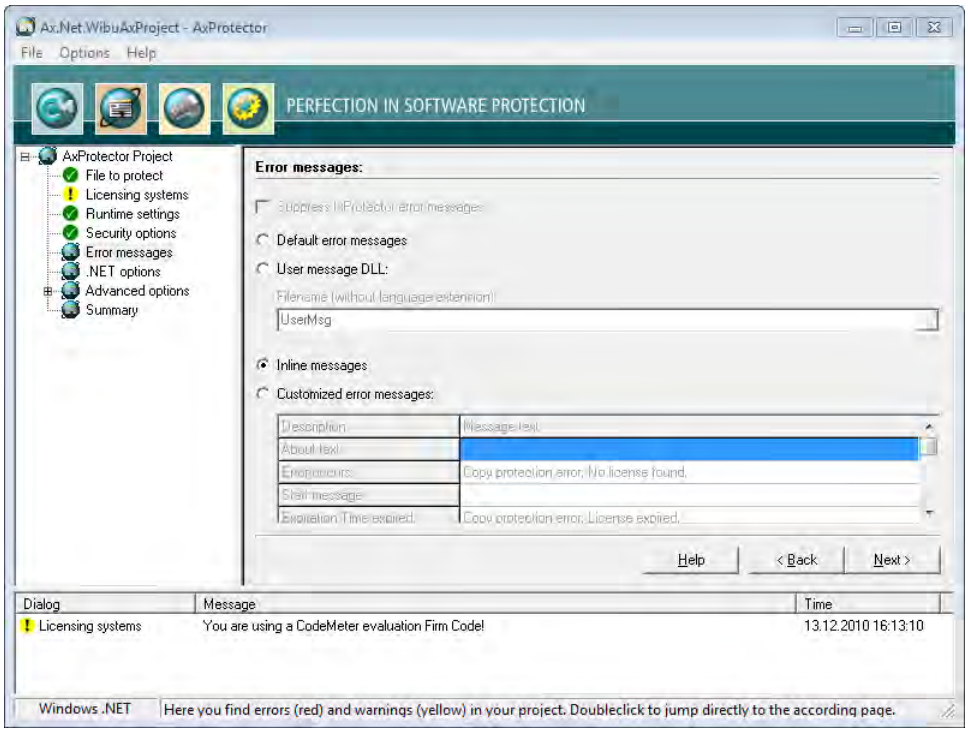


Figure 49: *AxProtector* .NET "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
User Message DLL	The ability to use the User Message DLL is activated. **File name (without Language Extension)** Enter the file name without specifying path and language file extension.

Element	Description
	Either you program an own User Message DLL and place it in the same directory as your protected application, or you use the Wibu-Systems sample User Message for .NET (`%CodeMeter_Samples%Software Protection\C#\UserMessage`) and place it in the same directory as your protected application.
Inline Messages	Links for .NET projects, with an inline assembly, can also be configured by `*.ini` files (commandline option see here 278). When using Inline UserMessages the logging is saved to the directory "`%CommonApplicationData%`". When you want to specify another path specify the parameter `LogPath<Path>` in the `*.ini` file.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.2.6 .NET Options

This page allows you to specify further .NET settings.

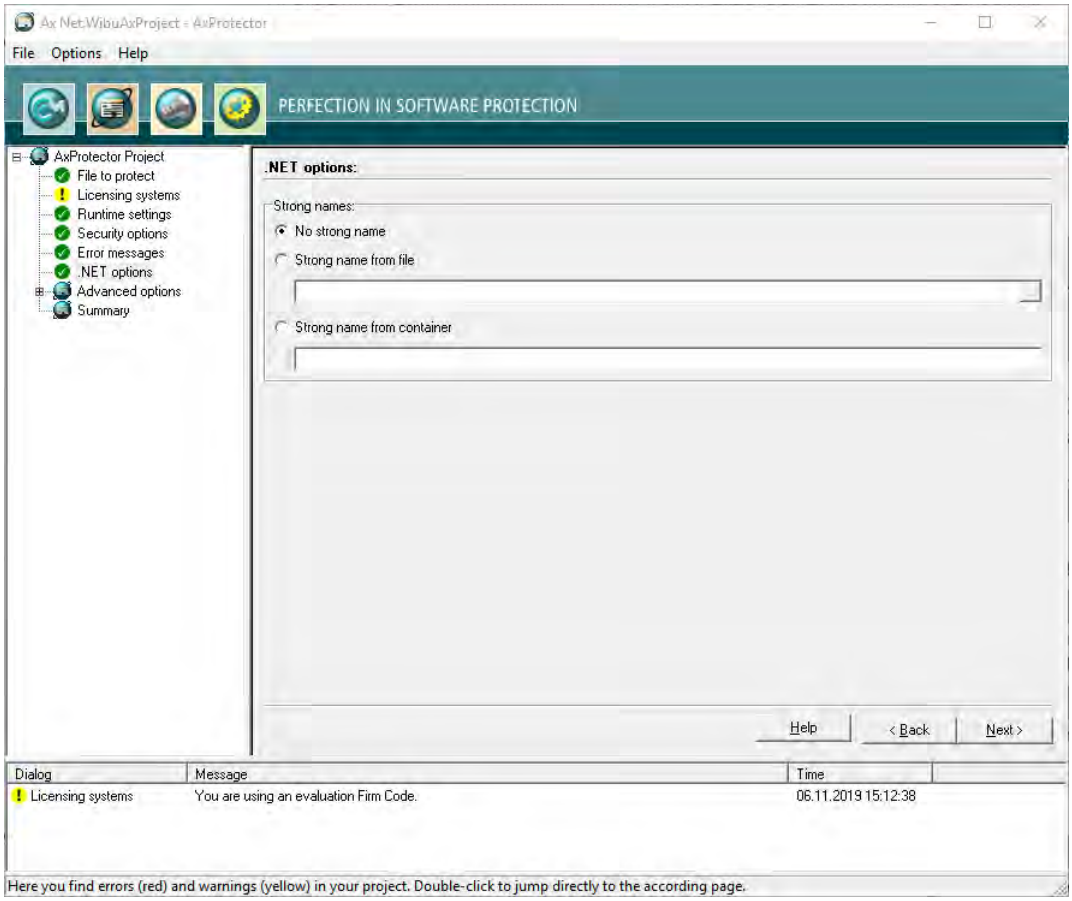


Figure 50: *AxProtector* .NET - ".NET Options"

.NET Options

Here you are able to specify whether your assembly is signed by *AxProtector*.

Element	Description
No Strong Name	Activate this checkbox to not sign your assembly.
Strong Name from File	Activate this checkbox to use a source file to sign the program class. Then specify a file holding the key pair to generate a strong name (commandline options see here 279).
Strong Name from Container	Activate this checkbox to use a container file to sign the program class (commandline options see here 279).

7.4.2.7 Advanced Options

This input window lets you set further options for the encryption using *IxProtector*.

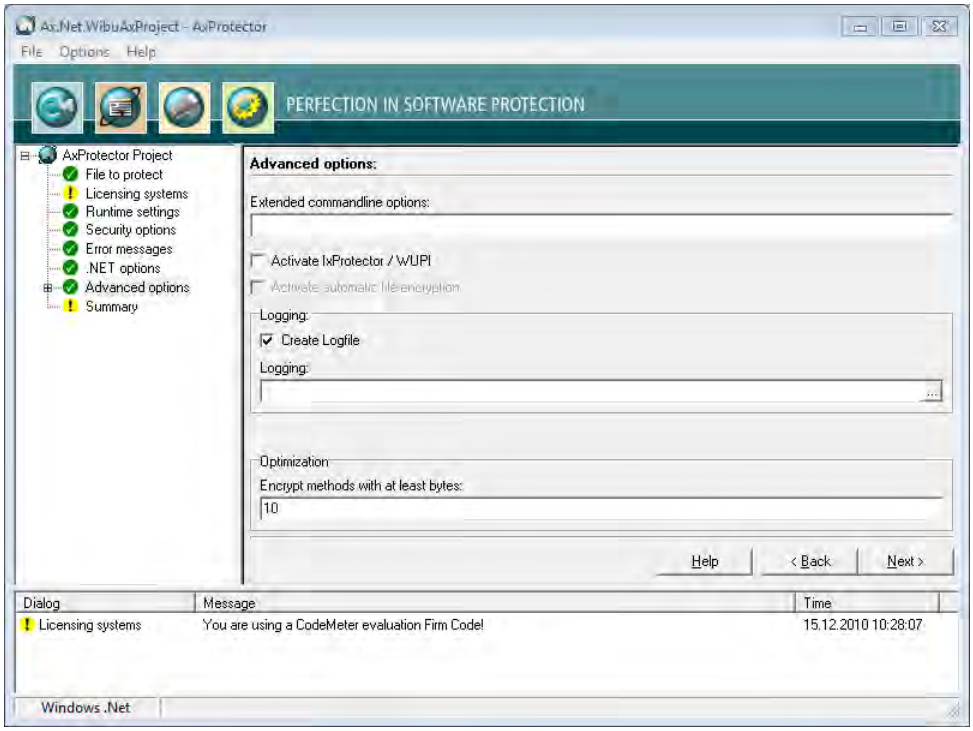


Figure 51: *AxProtector* .NET - "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (commandline option see here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector.*
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.
Optimization	For an optimized performance specify here the minimum size for assemblies to be encrypted. The default setting is 10 bytes. This way you are able to exclude methods from encryption which are smaller than the number of bytes you specify here. By setting a value of 0 this feature is deactivated. Commandline option see here 275.

7.4.2.7.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

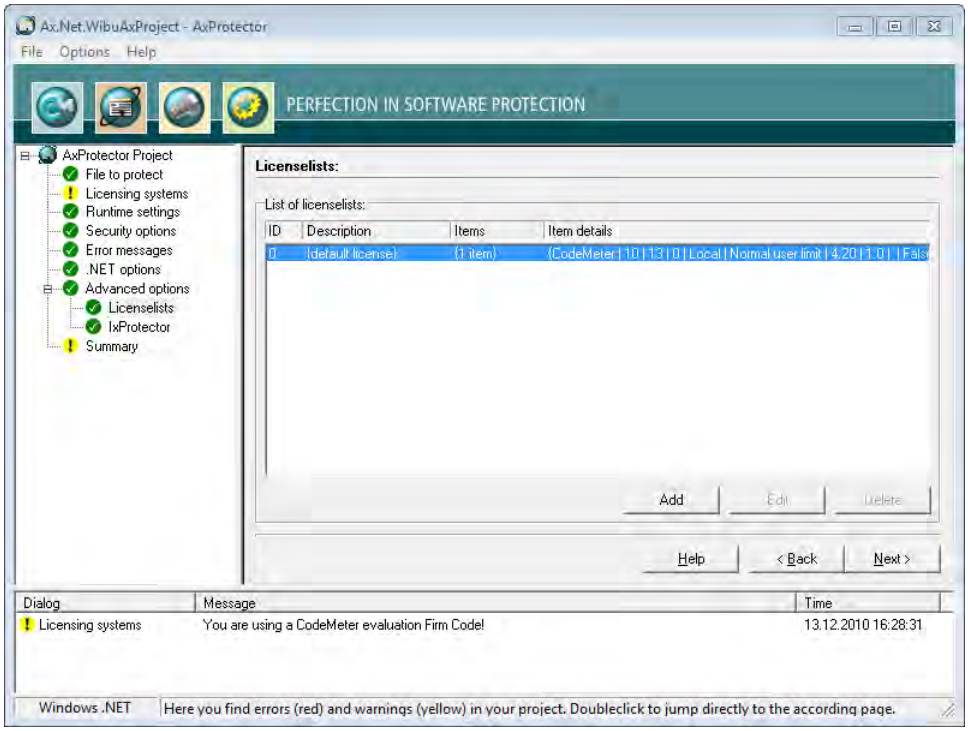
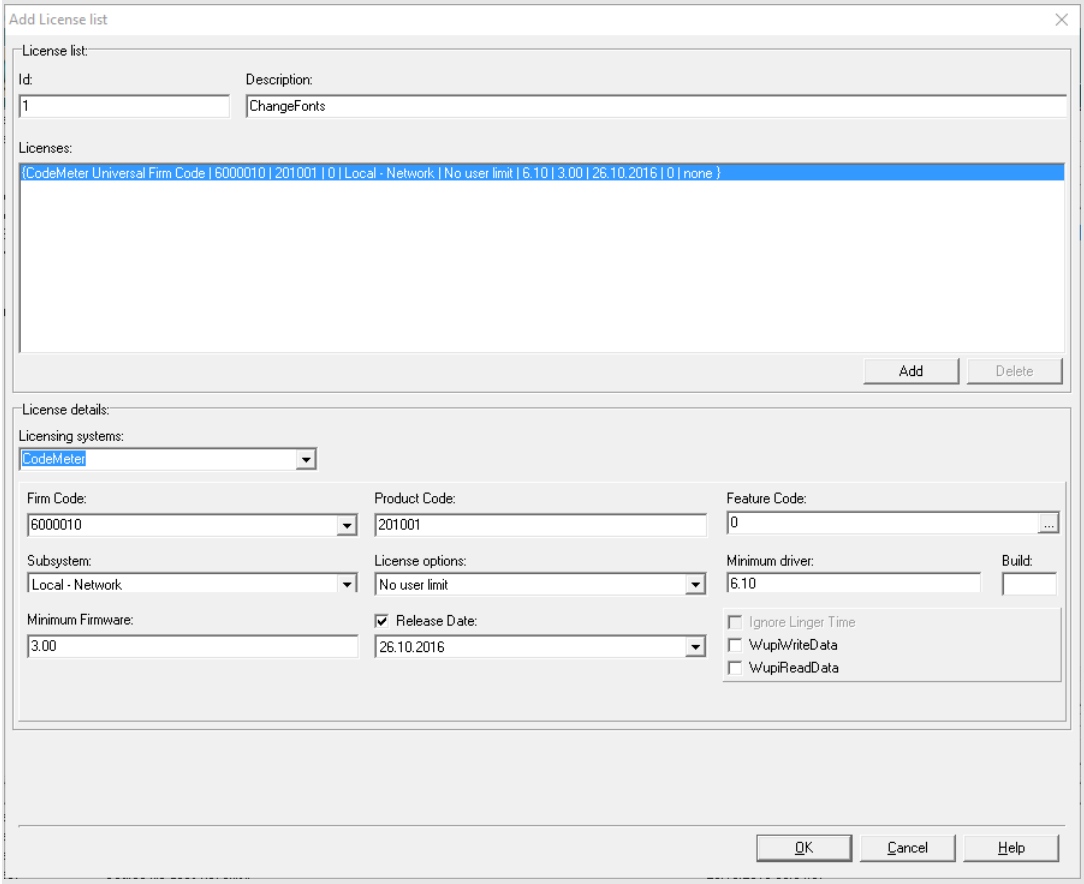


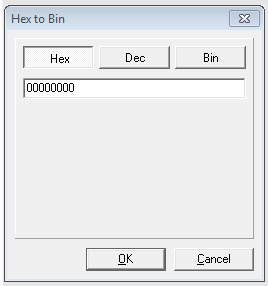
Figure 52: *AxProtector* .NET - "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 53: *AxProtector* .NET - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) / **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle `5010` *CmActLicense* Evaluation *Firm Code* — CmActLicense Commandline option see here 264.

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 54: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description
	Firm Codes (licensing system) / Version `5010, 5.000.000-5.999.999` (*CmActLicense*) / 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

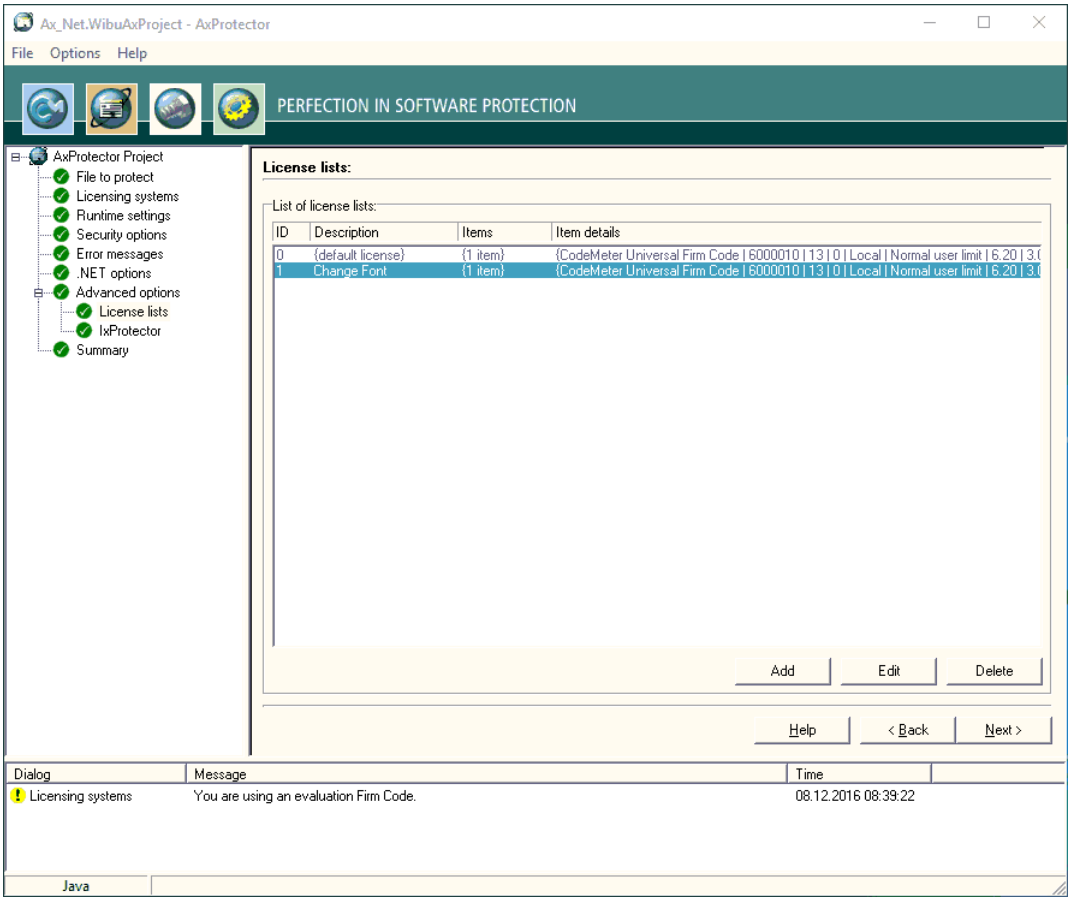


Figure 55: *AxProtector* .NET - "Completed License Lists"

7.4.2.7.2 IxProtector

Using this menu item allows you to separately define single encryption types for single assembly elements.

In the case you activated the checkbox **"IxProtector"** in the menu item **"Advanced options"** the source assembly is loaded and displayed in a tree view making available all name spaces, classes, and modules.

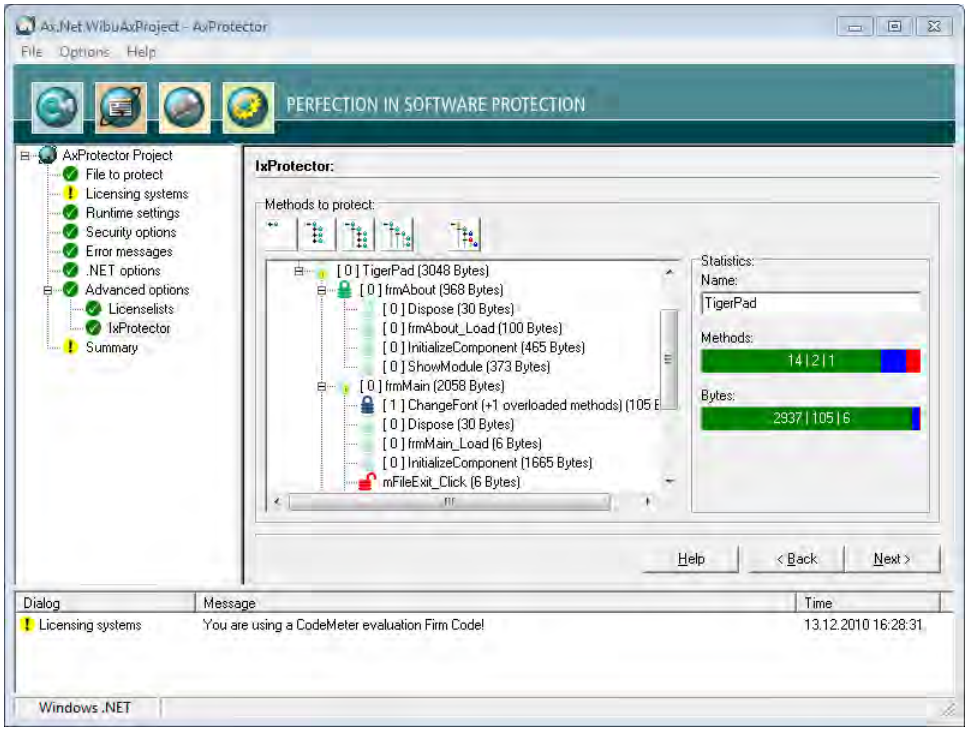


Figure 56: *AxProtector* .NET - "IxProtector"

Click the different buttons in the upper "IxProtector" area to select from different assembly views.

Buttons	Description
	Closes all assembly levels of the tree structure.
	Expands the name space level of the assembly.
	Expands the class level of the assembly.
	Expands the method level of the assembly.
	Expands all parent levels of the assembly. In this view see all levels where modifications have been made.

The area "Statistics" on the right shows you more encryption details depending on the selection you have made for the tree view.

Element	Description
Name	This field refers to the name of the element you have marked in the tree view.
Methods	Using different colors the bar 'Methods' shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted methods for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.
Bytes	Using different colors the bar 'Bytes' also shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted bytes for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.

You also have the option to separately assign the protection technologies *AxProtector* and *IxProtector* to single assembly elements, or exclude single elements from encrypting. To assign a protection technology by using the secondary menu, please proceed as follows:

1. In the left tree view, select the favored assembly element (name space, class, or method).
2. Click the right mouse button.
   The secondary menu opens.
3. Assign the favored encryption types by using symbols.

The License List IDs you are prompted are automatically transferred from the entries you added to the license list.

Symbol	Description
	Excludes the selected element from encryption.
	Encrypts the selected element using *AxProtector* (License List ID with a value of 0, i.e. default license).
	Encrypts the selected element using *IxProtector* (License List ID with a value unequal to 0, i.e. according to existing license list entries).
	This icon marks methods that are excluded from encryption due to the size of the method. The threshold can be set on the page 'Advanced Options' in the area optimizing

The modifications you made instantly display in the left area.

7.4.2.8 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

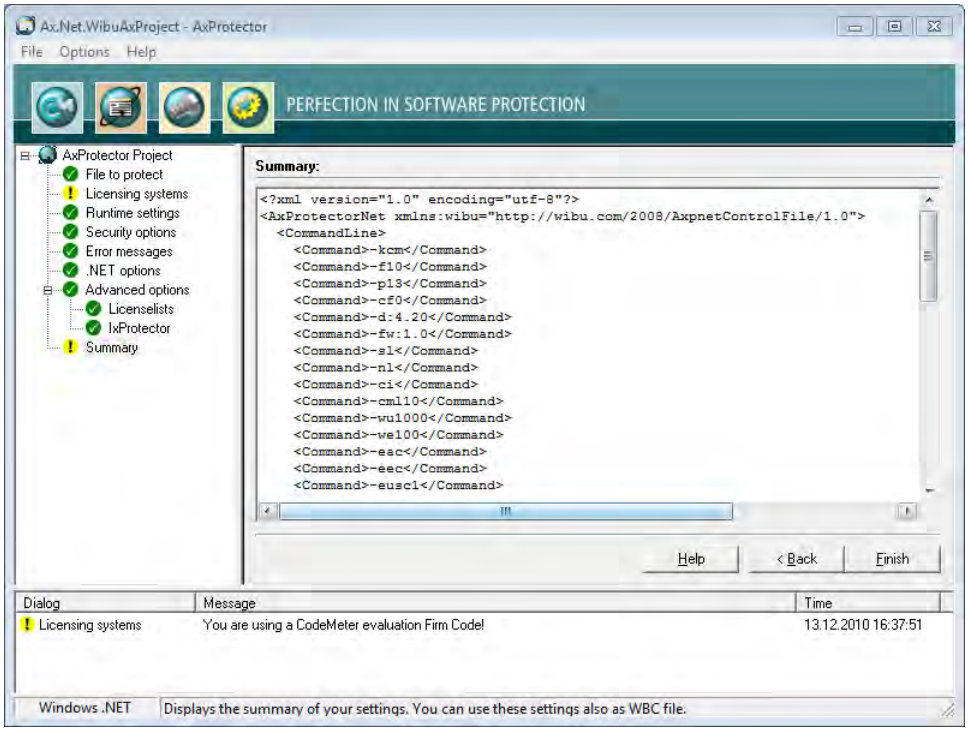


Figure 57: *AxProtector* .NET "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

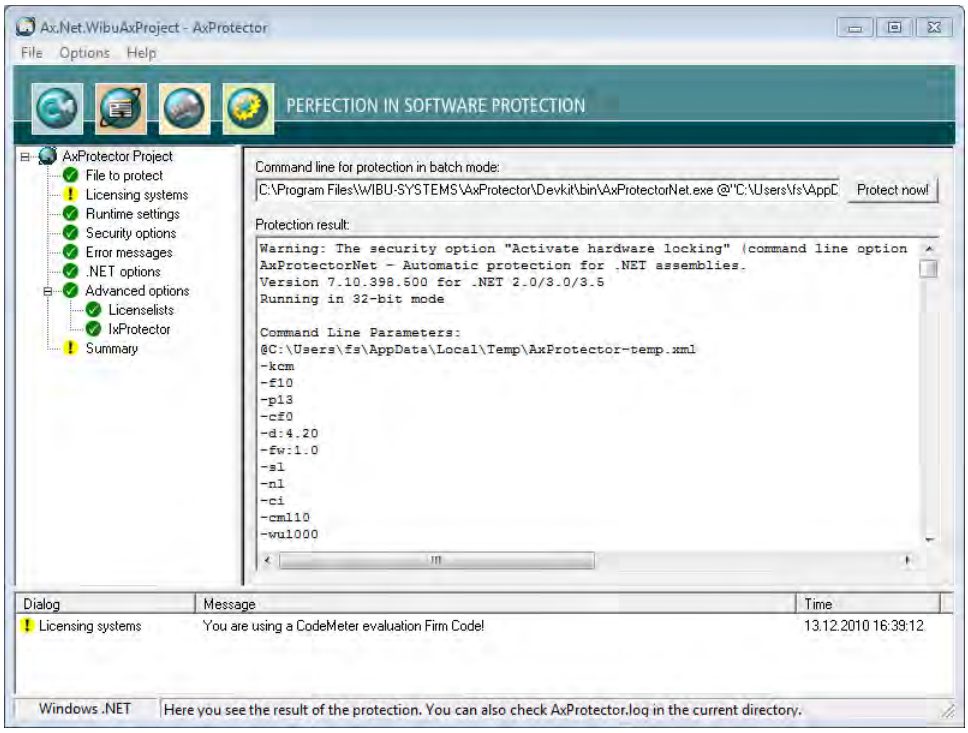


Figure 58: *AxProtector* .NET - "Encryption Result"

Element	Description
Protect now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.4.3 .NET Standard 2.0 Assembly

AxProtector supports the new framework .NET Core 2.0. The framework also implements '.NET Standard 2.0' as technical specification for programming interfaces (APIs) allowing to exchange program code among different implementations of .NET.
This results in the following changes for *AxProtector* in different areas:

Please note, that *AxProtector* .NET Standard currently runs on Windows only!

Commandline

Alternatively, *AxProtector* .NET Standard can be started by command line (directory: `C:\Program Files (x86)\WIBU-SYSTEMS\AxProtector\Devkit\bin\netstandard2.0`). The calling parameter match the common `AxProtectorNet` parameter.

On configuring please note the following:
One of the two parameter '`-ui`' for inline user messages or '`-um`' for messages is mandatory.
Otherwise the following error message is issued: "Using AxProtectorNet for Netstandard requires "-ui" or "-um" parameter to be set."

On encrypting please note the following:
For logical reasons the source directory of the application to be encrypted is to be the "publish" directory.
At encryption several files are transferred to the target directory "`/protected`":
- the encrypted own application, e.g. "`Protectee.dll`"
- all required assembly files and the files in the subdirectories the *AxEngine* requires.

Delivery / Shipping

However, the encrypted application in the directory "`protected`" is not immediately executable because other dependcies to external DLLs may exist or eventually the file "`Protectee.runtimeconfig.json`" has not been modified and thus not been copied from the source to the target directory.

Please make sure that the files locating in the target directory are copied to the publish directory, that then holds all required data for the executable product.

The *CodeMeter* API for .NET has been adapted to state-of-the-art cross-platform status, i.e. the *WibuCmNET.DLL* has been cross-platform-implemented in .NET Standard 2.0. This new .NET Standard library is to become the basis for all versions of .NET and provide required APIs to share .NET code across several platforms.
In order to implement this switch the package *NuGet* Package has been created integrating the newly created *WibuCmNET.DLL* file and language satellites.
A respective document how the Wibu-System ISV customer is able to use applications with the *WibuCmNET.DLL* for cross-platform support is available as knowledge base item in the separate CodeMeter Core API[296] help.

The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
.NET Standard Assembly	AxProtector .NET Standard	√	.NET commandline[263] to be found in directory: `C:\Program Files (x86)\WIBU-SYSTEMS\AxProtector\Devkit\bin\netstandard2.0`

How does it work?

AxProtector works as follows:

- Your assembly is disassembled by *AxProtector* .NET Standard.
- Classes, methods and fields are extracted from the original assembly.
- A new assembly is created.
- Classes are created with the same names, methods and fields.
- The newly created methods, however, do not hold the original code but instead make calls to the *AxEngine*.
- The original code is encrypted by the license you select, and is appended to the data section.

At the first call of the encrypted method, the code inserted by *AxProtector* .NET Standard calls the *AxEngine*. The *AxEngine* decrypts the original code stored in the data section, and calls the encrypted code. Because the original methods keep their original names, you are still able to call them from outside. Even the parameters (type and description) stay the same.

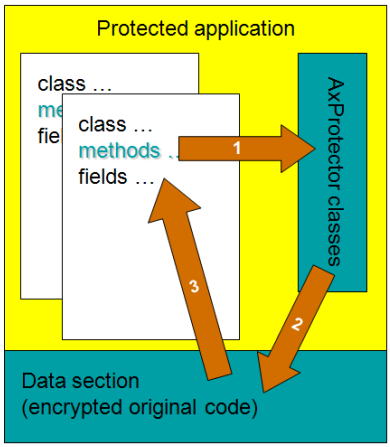


Figure 59: .NET Standard encryption

However, disassembling the encrypted code is not possible.

You can define for yourself which methods are encrypted, and which locate unencrypted in the assembly. This you define optionally for a complete name space, a complete class, or a single method.

A definition at the method level overrules definitions at the class level. The same holds for the class and name space level.
At the same time, you determine whether encryption takes place using the default license, not at all, or separate license lists are used.

With the latter option you automatically implement modular software protection.

The following menu items are available in the navigation windows:

- File to protect 107
- Licensing Systems 107
- Runtime Settings 113
- Security Options 116
- Error Messages 118
- Advanced Options 120
    - License Lists 120
    - IxProtector 125
- Summary

7.4.3.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

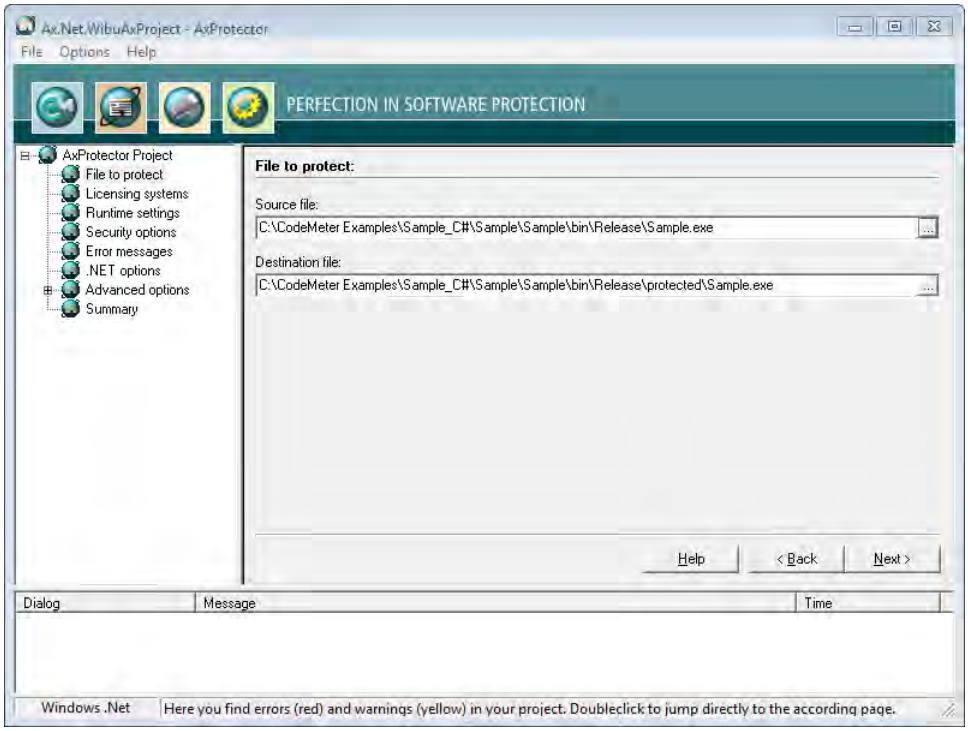


Figure 60: *AxProtector* .NET Standard - "File to Protect"

File to Protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.4.3.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

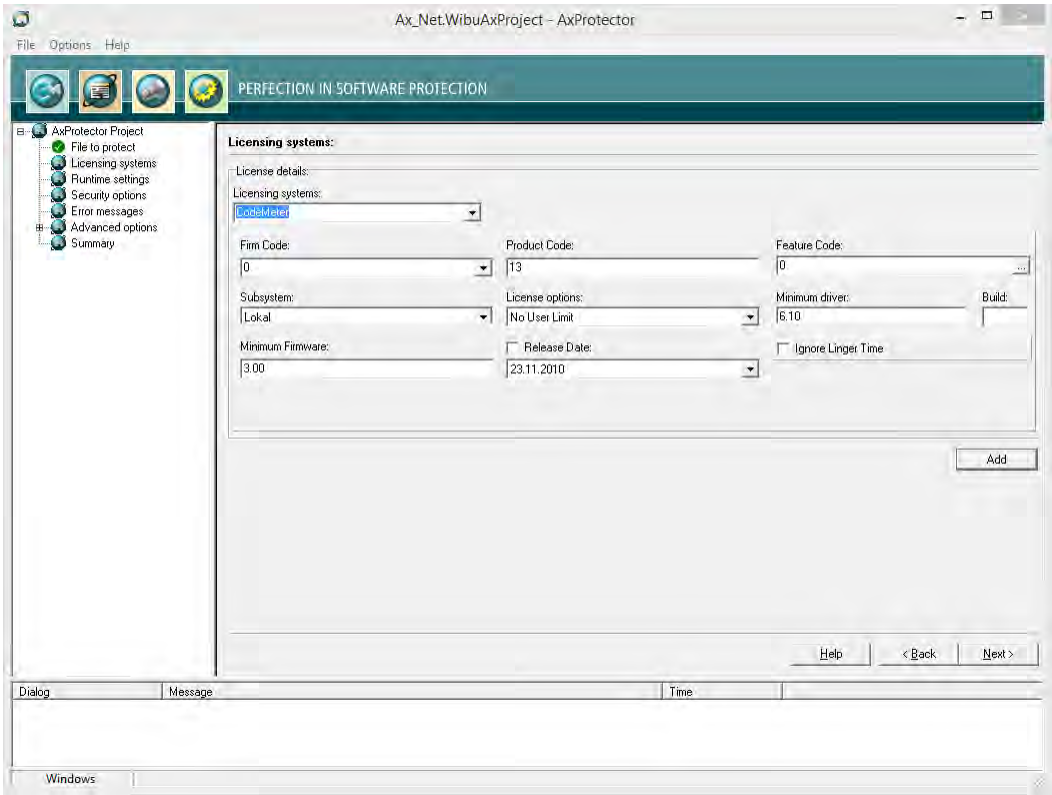
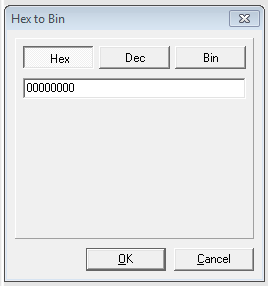


Figure 61: *AxProtector* .NET Standard - "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	**Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) / **Licensing system** `6000010` Evaluation *Universal Firm Code* / *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* / *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* / *CmActLicense* Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.

Element	Description
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 62: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Firm Codes (licensing system)	Version
`6000010; >= 6.000.000` (*Universal Firm Code*)	6.10 This supports the License Transfer feature.
`10, 100.000-4.999.999` (*CmDongle Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description	
	Firm Codes (licensing system)	Version
	`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.
	Commandline option see here 264 .	
Build	Enter the Build number of the minimum driver version.	
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*	
Minimum Firmware	The following default settings exist:	
	Firm Codes (licensing system)	Version
	`6000010; >= 6.000.000` (*Universal Firm Code*)	3.00 This supports the License Transfer feature.
	`10, 100.000-4.999.999` (*CmDongle Firm Code*)	1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)	1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	Commandline option see here 265 .	
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265 .	

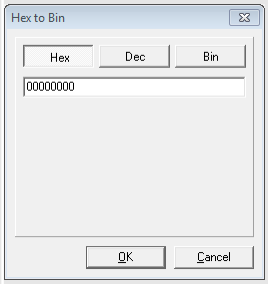
If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

7.4.3.2.1 Licensing Systems - Add licenses

Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description	
Licensing systems	Selecting the desired licensing system to be applied:	
	Entry	Description
	`CodeMeter`	Applying the licensing system *CodeMeter*.
	`IP Protection`	Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.
	`WibuKey`	Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Element	Description
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: `6000010` Evaluation *Universal Firm Code* — *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* — *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* — *CmActLicense* Commandline option see here 264 .
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 63: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** — **Description** Local — This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network — This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network — This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local — This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** — **Description** Normal user limit — Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share — Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode — Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode — Here a protected application can be started only once on a PC. No user limit — Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Element	Description
	<table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>6.10 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr></table>
	Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	The following default settings exist:
	<table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>3.00 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr></table>
	Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.3.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

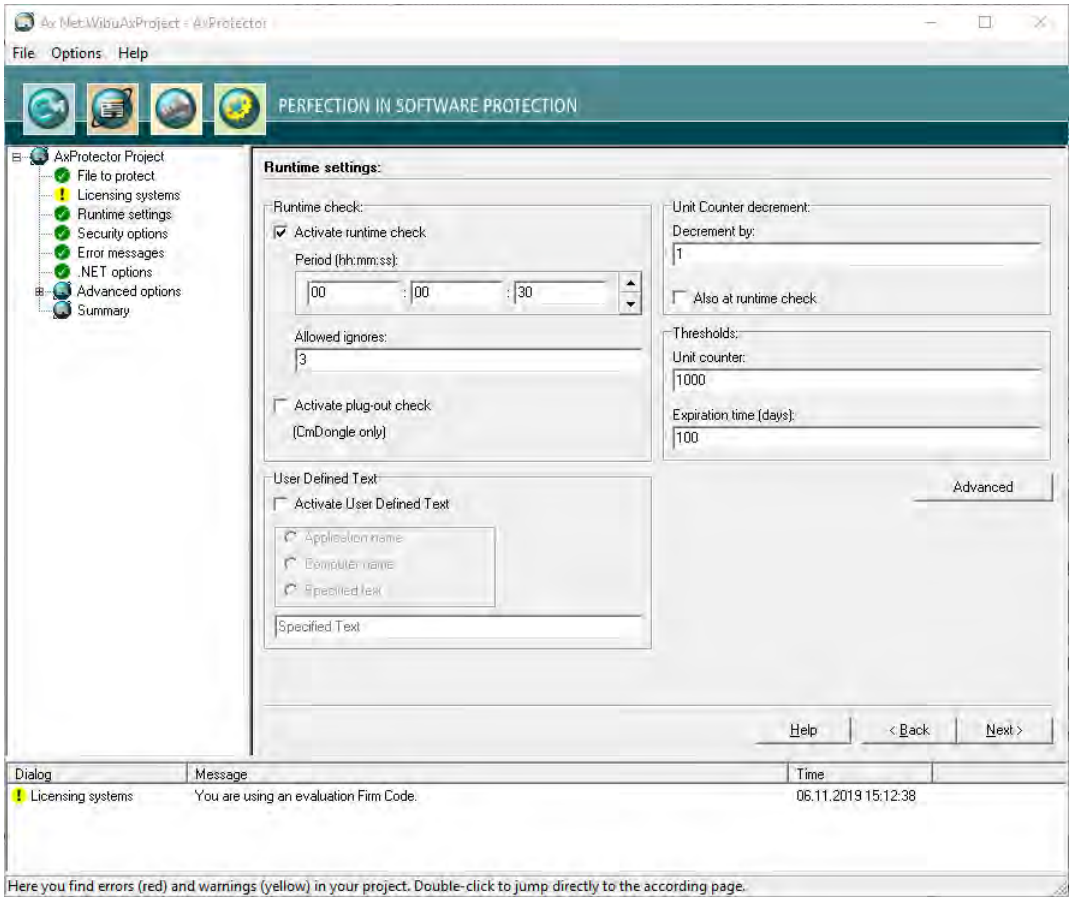


Figure 64: *AxProtector* .NET Standard - "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: hours: minutes: seconds.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.
Activate Plug-out Check (only CmDongle)	This option closes the protected application when the *CmDongle* is removed while the application is running. Immediately, an error message is issued. This option is valid for *CmDongle* only. Commandline option see here 267 .

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the *Unit Counter* is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above, every 30 seconds (see the defined period) a set *Unit Counter* is decremented by a value of 1.
Also at Runtime Check	Decrements the *Unit Counter* also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime Check 90" group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here 95.

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here 277.
Expiration Time (days)	If the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here 277.

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here 279.

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. Element / Description Application name — uses the application name. Computer name — uses the computer name. Specified text — uses the specified text in the field of the same name.

7.4.3.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

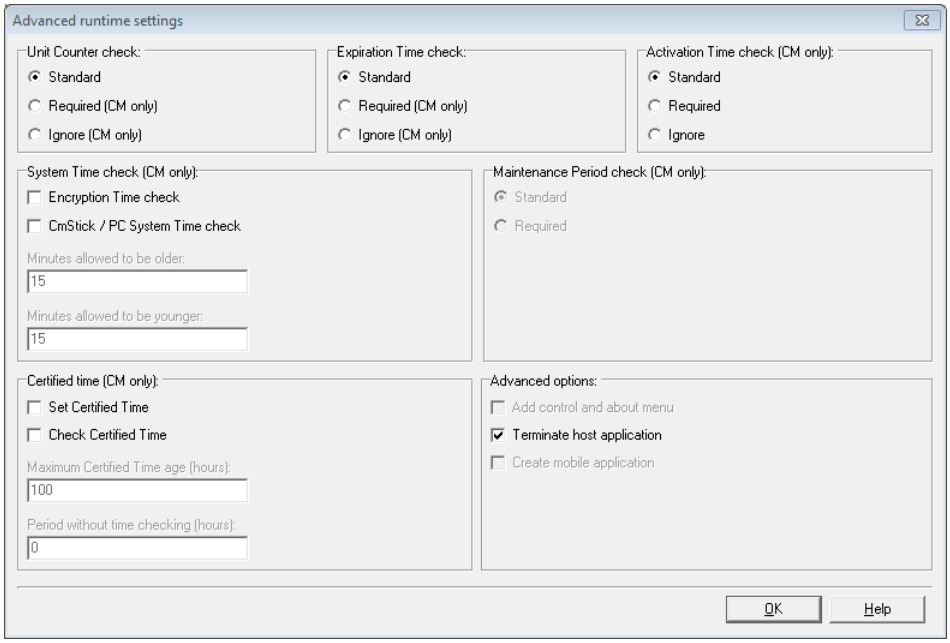


Figure 65: *AxProtector* .NET Standard - "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignore
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here 276).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches 0 (null) the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here[275]).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts when no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, when the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here[275]).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time[357] is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, when the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 (commandline option see here[276]).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems[84].

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only if a *Maintenance Period* exists. This corresponds to the default setting, even if on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmContainer* is connected with the computer or activated. When the *CmContainer* is connected or activated, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *Certified Time*. On updating the *certified time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here[270]).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here[357]..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmContainer*. The *certified time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check" you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time*.
Period without time checking (hours)	Specifies the period (in hours) if no check of the *Certified Time* certificate is taking place. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here[267]).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only if the *CmContainer* System Time is newer than the encryption time.

Element	Description
	Requires at least *CodeMeter*® 4.10.
CmContainer / PC System Time check	When activated these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor, the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

Advanced options

This group allows to set further options.

Element	Description
Terminate host application	When no valid license is found, in the case of protected DLL application files the calling `*.exe` is terminated (commandline option see here 277).
Create mobile application	[not yet implemented]

7.4.3.4 Security Options

This input window lets you select from different mechanisms and methods for protecting your application. You are able to scale the degree of security for yourself, for example, the search intensity for debugger or if a *CmContainer* is locked.

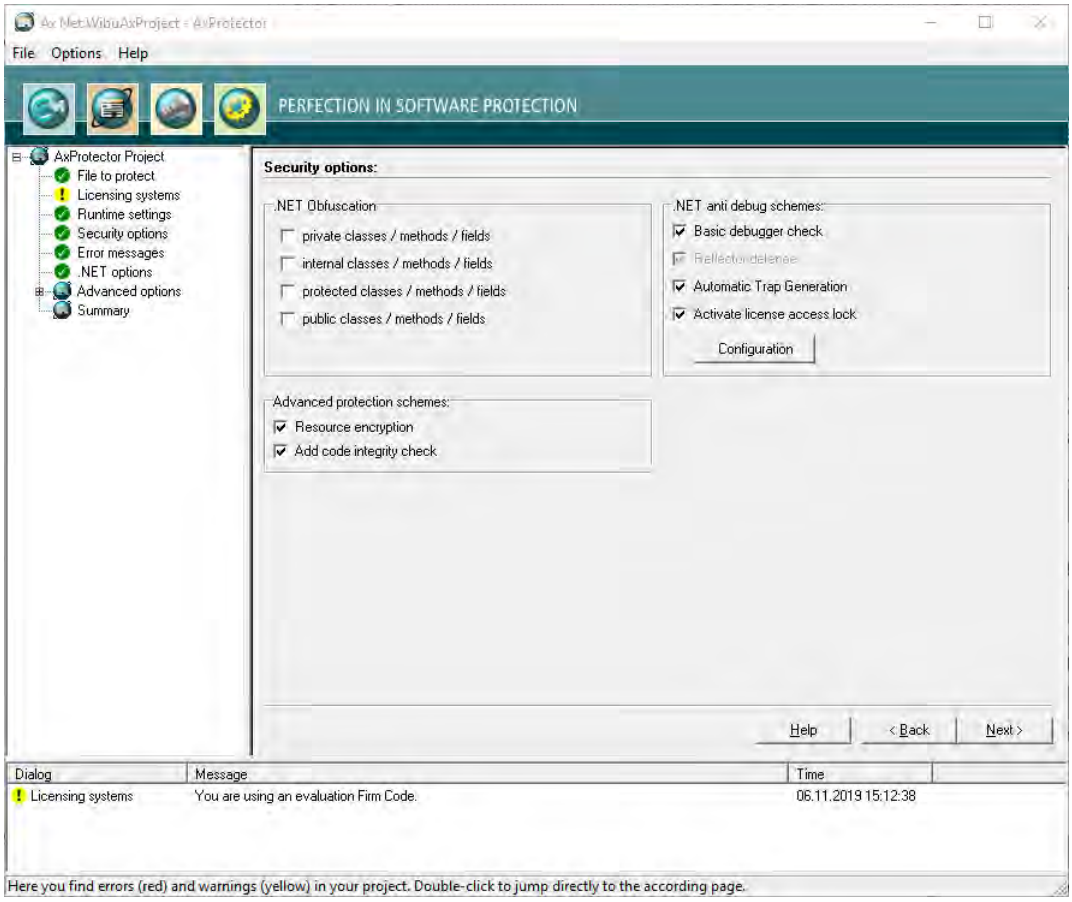


Figure 66: *AxProtector* .NET Standard - "Security Options"

.NET Obfuscation

The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information (commandline option see here 274). Elements comprise classes, methods, and fields.

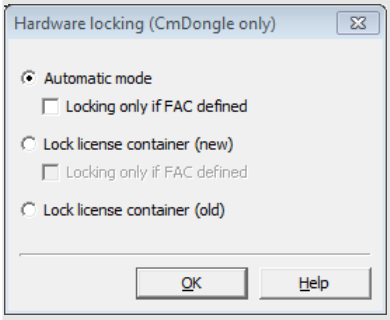
Element	Description
private classes / methods / fields	obfuscates private elements
internal classes / methods / fields	obfuscates internal elements
protected classes / methods / fields	obfuscates protected elements

Element	Description
public classes / methods / fields	obfuscates public elements

Anti-Debugging Schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here[267]).

Element	Description
Basic Debugger Check	The 'Basic Debugger Check', checks to see if a debugger is attached to your application. If a debugger is found, your application will not be started or exited.
Reflector defence	For protected .NET assemblies automatically a reflector defence is activated preventing decompiling.
Automatic Trap Generation	Automatically inserts hacker traps into the protected assembly (commandline option see here[279]).
Activate license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the **"Configuration"** button. This button is activated only for *CodeMeter*.
Configuration	If the option **"Activate license access lock"** is activated, you are able to define further settings in the dialog which opens by clicking the **"Configuration"** button: Depending on the Firmware used this dialog allows to define separate locking scenarios (for more detailed information see separate CodeMeter Developer Guide, section "Advanced CodeMeter Features | Locking a CmContainer").

Locking Scenario	Description
immediate locking	is performed starting with Firmware Version 1.14 as soon as a debugger is detected.
prepared locking	is performed by checking the *Firm Access Counter* (FAC). The *Firm Access Counter* locates 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`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked. The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.  Figure 67: *AxProtector* -.NET "Security Options - Hardware Locking"

The following settings are available:

Option	Description
"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher, the *Firm Item* is immediately locked. For compatibility reasons this represents the default setting.
"Automatic Mode" activated and "Locking only if FAC defined" activated	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher and a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (new)" activated and "Locking only if FAC defined" not activated	This option requires a Firmware version 1.14 or higher. The *Firm Item* is immediately locked. Seen from a security point of view this is the recommended option. This, however, requires that all *CmContainer* in the field must have a Firmware Version 1.14 an higher.
"Lock License Container (new)" and "Locking only if FAC defined" activated	This option requires a Firmware version 1.14 or higher. If a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.

Element	Description	
	Option	Description
	"Lock License Container (old)" activated	Independent from the Firmware version, if a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. This holds for all Firmware versions. If 'prepared locking' is programmed, the *Firm Access Counter* is decremented by a value of `1`.

Advanced protection schemes

The advanced protection schemes deeply intervene into your application. In some cases, this may mean that some single mechanisms will not work due to compatibility reasons (commandline options see here 266).

Element	Description
Resource encryption	Also encrypts the .NET resources of your protected application. After the start of your application, the resources located in the PC memory and are decrypted "on demand".
Add code integrity check	The protected application is checked for code integrity using asymmetric authentication asymmetric authentication 47 mechanisms, if you check this box (commandline options see here 270). On code integrity check first a check sum (hash value) of the application is created and signed with the private key of the Individual Software Vendor (ISV). The hash value and the signature are added to the application. The recalculation and the integrity check of the hash value and thus of the application is performed at runtime check using the public key located in the software (AxEngine). Alternatively to the default private key you can also apply the commandline option -sig 266 to use an entry of a *Hidden* or *Secret Data* field to define another private key. Moreover, the code integrity check may also cover several executable files / libraries. Then each file is able to check all other files for integrity. Each file then requires the public key of the ISV: The hash value of the files to be checked then is recalculated and compared to the hash value signed with the private key.

7.4.3.5 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used or whether you use default error message windows.

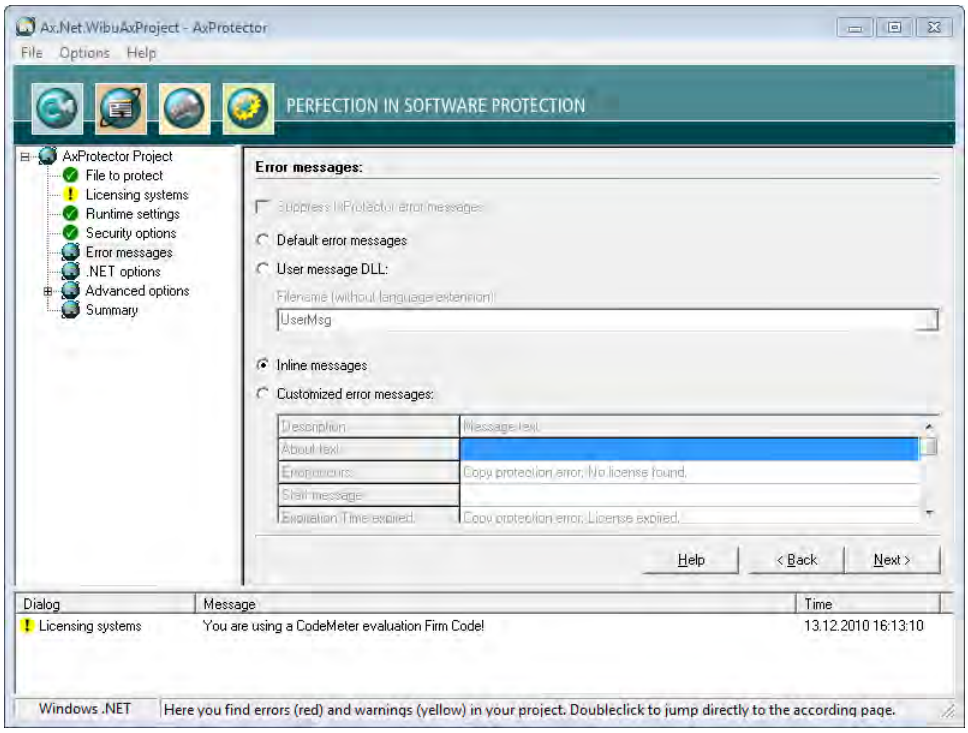


Figure 68: *AxProtector* .NET Standard "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
User Message DLL	The ability to use the User Message DLL is activated. **File name (without Language Extension)** Enter the file name without specifying path and language file extension.

Element	Description
	Either you program an own User Message DLL and place it in the same directory as your protected application, or you use the Wibu-Systems sample User Message for .NET (`%CodeMeter_Samples%Software Protection\C#\UserMessage`) and place it in the same directory as your protected application.
Inline Messages	Links for .NET projects, with an inline assembly, can also be configured by `*.ini` files (commandline option see here[278]). When using Inline UserMessages the logging is saved to the directory "`%CommonApplicationData%`". When you want to specify another path specify the parameter `LogPath<Path>` in the `*.ini` file.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.3.6 .NET Options

This page allows you to specify further .NET Standard settings.

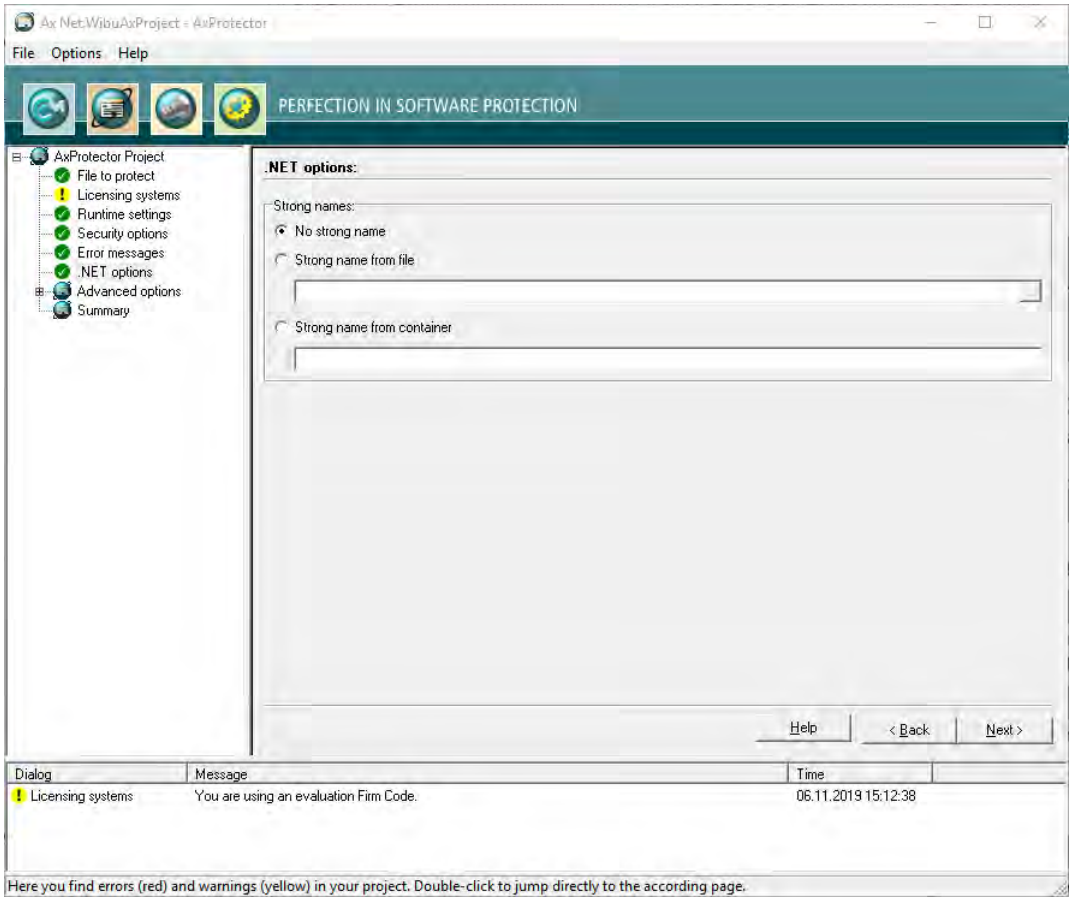


Figure 69: *AxProtector* .NET Standard - ".NET Options"

.NET Options

Here you are able to specify whether your assembly is signed by *AxProtector*.

Element	Description
No Strong Name	Activate this checkbox to not sign your assembly.
Strong Name from File	Activate this checkbox to use a source file to sign the program class. Then specify a file holding the key pair to generate a strong name (commandline options see here[279]).
Strong Name from Container	Activate this checkbox to use a container file to sign the program class (commandline options see here[279]).

7.4.3.7 Advanced Options

This input window lets you set further options for the encryption using *IxProtector*.

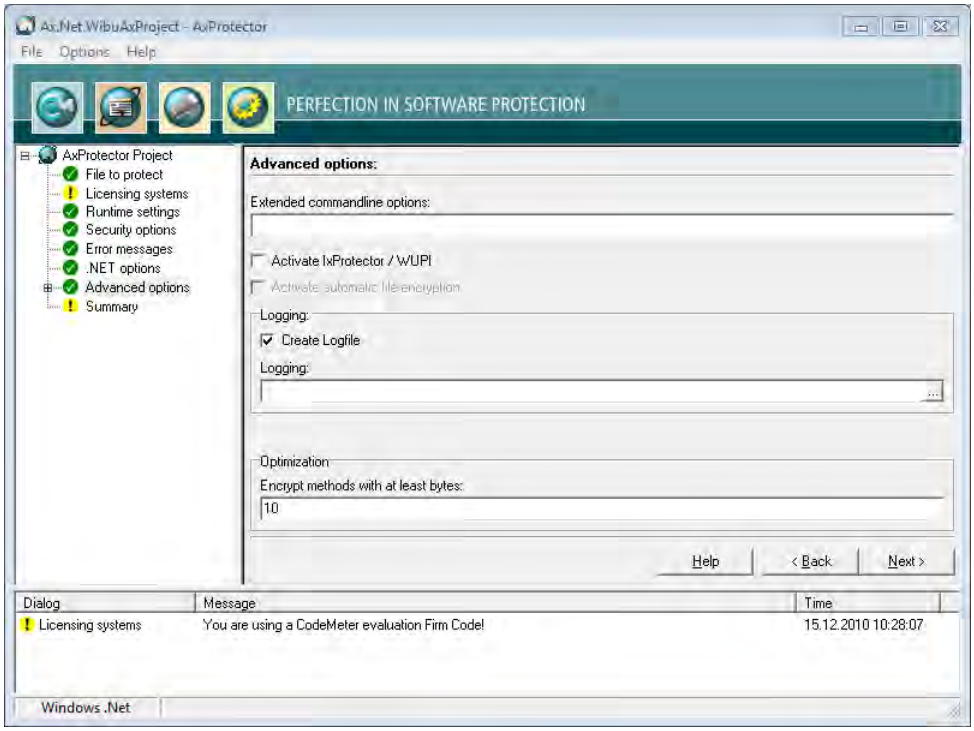


Figure 70: *AxProtector* .NET Standard - "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (commandline option see here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector.*
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.
Optimization	For an optimized performance specify here the minimum size for assemblies to be encrypted. The default setting is 10 bytes. This way you are able to exclude methods from encryption which are smaller than the number of bytes you specify here. By setting a value of 0 this feature is deactivated. Commandline option see here 275.

7.4.3.7.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

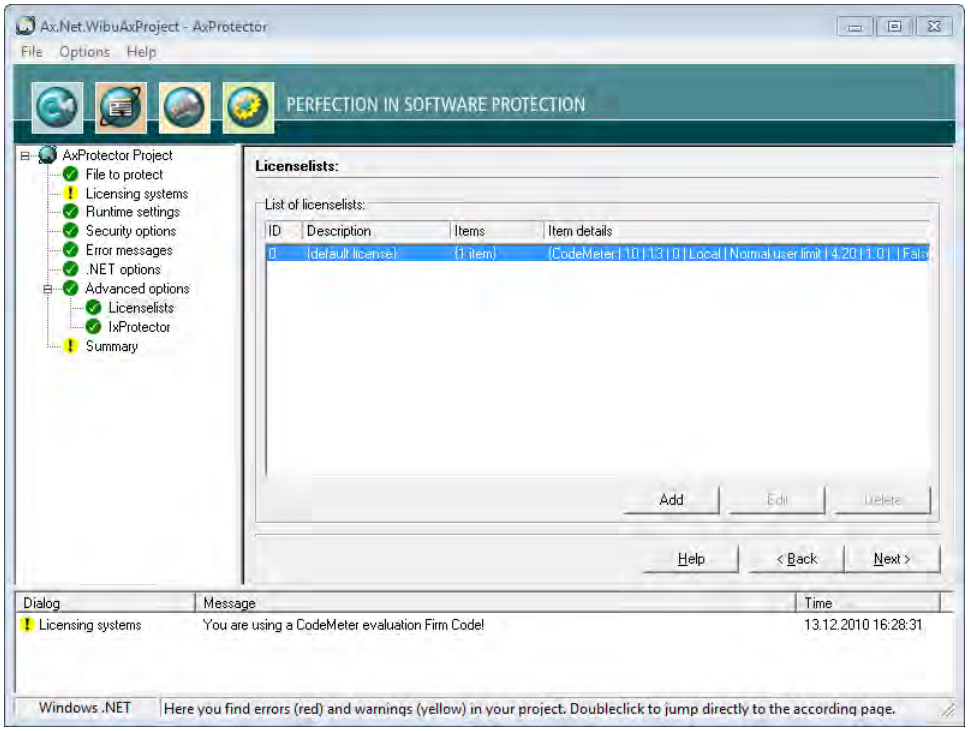
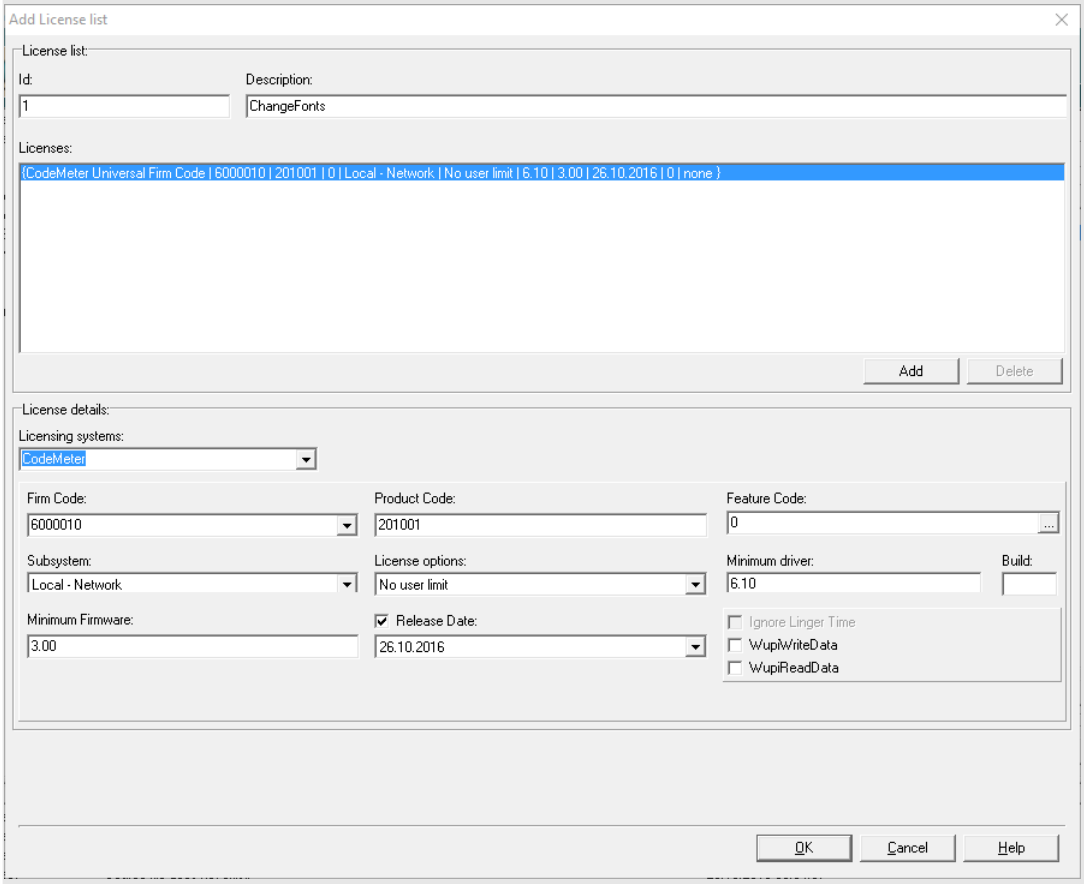


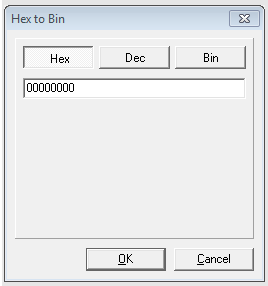
Figure 71: *AxProtector* .NET Standard - "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 72: *AxProtector* .NET Standard - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) / **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle `5010` *CmActLicense* Evaluation *Firm Code* — CmActLicense Commandline option see here 264 .

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 73: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description
	Firm Codes (licensing system) / Version `5010, 5.000.000-5.999.999` (*CmActLicense*) / 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

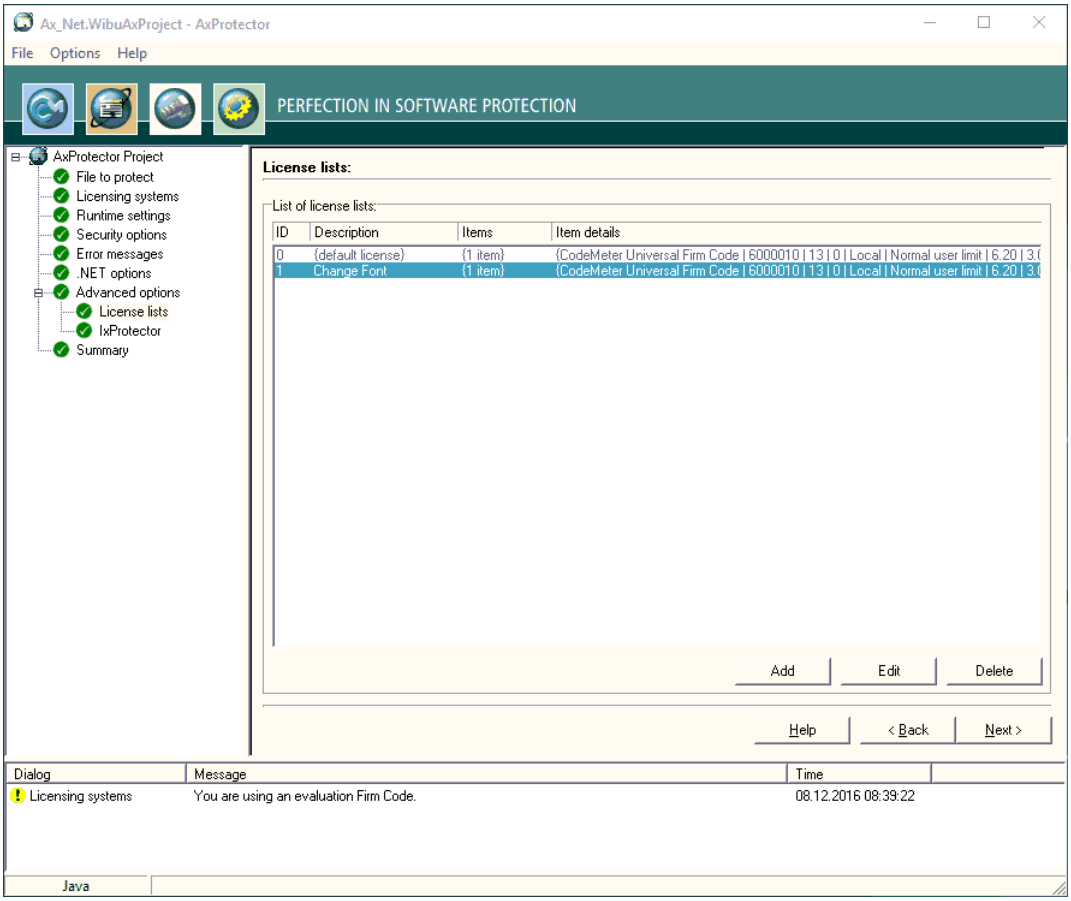


Figure 74: *AxProtector* .NET Standard - "Completed License Lists"

7.4.3.7.2 IxProtector

Using this menu item allows you to separately define single encryption types for single assembly elements.

In the case you activated the checkbox **"IxProtector"** in the menu item **"Advanced options"** the source assembly is loaded and displayed in a tree view making available all name spaces, classes, and modules.

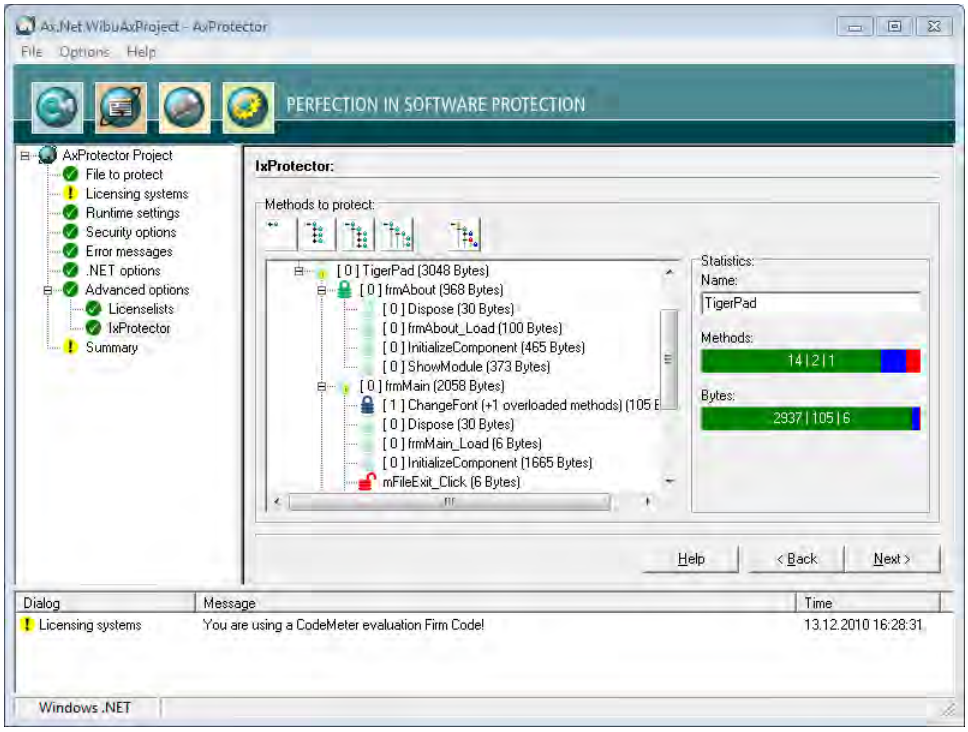


Figure 75: *AxProtector* .NET Standard - "IxProtector"

Click the different buttons in the upper "IxProtector" area to select from different assembly views.

Views

Buttons	Description
	Closes all assembly levels of the tree structure.
	Expands the name space level of the assembly.
	Expands the class level of the assembly.
	Expands the method level of the assembly.
	Expands all parent levels of the assembly. In this view see all levels where modifications have been made.

The area "Statistics" on the right shows you more encryption details depending on the selection you have made for the tree view.

Element	Description
Name	This field refers to the name of the element you have marked in the tree view.
Methods	Using different colors the bar 'Methods' shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted methods for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.
Bytes	Using different colors the bar 'Bytes' also shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted bytes for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.

You also have the option to separately assign the protection technologies *AxProtector* and *IxProtector* to single assembly elements, or exclude single elements from encrypting. To assign a protection technology by using the secondary menu, please proceed as follows:

1. In the left tree view, select the favored assembly element (name space, class, or method).
2. Click the right mouse button.
   The secondary menu opens.
3. Assign the favored encryption types by using symbols.

The License List IDs you are prompted are automatically transferred from the entries you added to the license list.

Symbol	Description
	Excludes the selected element from encryption.
	Encrypts the selected element using *AxProtector* (License List ID with a value of 0, i.e. default license).
	Encrypts the selected element using *IxProtector* (License List ID with a value unequal to 0, i.e. according to existing license list entries).
	This icon marks methods that are excluded from encryption due to the size of the method. The threshold can be set on the page 'Advanced Options' in the area optimizing

The modifications you made instantly display in the left area.

7.4.3.8 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

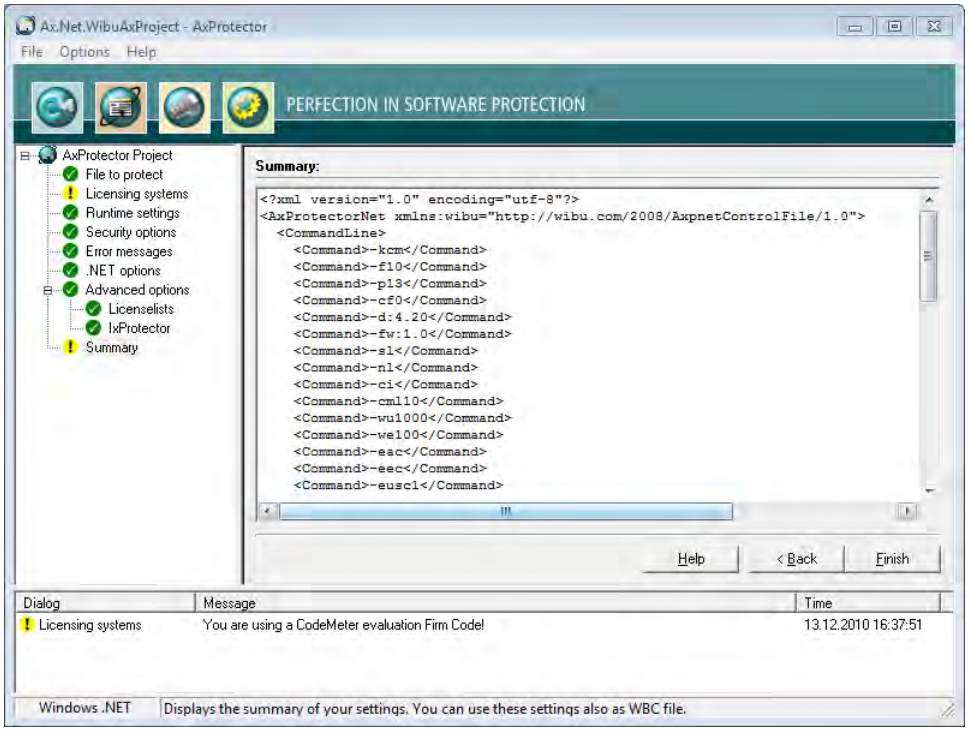


Figure 76: *AxProtector* .NET Standard "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

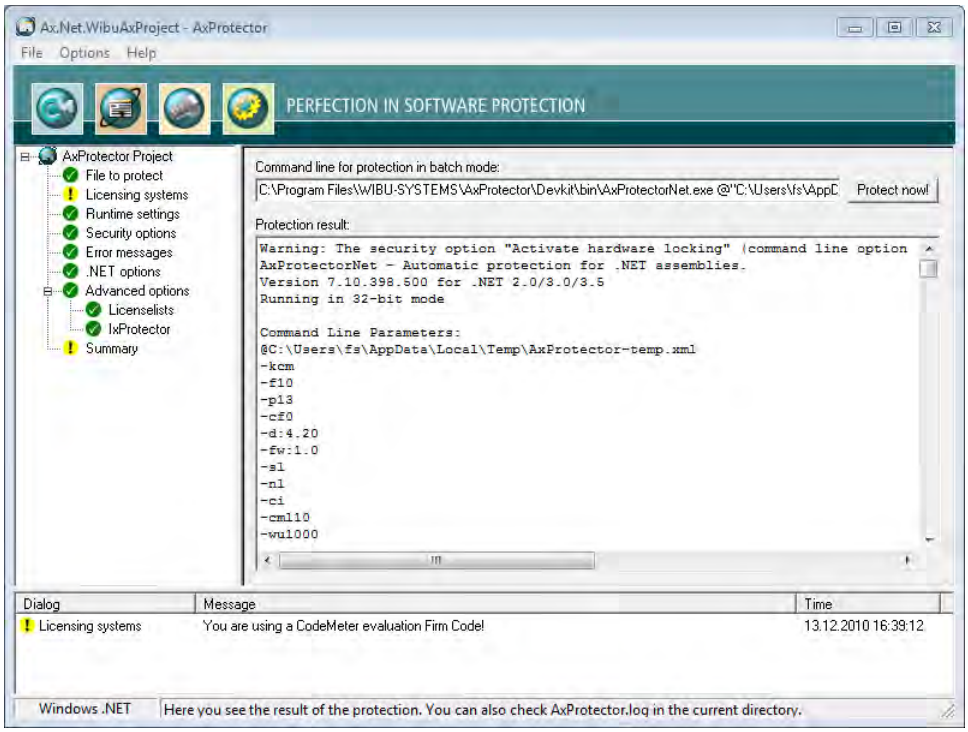


Figure 77: *AxProtector* .NET Standard - "Encryption Result"

Element	Description
Protect now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.4.4 macOS Application or Dylib

For this project type applications to be encrypted comprise macOS applications starting with Version 10.4. Application created for macOS 10.5 and higher require AxProtector Version 8.20.The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
macOS Application or Dylib	AxProtector macOS	√	Windows commandline [263] In a separate commandline for macOS, running on macOS operating systems, you are also able to insert encryption parameter [149] .

The following menu items are available in the navigation windows:

- File to protect [128]
- Licensing Systems [129]
- Runtime Settings [134]
- Security options [137]
- Error Messages [137]
- Advanced Options [141]
    - License lists [141]
    - IxProtector [146]
- Summary

7.4.4.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

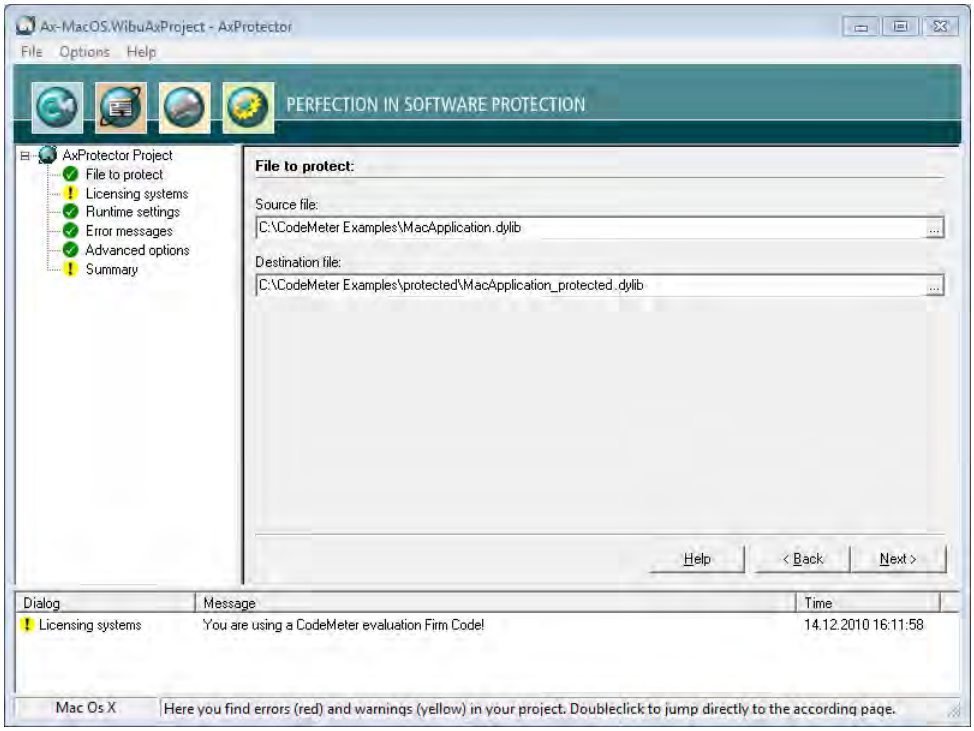


Figure 78: *AxProtector* - macOS "File to Protect"

File to Protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here [279] .

7.4.4.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

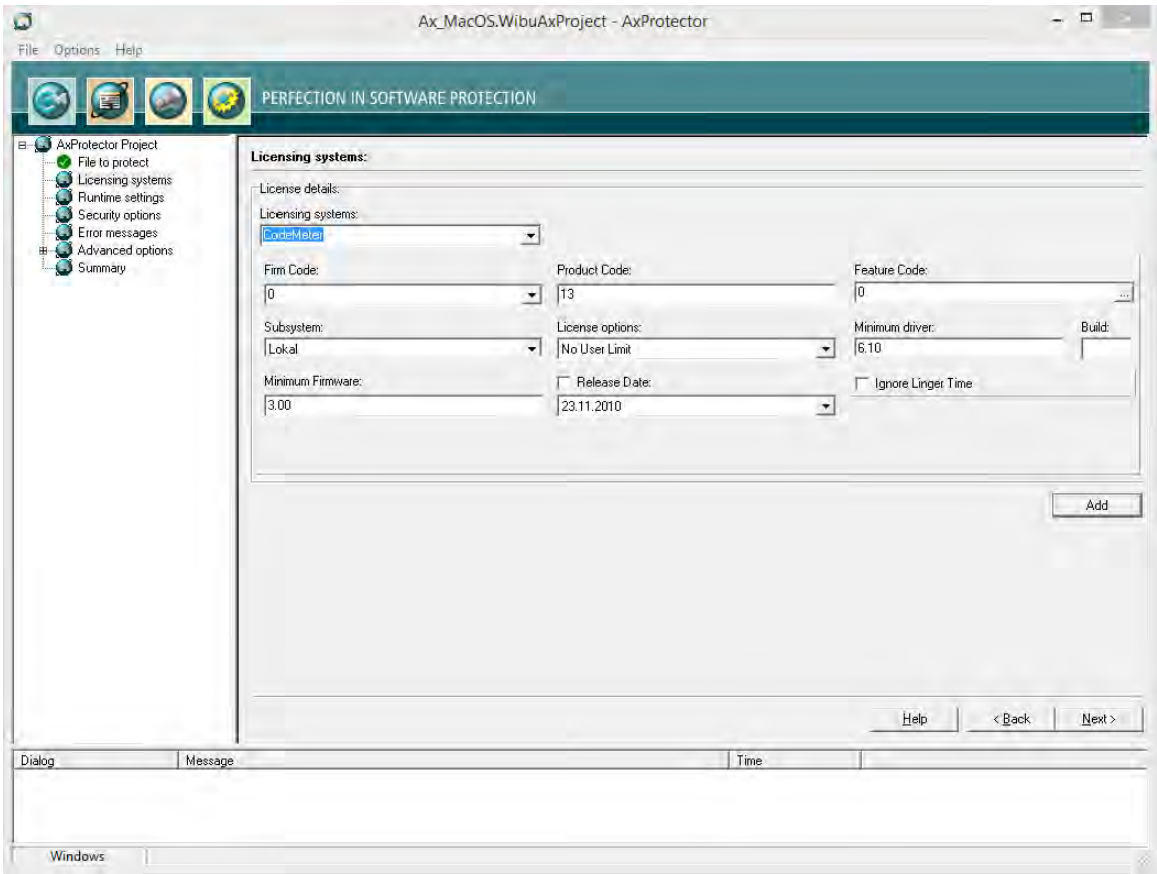
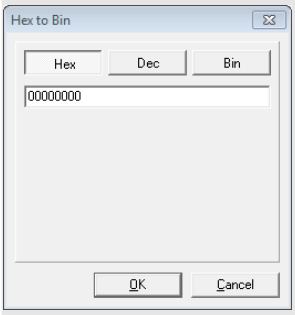


Figure 79: *AxProtector* - macOS "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: *Firm Code CodeMeter* Software Development Kit (SDK) — Licensing system `6000010` Evaluation *Universal Firm Code* — *CodeMeter*

Element	Description
	<table><tr><th>*Firm Code CodeMeter* Software Development Kit (SDK)</th><th>Licensing system</th></tr><tr><td>`10` *CmDongle* Evaluation *Firm Code*</td><td>*CmDongle*</td></tr><tr><td>`5010` *CmActLicense* Evaluation *Firm Code*</td><td>*CmActLicense*</td></tr></table> Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 80: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). <table><tr><th>Element</th><th>Description</th></tr><tr><td>Local</td><td>This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.</td></tr><tr><td>Network</td><td>This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.</td></tr><tr><td>Local - Network</td><td>This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.</td></tr><tr><td>Network - Local</td><td>This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.</td></tr></table>
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). <table><tr><th>Element</th><th>Description</th></tr><tr><td>Normal user limit</td><td>Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.</td></tr><tr><td>Station Share</td><td>Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.</td></tr><tr><td>WibuKey Compatibility Mode</td><td>Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.</td></tr><tr><td>Exclusive Mode</td><td>Here a protected application can be started only once on a PC.</td></tr><tr><td>No user limit</td><td>Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.</td></tr></table>
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>6.10 This supports the License Transfer feature.</td></tr></table>

Element	Description
	<table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr></table> Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>3.00 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr></table> Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so.

Please click the **"Add"** button to add additional license(s).

7.4.4.2.1 Licensing Systems - Add licenses

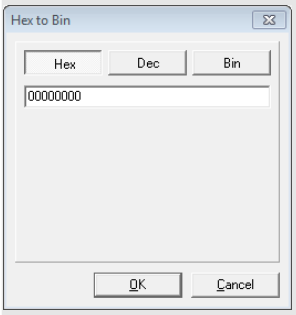
Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so.

Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description

Licensing systems	Select from the dropdown control the desired licensing system. Available are the following entries:
	`CodeMeter`
	`WibuKey`
	For setting *WibuKey* options, see the separate "WibuKey Developer Guide".
	If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.
	In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Element	Description
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: `6000010` Evaluation *Universal Firm Code* — *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* — *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* — *CmActLicense* Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 81: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** — **Description** Local — This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network — This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network — This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local — This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** — **Description** Normal user limit — Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share — Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode — Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode — Here a protected application can be started only once on a PC. No user limit — Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Element	Description
	Firm Codes (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.4.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

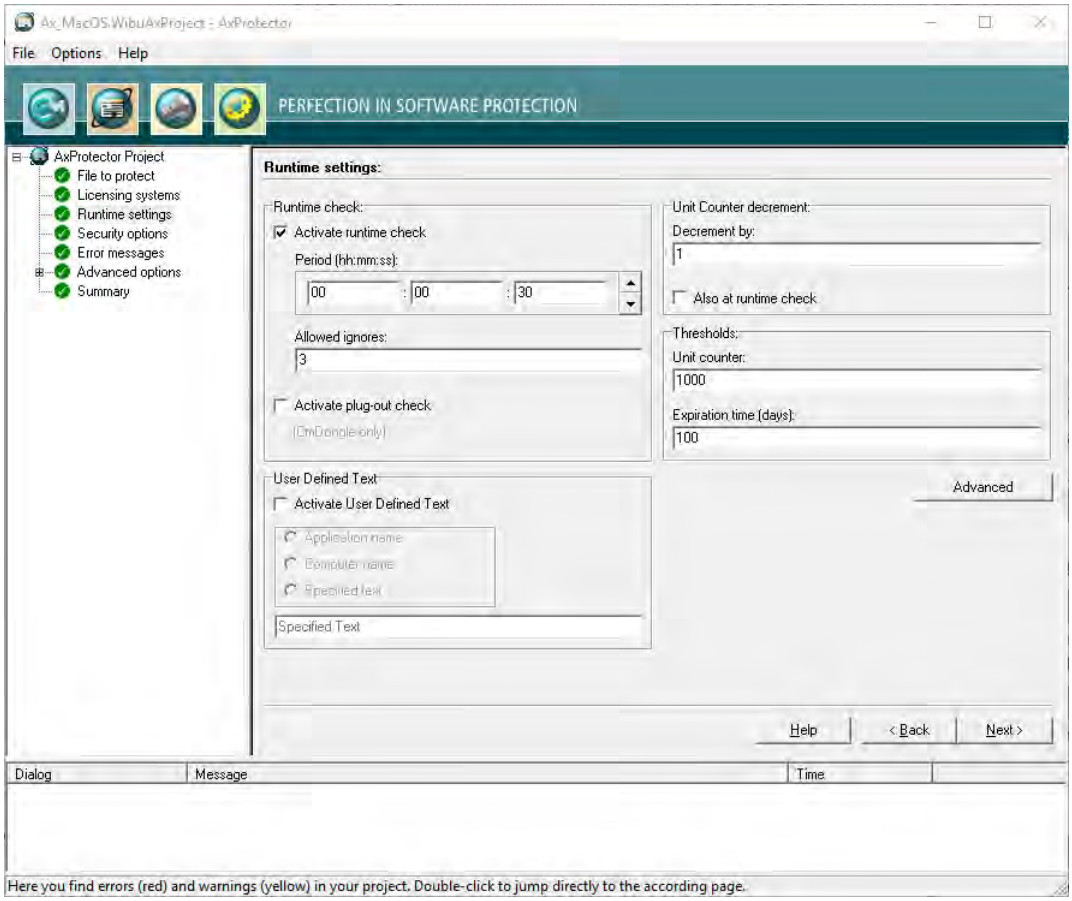


Figure 82: *AxProtector* - macOS "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: `hours: minutes: seconds`.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.
Activate Plug-out Check (only CmDongle)	This option closes the protected application if the *CmDongle* is removed while the application is running. Immediately, an error message is issued. This option is valid for *CmDongle* only. Commandline option see here 267 .

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the ***Unit Counter*** is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above, every 30 seconds (see the defined period) a set ***Unit Counter*** is decremented by a value of 1.
Also at Runtime Check	Decrements the ***Unit Counter*** also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime Check 134 " group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here[137].

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here[277].
Expiration Time (days)	When the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here[277].

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here[279].

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. Element / Description: Application name — uses the application name. Computer name — uses the computer name. Specified text — uses the specified text in the field of the same name.

7.4.4.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

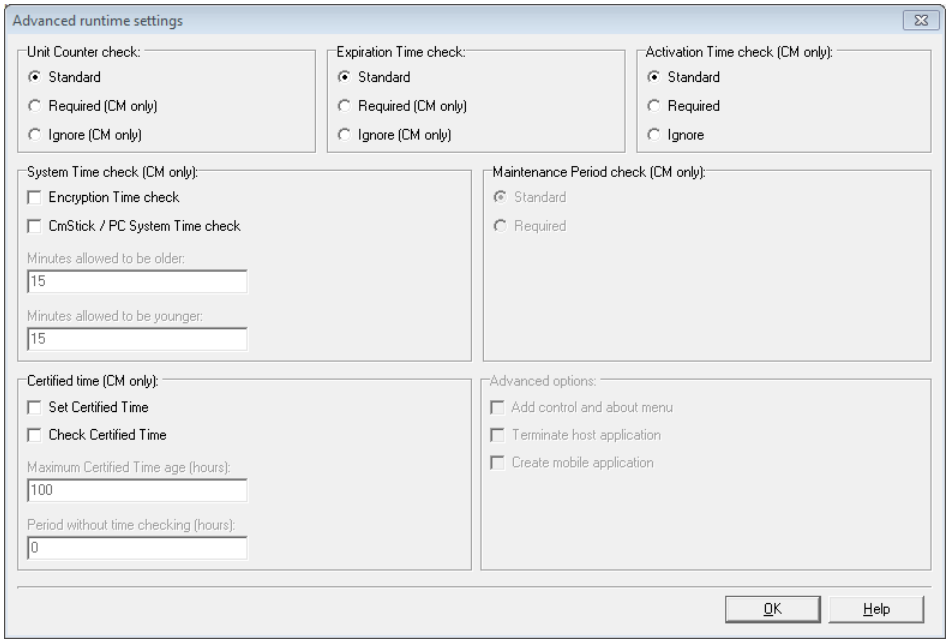


Figure 83: *AxProtector* - macOS "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignored
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here[276]).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches 0 (null) the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts when no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, when the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time 357 is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, when the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 performed if the date is within the defined period (commandline option see here 276).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems 129.

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only if a *Maintenance Period* exists. This corresponds to the default setting, even when on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmContainer* is connected with the computer or activated. If the *CmContainer* is connected or activated, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *certified time*. On updating the *Certified Time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here 270).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here 357 ..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmDongle*. The *Certified Time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check", you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time.*
Period without time checking (hours)	Specifies the period (in hours) when no check of the *Certified Time* certificate is performed. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here 267).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only when the *CmContainer* System Time is newer than the encryption time.

Element	Description
	Requires at least *CodeMeter*® 4.10.
CmContainer / PC System Time check	When activated these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor, the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

7.4.4.4 Error Messages

This input window lets you define the messages displayed if errors occur.

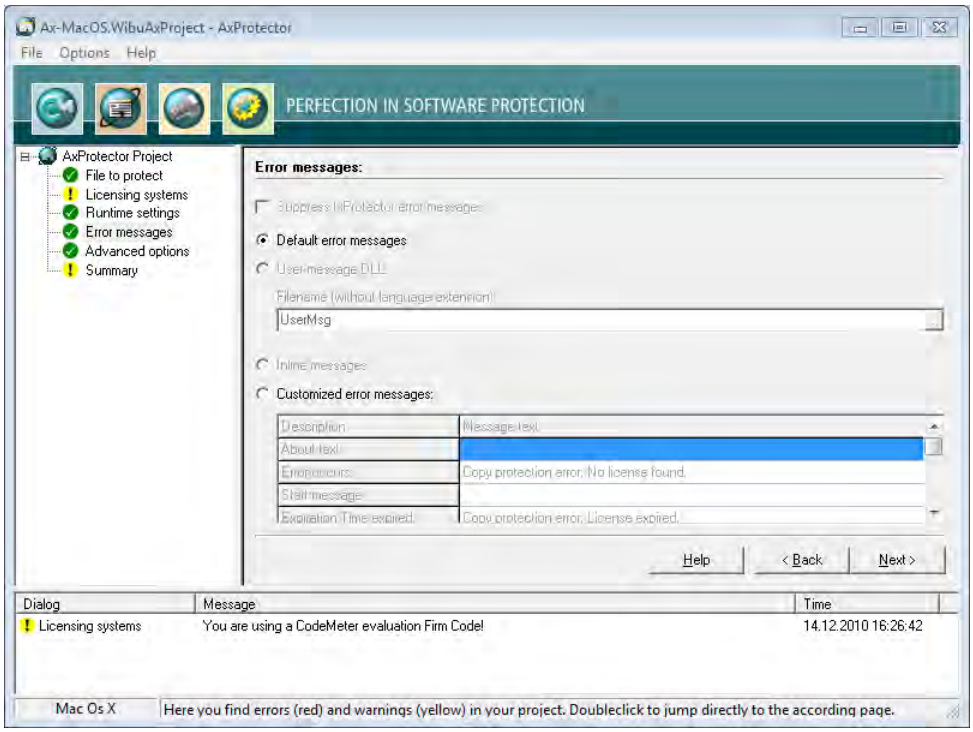


Figure 84: *AxProtector* - macOS "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.4.5 Security Options

This input window lets you select from different mechanisms and methods for protecting your application. You are able to scale the degree of security for yourself, for example, search intensity for debugger or whether a *CmContainer* is locked.

When the options you set here turn out to be incompatible with your protected application, you are also able to separately deactivate single security options.

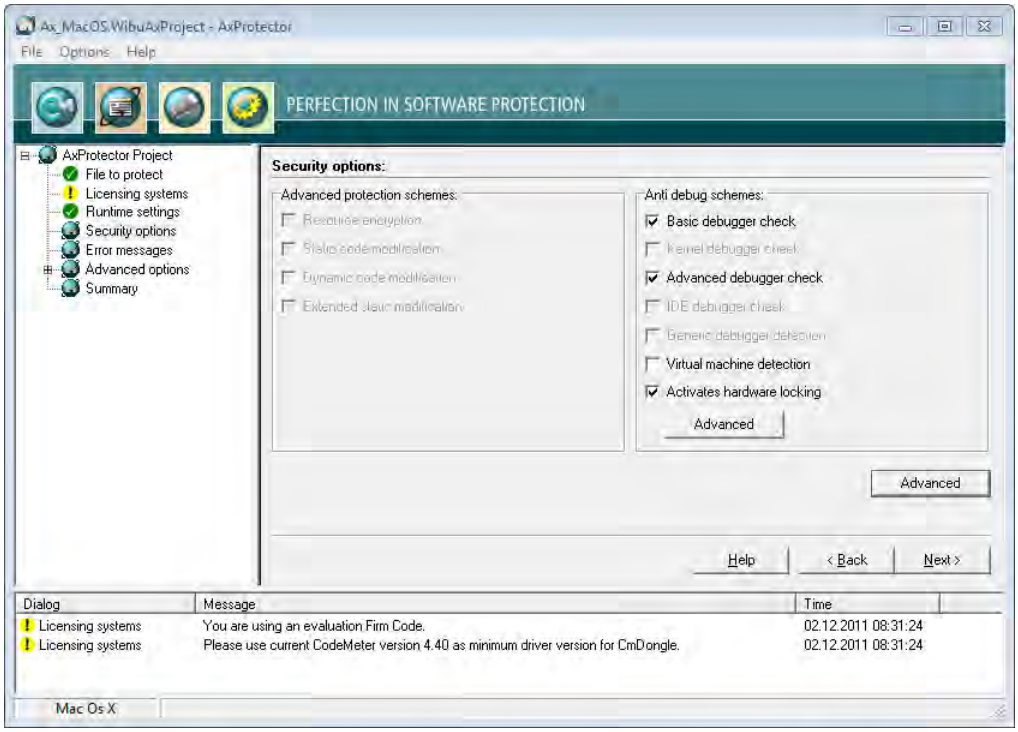


Figure 85: *AxProtector* - macOS "Security Options"

Anti-Debug Schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here 268).

Element	Description
Basic Debugger Check	The 'Basic Debugger Check', checks to see if a debugger is attached to your application. When a debugger is found, your application will not be started or exited.
Advanced Debugger Check	Checks in an advanced search for debugger programs which may run parallel to your application, also cracker tools, such as, ImpREC, are detected. In the case a debugger is found, your application will not be started.
Virtual Machine Detection	Detects if the application is to be started on a virtual machine and prevents this.
Activate license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the **"Configuration"** button. This button is activated only for *CodeMeter*.
Configuration	If the option **"Activate license access lock"** is activated, you are able to define further settings in the dialog which opens by clicking the **"Configuration"** button: Depending on the Firmware used this dialog allows to define separate locking scenarios (for more detailed information see separate CodeMeter Developer Guide, section "Advanced CodeMeter Features | Locking a CmContainer").

Locking Scenario	Description
immediate locking	is performed starting with Firmware Version 1.14 as soon as a debugger is detected.
prepared locking	is performed by checking the *Firm Access Counter* (FAC). The *Firm Access Counter* locates 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`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked. The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.

Element	Description
	Locking Scenario / **Description**

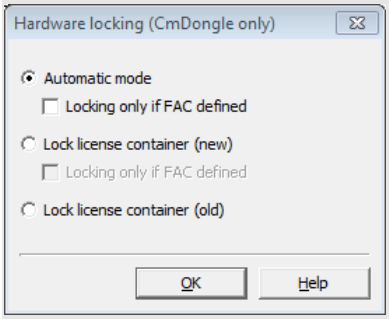


Figure 86: *AxProtector* - macOS "Security Options - Hardware Locking"

The following settings are available:

Option	Description
"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher, the *Firm Item* is immediately locked. For compatibility reasons this represents the default setting.
"Automatic Mode" activated and "Locking only if FAC defined" activated	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher and a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (new)" activated and "Locking only if FAC defined" not activated	This option requires a Firmware version 1.14 or higher. The *Firm Item* is immediately locked. Seen from a security point of view this is the recommended option. This, however, requires that all *CmContainer* in the field must have a Firmware Version 1.14 an higher.
"Lock License Container (new)" and "Locking only if FAC defined" activated	This option requires a Firmware version 1.14 or higher. If a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (old)" activated	Independent from the Firmware version, if a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. This holds for all Firmware versions. If 'prepared locking' is programmed, the *Firm Access Counter* is decremented by a value of `1`.

7.4.4.5.1 Advanced Security Options

This input window lets you define further settings.

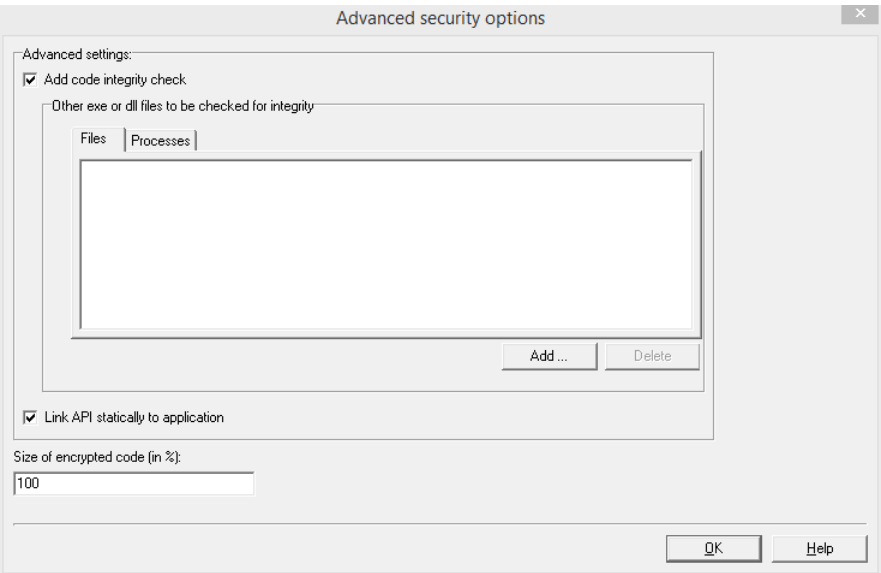
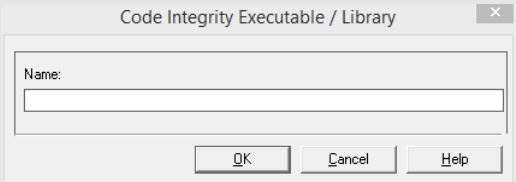
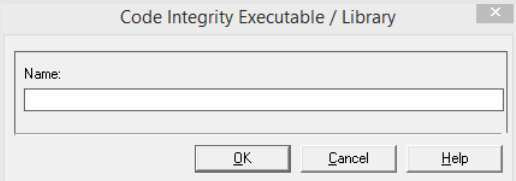


Figure 87: *AxProtector* - macOS "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

Element	Description
Add code integrity check	The protected application is checked for code integrity using asymmetric authentication 47 mechanisms, if you check this box (commandline options see here 270). On code integrity check first a check sum (hash value) of the application is created and signed with the private key of the Individual Software Vendor (ISV). The hash value and the signature are added to the application. The recalculation and the integrity check of the hash value and thus of the application is performed at runtime check using the public key located in the software (AxEngine). Alternatively to the default private key you can also apply the commandline option -sig 266 to use an entry of a *Hidden* or *Secret Data* field to define another private key. Moreover, the code integrity check may also cover several executable files / libraries. Then each file is able to check all other files for integrity. Each file then requires the public key of the ISV: The hash value of the files to be checked then is recalculated and compared to the hash value signed with the private key. To add other files for performing an integrity check, please proceed as follows. **1.** Set focus to tab "**Files**". **2.** Click the "**Add**" button. The dialog for adding displays.  **2.** Add a single or several executable files / libraries by completing the "**Name**" field. The sequence of the specified files does not matter. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button. Moreover, on encrypting a DLL also a list of applications can be transferred allowed to load these libraries. On loading the DLL then it is checked whether the process name includes one of the names specified in tab "**Files**". If not, an error message displays and subsequently the application closes. To add processes please proceed as follows: **1.** Set focus to tab "**Processes**". **2.** Click the "**Add**" button. The dialog for adding displays.  **3.** Add one or more processes which include one or more application names listed in tab "Files" by completing the field "**Name**". The sequence of the specified files does not matter. If the same application names are also specified in the list of tab "Files" also their code integrity is checked. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button.
Link API statically to Application	The *CodeMeter Core API* is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here 263).
Size of encrypted Code (in %)	Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here 270).

7.4.4.6 Advanced Options

This input window lets you set further encryption options.

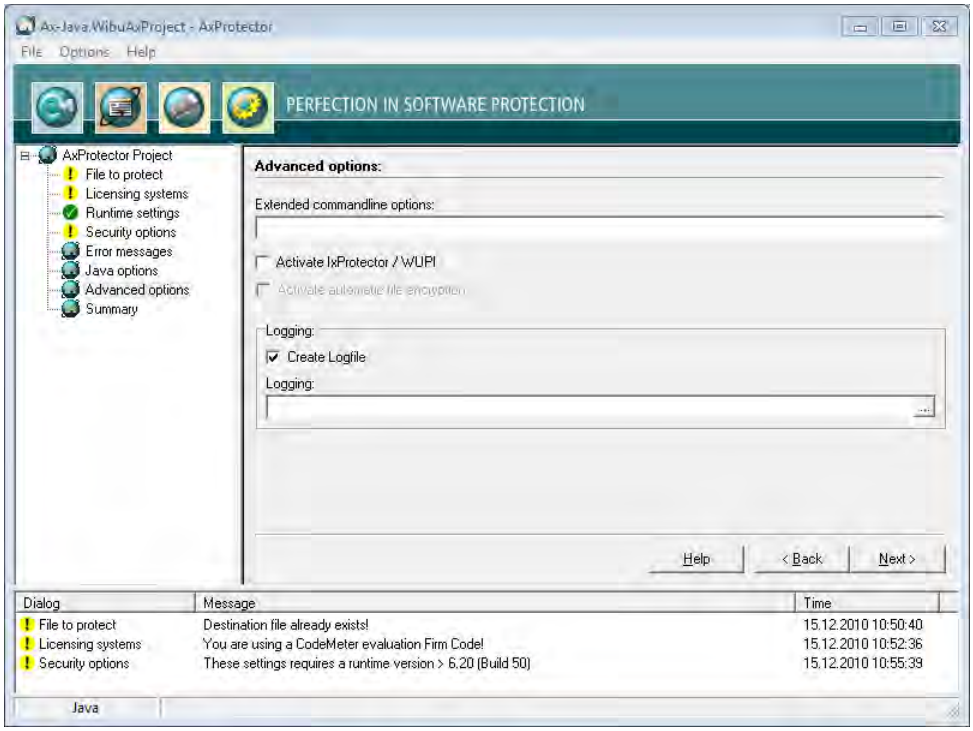


Figure 88: *AxProtector* - macOS "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector / WUPI	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289 . (commandline option see here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector.*
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.4.4.6.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

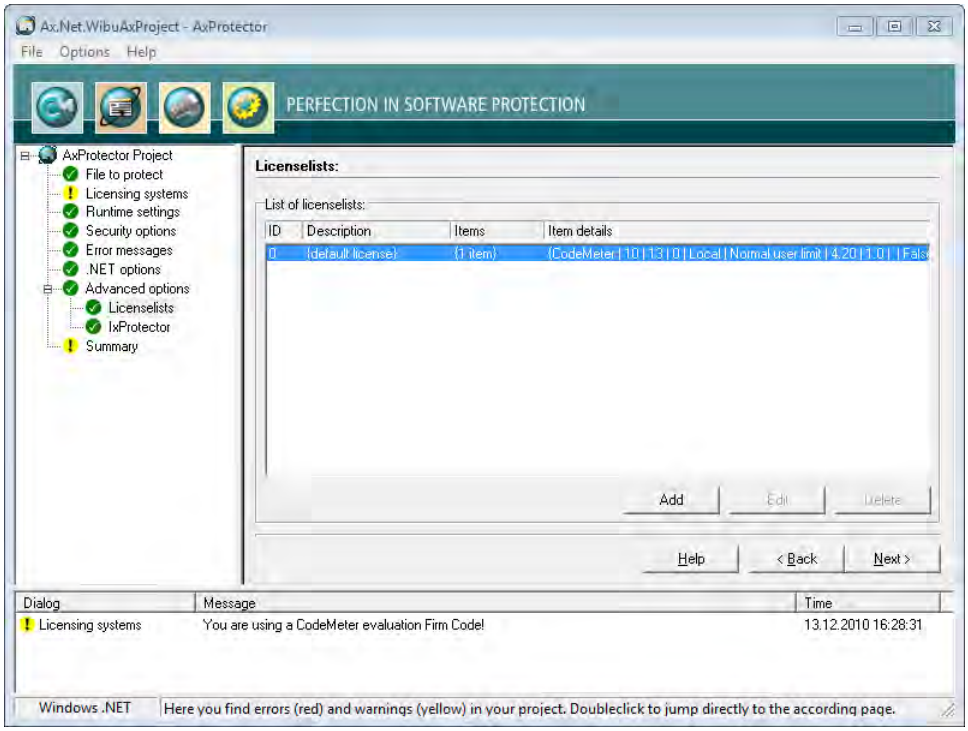
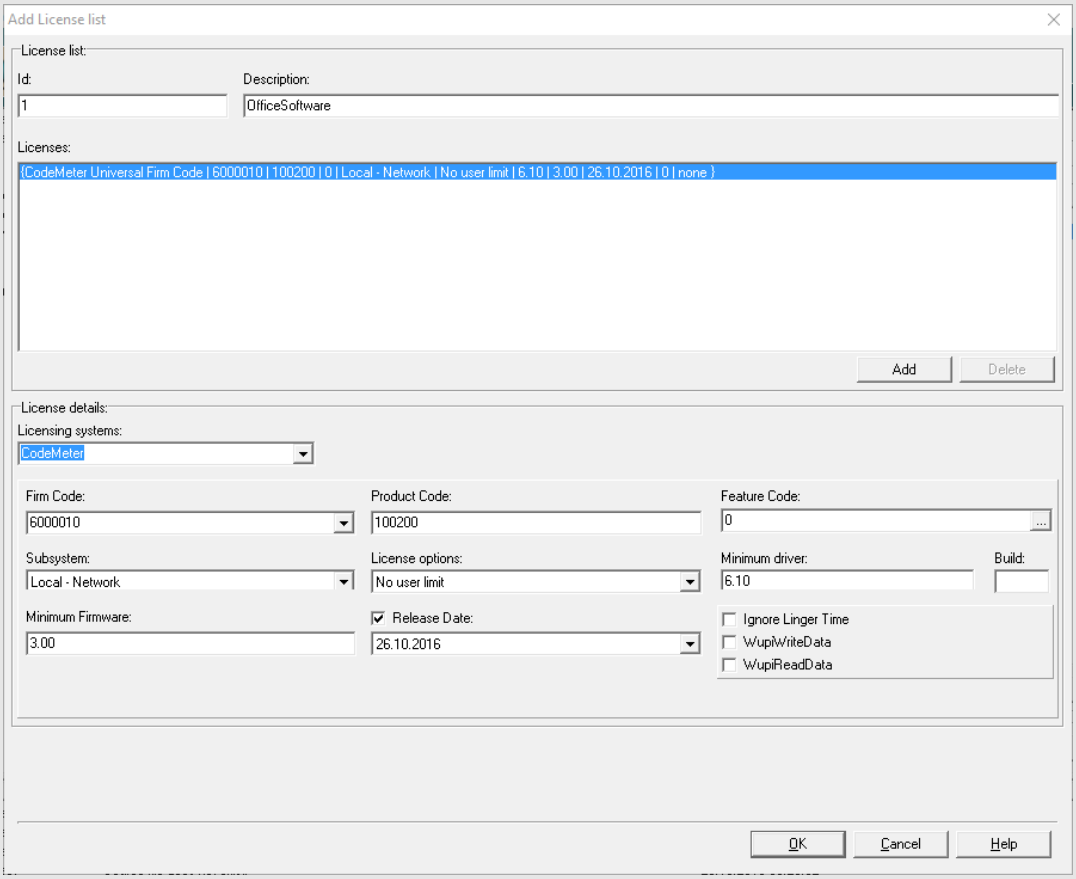


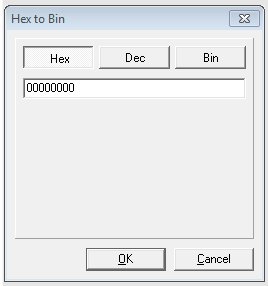
Figure 89: *AxProtector* macOS - "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 90: *AxProtector* macOS - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) — **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle `5010` *CmActLicense* Evaluation *Firm Code* — CmActLicense Commandline option see here 264.

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 91: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description
	Firm Codes (licensing system) / Version `5010, 5.000.000-5.999.999` (*CmActLicense*) / 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

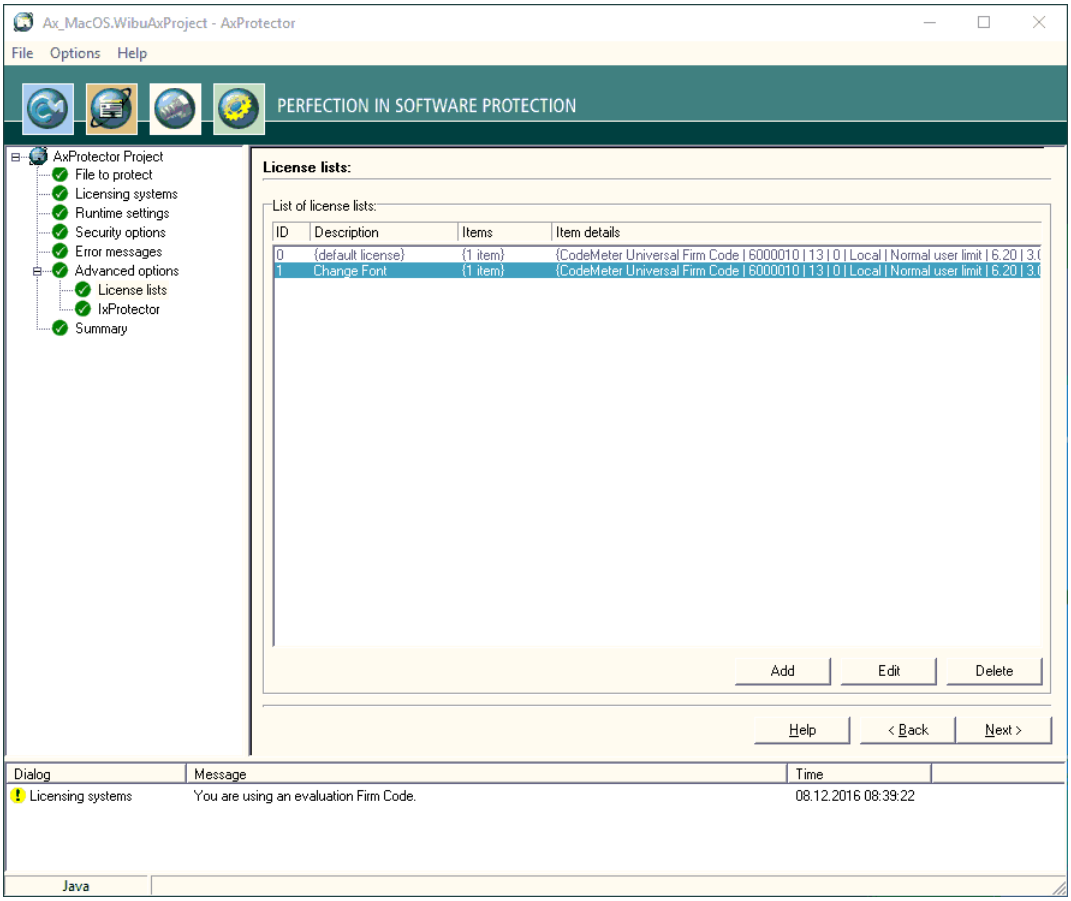


Figure 92: *AxProtector* macOS - "Completed License Lists"

7.4.4.6.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

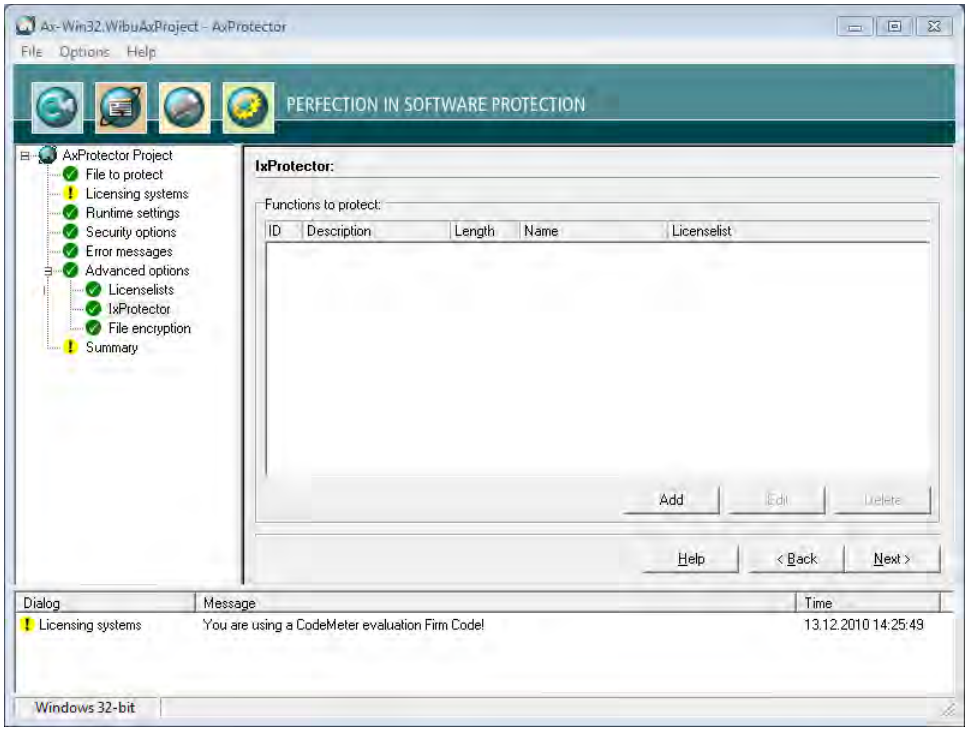


Figure 93: *AxProtector* - macOS - "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: 1. Click the **"Add"** button in the group "IxProtector Options". 2. Define the function by completing the fields in the "Function" group.

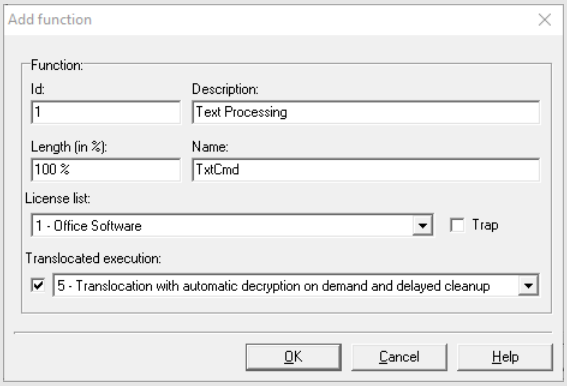


Figure 94: *AxProtector* - macOS - "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291.
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.
License List	Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td>Trap</td><td>Activates the trap function for the function.</td></tr><tr><td>Translocated execution</td><td>Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.<table><tr><th>Option</th><th>Description</th></tr><tr><td>1</td><td>Translocation with automatic decrpytion on demand and cleanup.</td></tr><tr><td>2</td><td>Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).</td></tr><tr><td>5</td><td>Translocation with automatic decrpytion on demand and delayed cleanup. (Default)</td></tr></table>Command line option see here 286 .</td></tr></table> **3.** Click the **"OK"** button. The new functions are added to the function list.

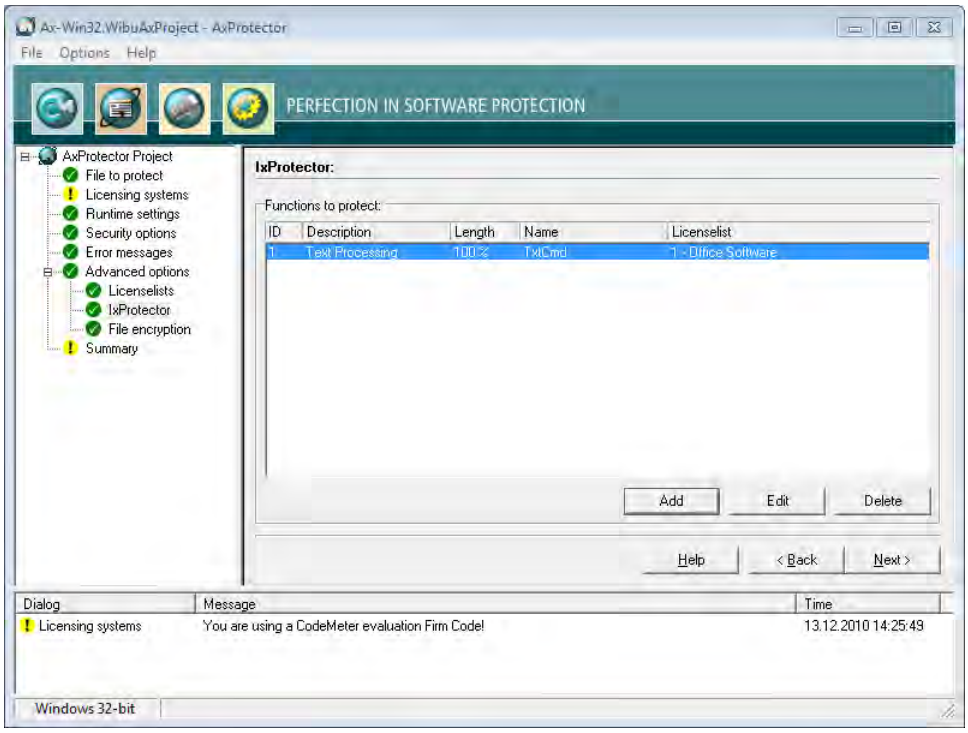


Figure 95: *AxProtector* - macOS - "Completed Function List"

7.4.4.7 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

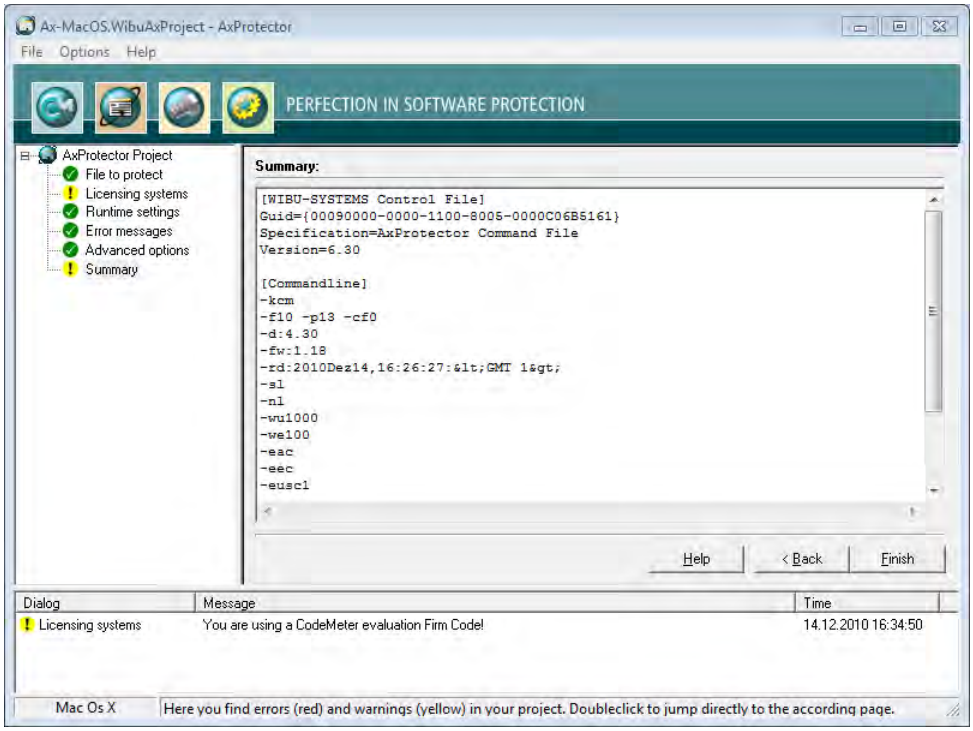


Figure 96: *AxProtector* - macOS "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

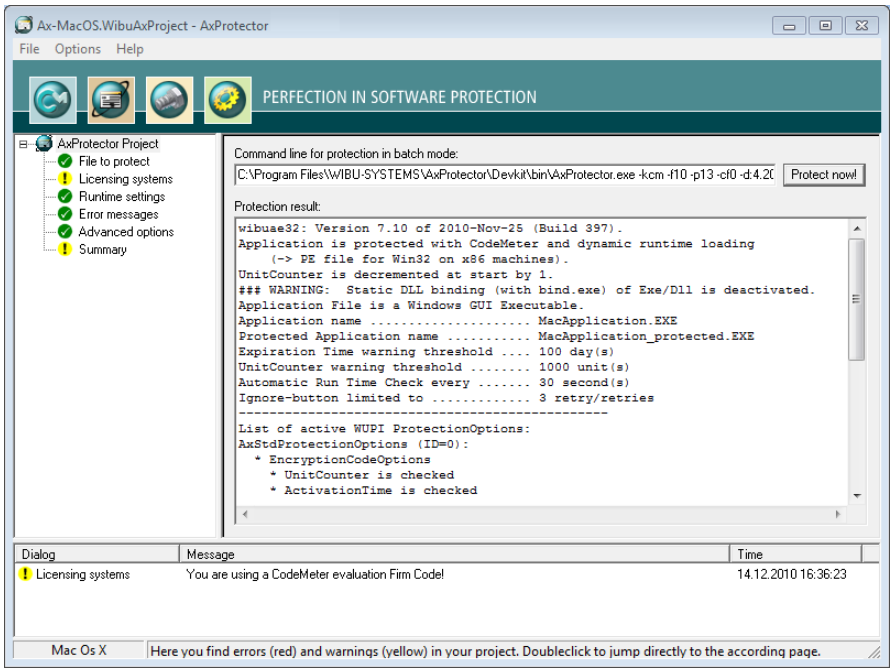


Figure 97: *AxProtector* - macOS "Encryption Result"

Element	Description
Protect now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.4.5 Java Application (jar file)

Compiled Java code, like .NET-Code, can be re-translated into uncompiled source code: easily and without any special programming knowledge required. Thus, almost everything what happens in the application is principally publicly available, and competitors are able to easily analyze the software. Your intellectual property is virtually unprotected. In addition, even a built-in license management can be easily removed from the software. Thus, sooner or later for each Java developer the question arises: How to protect intellectual property and to prevent use violations?

AxProtector Java solves this challenge. Basically, a Java compilation is composed of a mere collection of compiled classes, of class files. Usually, these are bundled, saved, and delivered as jar-archives. The basic principle of *AxProtector Java* is to separately encrypt each single class. For this purpose, automatically the `*.jar`-archive is unpacked, each class file is encrypted according to the selected settings, and afterward re-packed in the archive together with some necessary class files of Wibu-Systems.

For encrypting Java applications *AxProtector Java* supports Java version 8 and higher.

Additional security mechanisms

In addition to this loading principle, *AxProtector Java* extends the application by other security mechanisms. In order to ensure that the allocated license is still available for further use, and, for example, that the dongle was not disconnected, a periodical check at application runtime can be specified. Then the allocated license is re-checked by decryption operations in customizable intervals, and in the case that an error is returned, the application halts.

Signature check of the Runtime Environment

Since Version 6, Java sources are open and available. In principle, now anybody is able to assemble a slightly modified version of Java, and able to inward transfer own code into the native Java library to record the loading of decrypted classes. Therefore, in Java up to Version 9 the option exists to check the authenticity of the Java version in use. For that purpose, signatures of the native Java libraries are added to the application and checked on start. In the case a newer version of the Java library is used, *AxProtector* spots this, and offers to automatically download new signatures from the Wibu-Systems website. This way, the application is able to handle not yet released versions at the time of encryption.

Requirements

AxProtector Java supports Oracle Java and also the free and open source implementation of the Java platform OpenJDK. Along with the files located in the jar archive, the user requires the native `wibuXPM4J` library mentioned above. It is included for Windows and macOS in the Runtime Kits of *CodeMeter* and *WibuKey*, for Linux there exist small separate installer.

When encrypting an additional option is provided to include (white list), or to exclude (black list) specific classes. This allows, for example, to exclude classes of other vendors from encryption. Moreover, a minimum version can be specified.

This description so far related to Java applications, i.e. separate programs located on the user's hard drive. However, application scenarios using Java have become varied, and for example, also the protection of server applications becomes an option. For example, how to integrate software protection into the application server Tomcat?

Customized Use

AxProtector Java also meets protection requirements of, for example, Java Servlets, Eclipse Rich Client applications, or Java Web Start applications. When using *AxProtector Java* in such environments, you have to note some special requirements, and make customizations. Meanwhile, Wibu-Systems provides several `ClassLoader` especially designed to meet requirements in specific cases, for example, the `ServletClassLoader`, or the `EclipseClassLoader`. Contact Wibu-Systems Support and inquire for matching samples, or support on integration.The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
Java Application (Archive Format `*.jar`, Webarchive Format `*.war`)	AxProtector Java	√	Windows commandline 263 In a separate commandline for Java, running on Windows, macOS-, and Linux operating systems, you are also able to insert encryption parameter 174 .

The following menu items are available in the navigation windows:

- File to protect 151
- Licensing Systems 151
- Runtime Settings 157
- Security Options 160
- Error Messages 162
- Java Settings 163
- Advanced Options 164
    - License Lists 165
    - IxProtector 170
- Summary 172

7.4.5.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

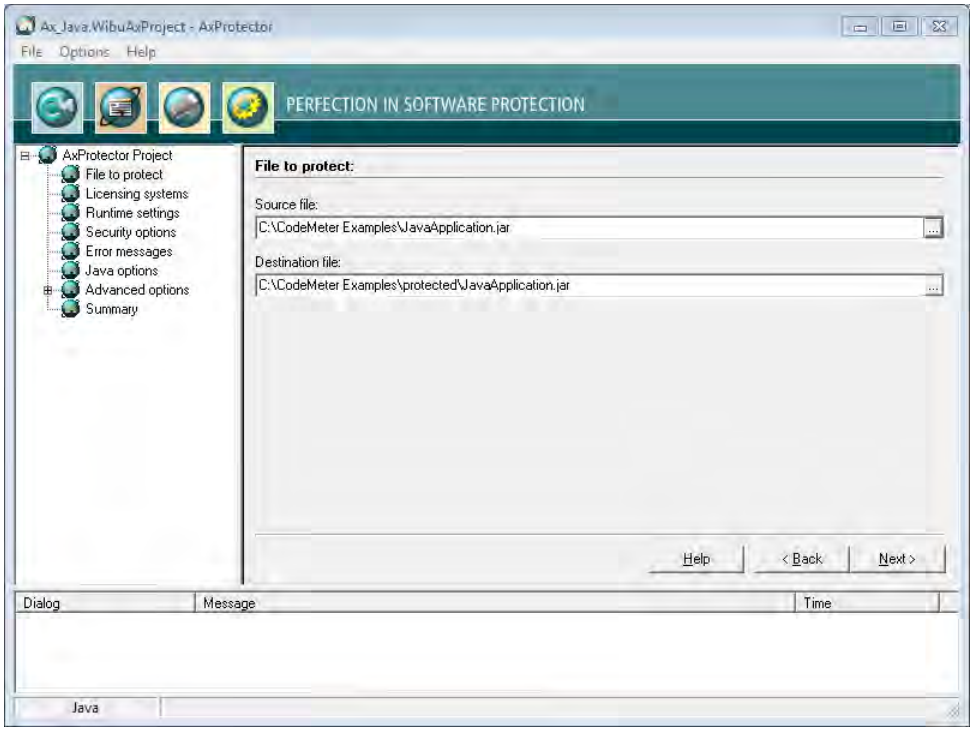


Figure 98: *AxProtector* - Java "File to Protect"

File to Protect

Element	Description
Source file	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination file	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.4.5.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

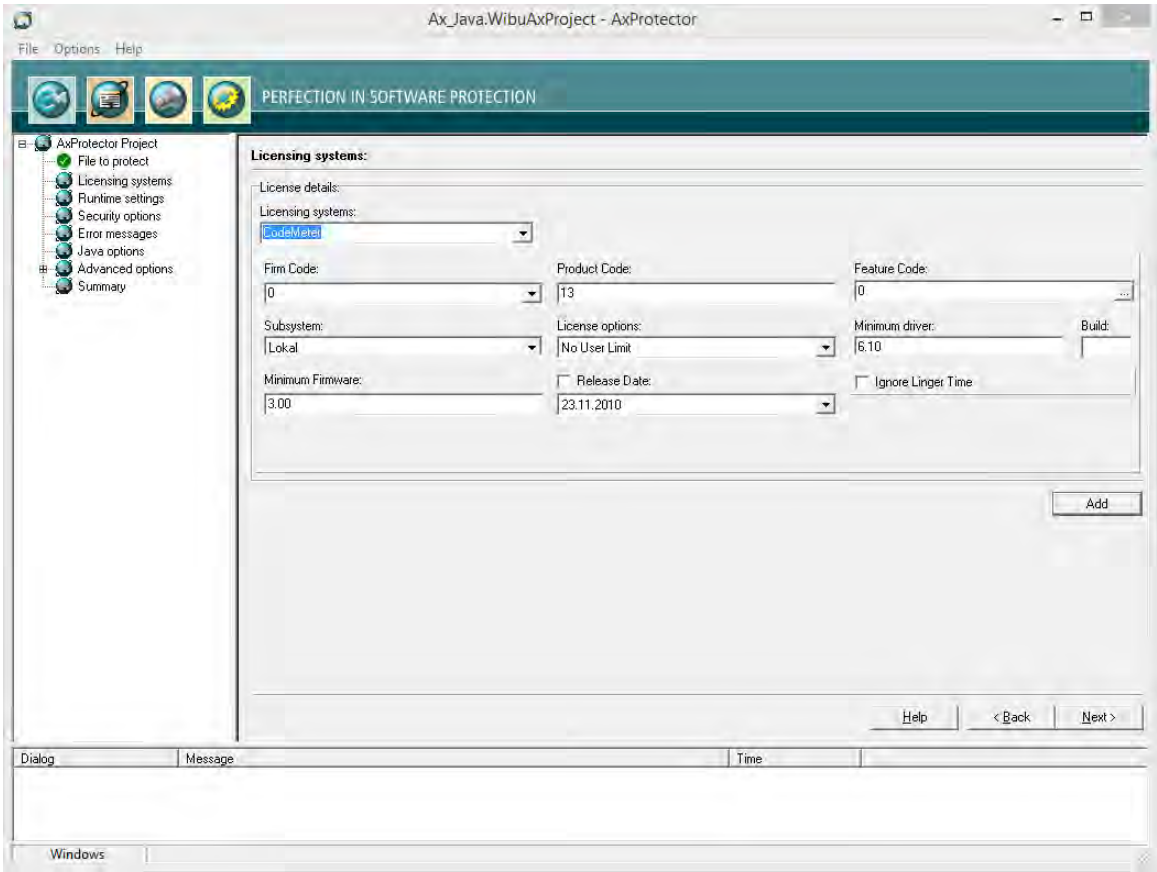


Figure 99: *AxProtector* - Java "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	Selecting the desired licensing system to be applied:

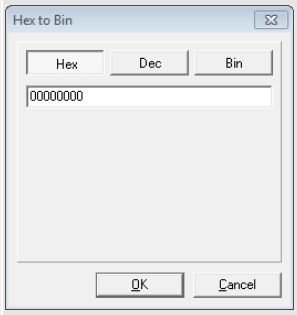
Entry	Description
`CodeMeter`	Applying the licensing system *CodeMeter*.
`IP Protection`	Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 283 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.
`WibuKey`	Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Firm Code

Specify the *Firm Code* to be used for encrypting the software.
As a registered licensor, you will be issued your own unique *Firm Code*(s).
The following default settings exist:

Firm Code CodeMeter Software Development Kit (SDK)	Licensing system
`6000010` Evaluation *Universal Firm Code*	*CodeMeter*
`10` *CmDongle* Evaluation *Firm Code*	*CmDongle*
`5010` *CmActLicense* Evaluation *Firm Code*	*CmActLicense*

Commandline option see here 264 .

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 100: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: ***Firm Codes*** (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses.

Element	Description
	Firm Codes (licensing system) / Version Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*): 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*): 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

7.4.5.2.1 Licensing Systems - Add licenses

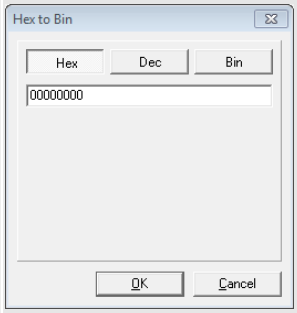
Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description

Licensing systems	Select from the dropdown control the desired licensing system. Available are the following entries:
	`CodeMeter`
	`WibuKey`
	For setting *WibuKey* options, see the separate "WibuKey Developer Guide".
	If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.
	In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist:

Element	Description
	<table><tr><th>*Firm Code CodeMeter* Software Development Kit (SDK)</th><th>Licensing system</th></tr><tr><td>`6000010` Evaluation *Universal Firm Code*</td><td>*CodeMeter*</td></tr><tr><td>`10` *CmDongle* Evaluation *Firm Code*</td><td>*CmDongle*</td></tr><tr><td>`5010` *CmActLicense* Evaluation *Firm Code*</td><td>*CmActLicense*</td></tr></table> Commandline option see here 264 .
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 101: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). <table><tr><th>Element</th><th>Description</th></tr><tr><td>Local</td><td>This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.</td></tr><tr><td>Network</td><td>This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.</td></tr><tr><td>Local - Network</td><td>This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.</td></tr><tr><td>Network - Local</td><td>This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.</td></tr></table>
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). <table><tr><th>Element</th><th>Description</th></tr><tr><td>Normal user limit</td><td>Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.</td></tr><tr><td>Station Share</td><td>Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.</td></tr><tr><td>WibuKey Compatibility Mode</td><td>Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.</td></tr><tr><td>Exclusive Mode</td><td>Here a protected application can be started only once on a PC.</td></tr><tr><td>No user limit</td><td>Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.</td></tr></table>
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Element	Description
	Firm Codes (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.5.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

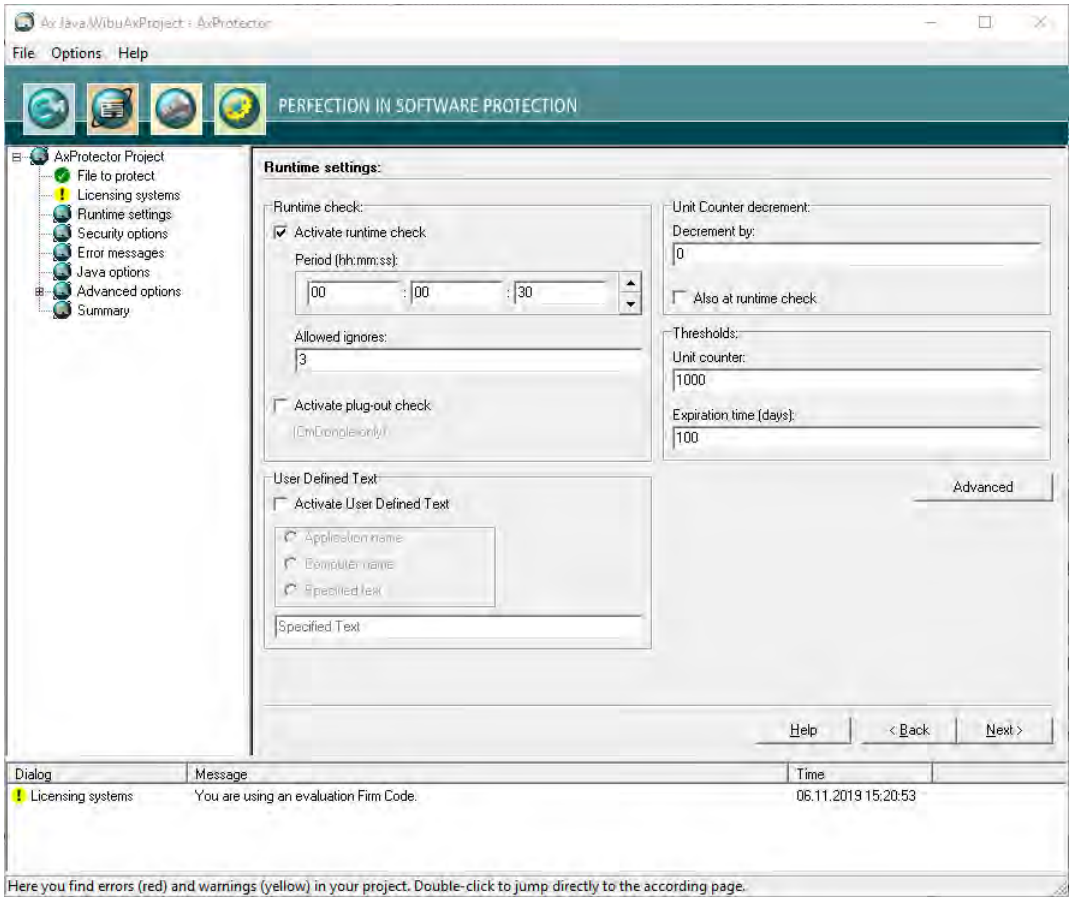


Figure 102: *AxProtector* - Java "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: hours: minutes: seconds.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the *Unit Counter* is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above, every 30 seconds (see the defined period) a set *Unit Counter* is decremented by a value of 1.
Also at Runtime Check	Decrements the *Unit Counter* also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime Check 157 " group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here 162 .

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here 277 .
Expiration Time (days)	When the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here 277 .

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here 279 .

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. Element / Description Application name — uses the application name. Computer name — uses the computer name. Specified text — uses the specified text in the field of the same name.

7.4.5.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

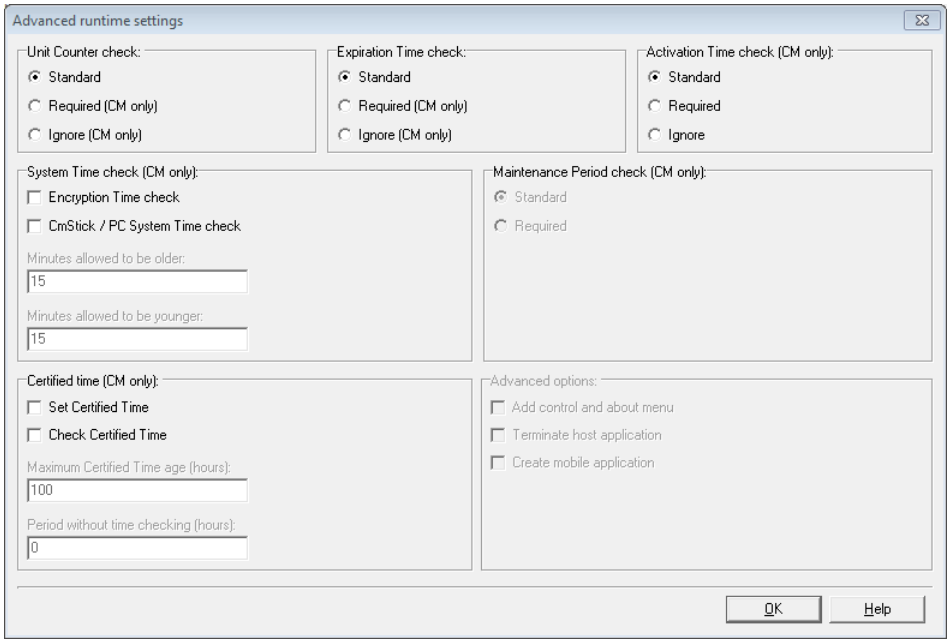


Figure 103: *AxProtector* - Java "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignored
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here 276).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches 0 (null), the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts when no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, when the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here 275).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time 357 is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, when the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 (commandline option see here 276).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems 151.

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only in the case a *Maintenance Period* exists. This corresponds to the default setting, even when on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmDongle* or the *CmActLicense* is connected with the computer. When the *CmContainer* is connected, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *Certified Time*. On updating the *Certified Time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here 270).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here 357..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmDongle*. The *certified time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check" you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time*.
Period without time checking (hours)	Specifies the period (in hours) when no check of the *Certified Time* certificate is taking place. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here 267).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only when the *CmContainer* System Time is newer than the encryption time. Requires at least *CodeMeter*® 4.10.

Element	Description
CmContainer / PC System Time check	When activated these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor , the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

7.4.5.4 Security Options

This input window lets you select from different mechanisms and methods for protecting your application.

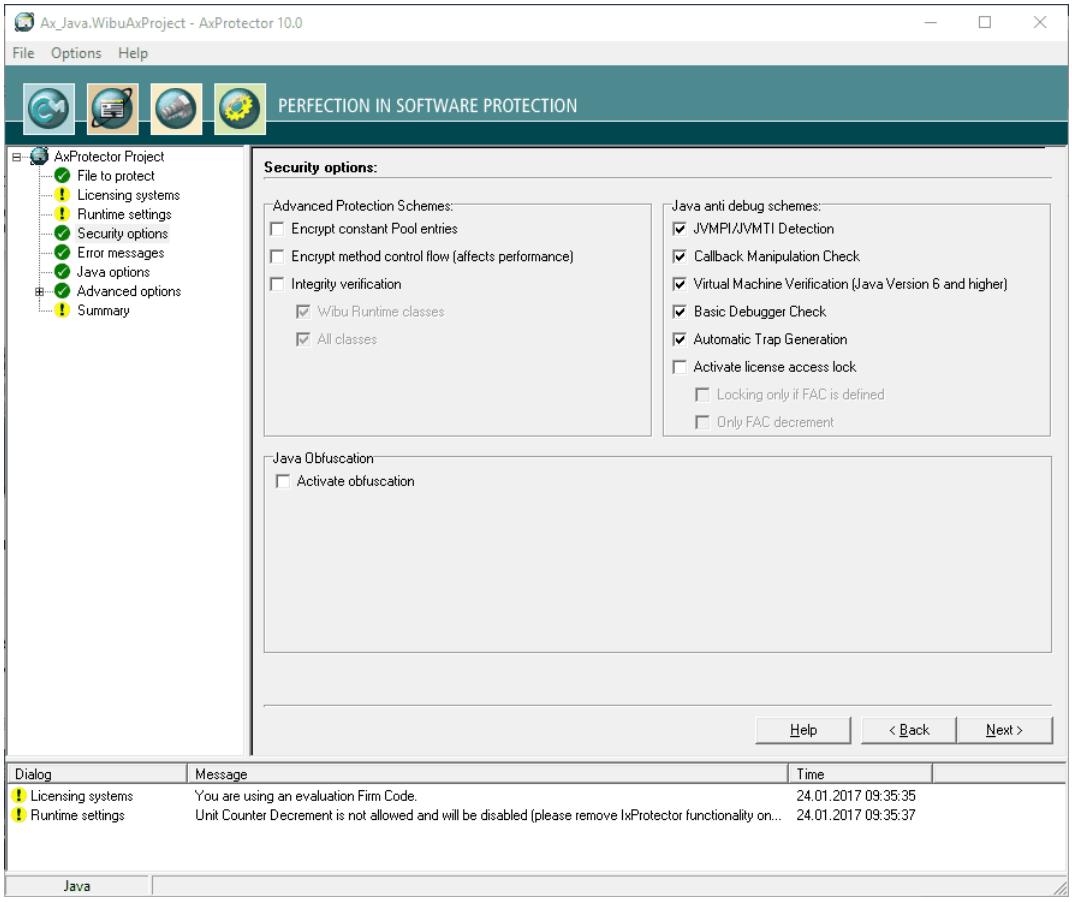


Figure 104: *AxProtector* - Java "Security Options"

Advanced Protection Schemes

The advanced protection schemes deeply intervene into your application.

Element	Description
Encrypt constant Pool entries	Encryption of selected values from the constants pool (commandline option see here 267).
Encrypt method control flow (affects performance)	Enrypts method calls (commandline option see here 267). This option must be combined with option `-ci` for method encryption.
Integrity verification	The protected application is checked for code integrity using asymmetric authentication mechanisms, if you check this box. **Wibu Runtime classes** are automatically checked. **All classes** can be checked for integrity.

Java anti debug schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here 269).

Element	Description
JVMPI / JVMTI Detection	Activating this checkbox starts the detection of the Java Virtual Machine Profiler Interface (JVMPI) and Java Virtual Machine Tool Interface (JVMTI). Using JVMPI the Java Virtual Machine is manipulable sending messages to the native code. In particular, the event `JVMPI_EVENT_CLASS_LOAD_HOOK` may be used to intercept the unaltered byte code of the class actually loaded. The activation of this option prevents this interception.
Callback Manipulation Check	Activating this checkbox protects against the manipulation of callback functions, i.e. functions which are transferred as parameters to other functions are checked.

Element	Description
Virtual Machine Verification (Java 6 and higher)	Activating this checkbox checks for the correct Java Virtual Machine runtime environment for Java 6 and higher.
Basic Debugger Check	The 'Basic 269 Debugger Check', checks to see if a debugger is attached to your application. If a debugger is found, your application will not be started or exited.
Automatic Trap Generation	Automatically inserts hacker traps into the protected assembly. Automatic traps are generated for methods only, if they have been encrypted using IxProtector license lists. Only those methods are transformed into new classes (commandline option see here 279).
Activate license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. **Option** / **Description** Locking only if FAC defined: It is checked whether a prepared locking of the *Firm Access Counter* (FAC) is programmed. If a locking is prepared, the Firm Items is locked. Only FAC decrement: The *Firm Access Counter* (FAC) is decremented by the value of 1.

Java Obfuscation

The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information.

Element	Description
Activate obfuscation	The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information (commandline option see here 274). If activated the following selectable options display: ☑ Activate obfuscation ☑ Class names ☑ Private elements ☐ Ignore detection of reflection ☑ Method names ☑ Inner elements ☐ Print name mapping ☑ Local variable names ☑ Protected elements ☑ Field names ☐ Public elements ☐ Package Names
	Elements to be obfuscated comprise: Class names, Method names, Local variable names, Field names, Package names, Private elements, Inner elements, Protected elements, and Public elements,
Ignore detection of reflection	Ignores detection of reflection. • If the class name is known at encryption, e.g. `Class.forName("HelloWorld")`, automatically a customization of the class name is performed. Please note, that in this case methods and fields of the class are excluded from obfuscation. • If the class name is not known at encryption, e.g. `Class.forName(getClassName())`, an exception is thrown together with the specification where the reflection call has been found and what can be done, e.g. `Replace reflection, use constant class names, force the obfuscation or disable obfuscation`. • For reflection calls, such as, `getMethod('aMethod'), getField('aField')`, you must make sure that `'aMethod'` and `'aField'` are not obfuscated.
Print name mapping	At encryption an obfuscation mapping is issued to the console output.

7.4.5.5 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a `User Message Class` with a separate error display is used, or whether you use default error message windows.

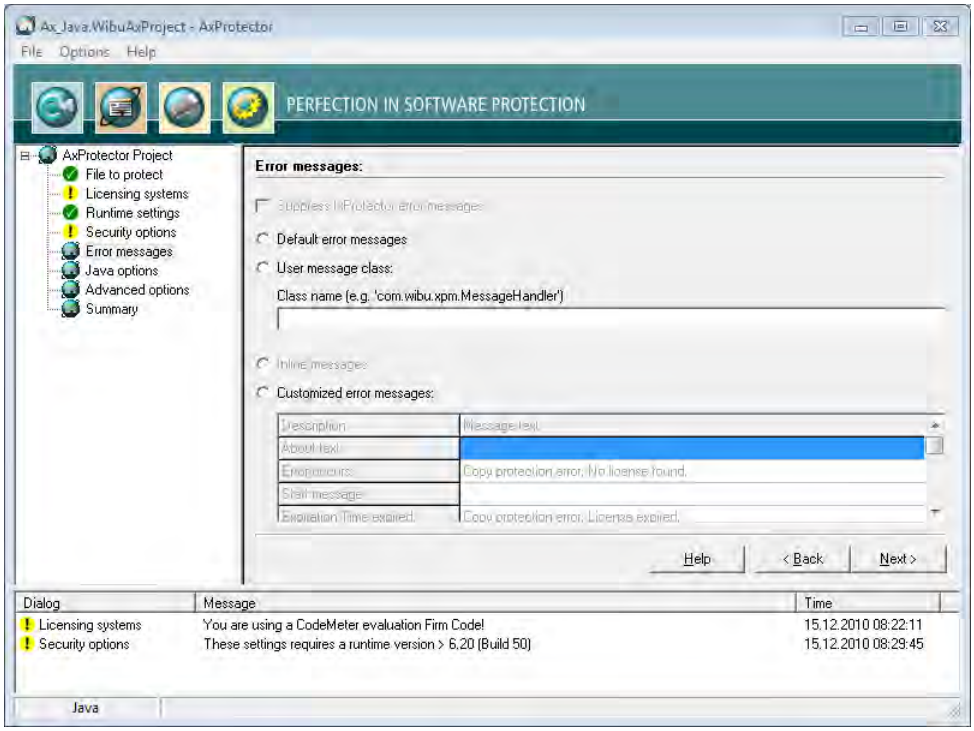


Figure 105: *AxProtector* - Java "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
User Message Class	Activates the use of a User Message Class.
Class name	Specify here the file name without path information and extension.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.5.6 Java Options

This input window lets you determine some parameters for the configuration of the Java runtime environment.

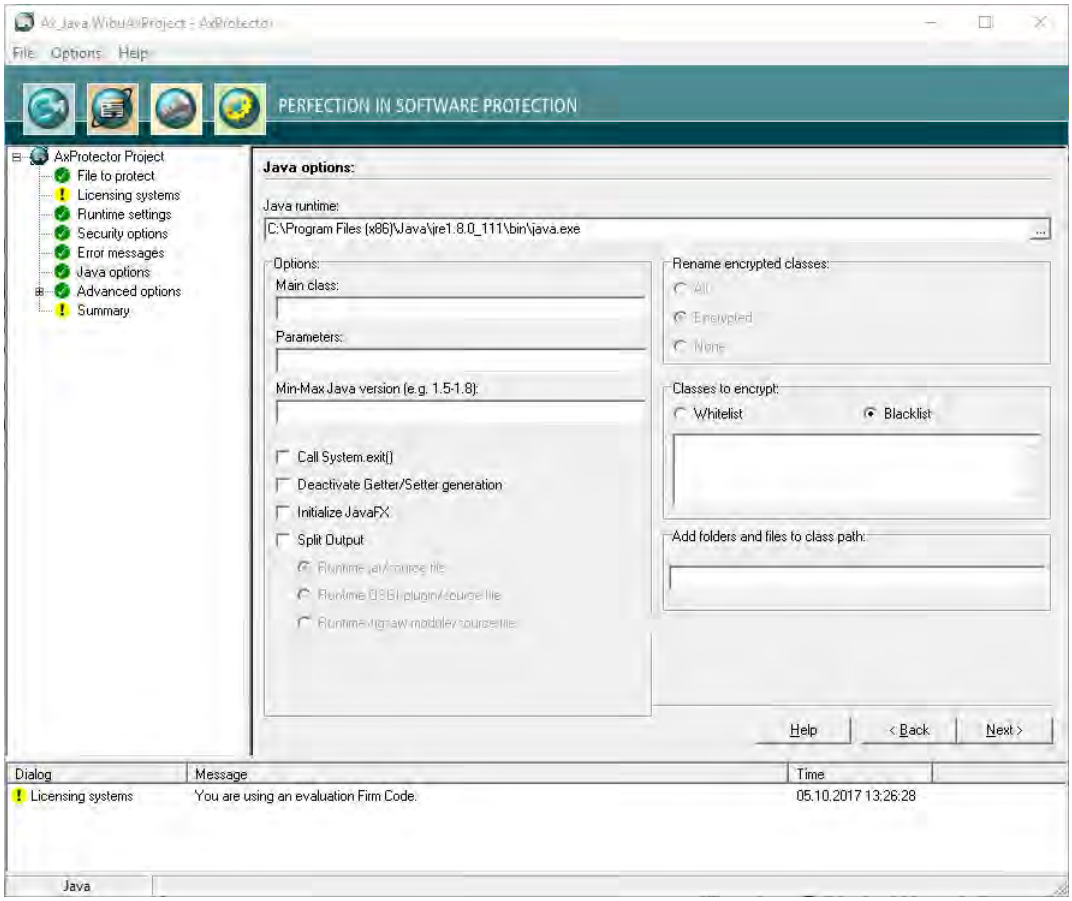


Figure 106: *AxProtector* - Java "Java Settings" Java Runtime (java.exe)

Element	Description
Java Runtime (java.exe)	Using the **"..."** button specify the `java.exe` file of the installed runtime environment.
Main class	Enter here the name of the Java main class (commandline option see here 284).
Parameters	Define here the parameters for calling the Java main class (commandline option see here 282).
Min-Max Java Version	Enter here the required minimum Java version (commandline option see here 282). When the check fails, a respective error message is issued. This ensures already at start of the protected application that the functionality of your application requires is guaranteed.
Call System.exit()	Activate this option to exit the application by the call of `System.exit()` after return to the Java main class. This ensures that in the case errors occur, the protected application correctly and completely shuts down. Even when the error occurred outside the Java main class (commandline option see here 284).
Deactivate Getter / Setter generation	Deactivating of the default generation of getter and setter methods (commandline option see here 283).
Initialize JavaFX	Initializes JavaFX. This is required for encrypting some JavaFX applications (commandline option see here 283).
Split Output	**Runtime jar / source file** By default, runtime classes are saved to the separate `WibuXpm4Jruntime.jar` file (commandline option see here 284). Swapping the Wibu `ClassLoader` to a separate file increases performance of the protected application. Then even in the case of multiple encrypted classes, the Wibu `ClassLoader` will be only one-time loaded. **Runtime OSGI plugin / source file** The WIBU runtime classes are created in form of a OSGI bundle named `WibuXpm4JRuntimePlugin.jar`. The dependencies of the encrypted source *.jar from this OSGI bundle are automatically created at encryption (commandline option see here 284). **Runtime jigsaw module / source file** Modular jar files created using Java 9 are encrypted. Then on encryption a modular Wibu Runtime jar file with the name `com.wibu.xpm.jar` is created. Dependencies of the protectee to this Wibu Runtime jar file are automatically added to the `module-info.class` of the protectee (commandline option see here 284).

Element	Description
Rename encrypted classes	This group allows you to determine the classes which classes will be renamed, and loaded into the Wibu `ClassLoader` (commandline option see here 283). Please note that this group can be edited only, if *IxProtector* is not activated. For all class-related settings, the classes are renamed, and follow the pattern: `<MyClass>.class.wibu`. **Element** / **Description** All: Activate this option to rename all existing classes. Encrypted: Activate this option to rename encrypted classes only. None: Activate this option to rename no classes. When you rename encrypted classes only, only these classes are loaded by the Wibu `ClassLoader`. This improves the performance of the application. When you rename all classes, the security is increased at a small margin but eventually the performance of the protected application is negatively affected.
Classes to encrypt	This allows you to assign white or black list to classes (commandline option see here 284). **Element** / **Description** Whitelist: All classes referred to in the whitelist will be encrypted. This whitelist is saved to the jar-archive as an unencrypted text file `com/wibu/xpm/encrypted`. Blacklist: All classes referred to in the blacklist will not be encrypted. `AxProtector Syntax: -JL[W|B]:<whitelist|blacklist>` Using these list give you direct bearing on the classes to be encrypted. For example, eventually it does not make sense to protect classes of third party providers, and stress the application performance. For the output of error messages at the runtime of the encrypted Java application you may use the error class `com.wibu.xpm.MessageHandler`.
Add folders and files to class path	Adds folder and file information to the Java class path to allow resolving dependencies of the Java application (commandline option see here 282).

7.4.5.7 Advanced Options

This input window lets you set further encryption options.

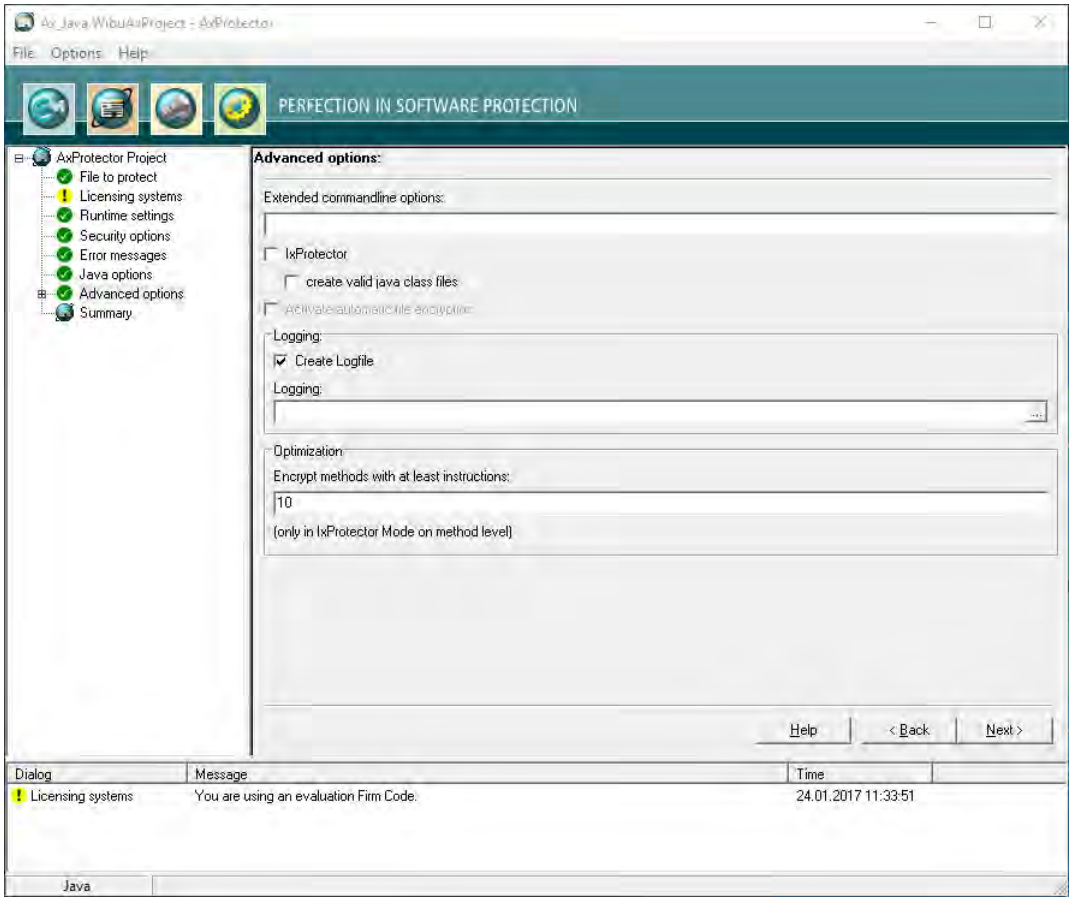


Figure 107: *AxProtector* - Java "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289 . (commandline option see here 274).
create valid java class files	Creates machine-readable class files (commandline option see here 283).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.
Optimization	For an optimized performance specify here the minimum number of instructions a method must at least have to be encrypted. The default setting is 10 instructions. This way you are able to exclude methods from encryption which have less instructions than the number of instructions you specify here. By setting a value of 0 this feature is deactivated. Commandline option see here 282 .

7.4.5.7.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 . License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

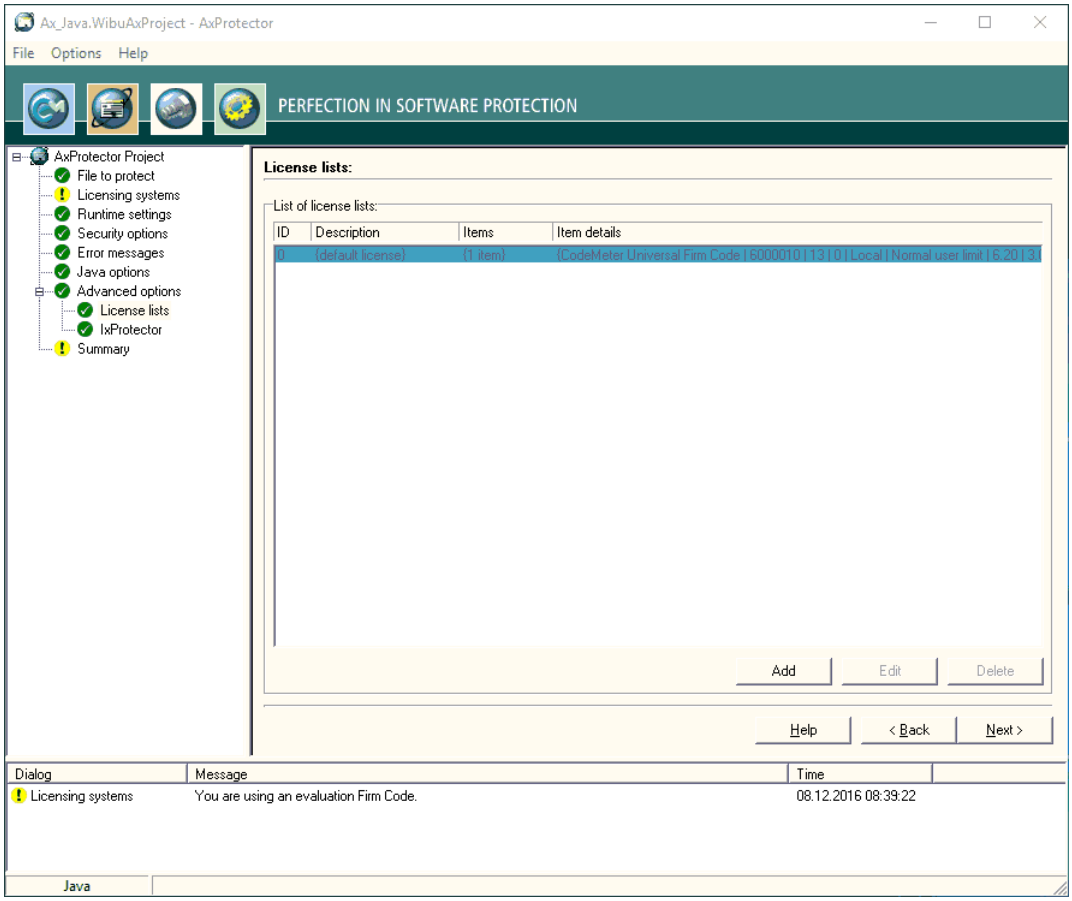
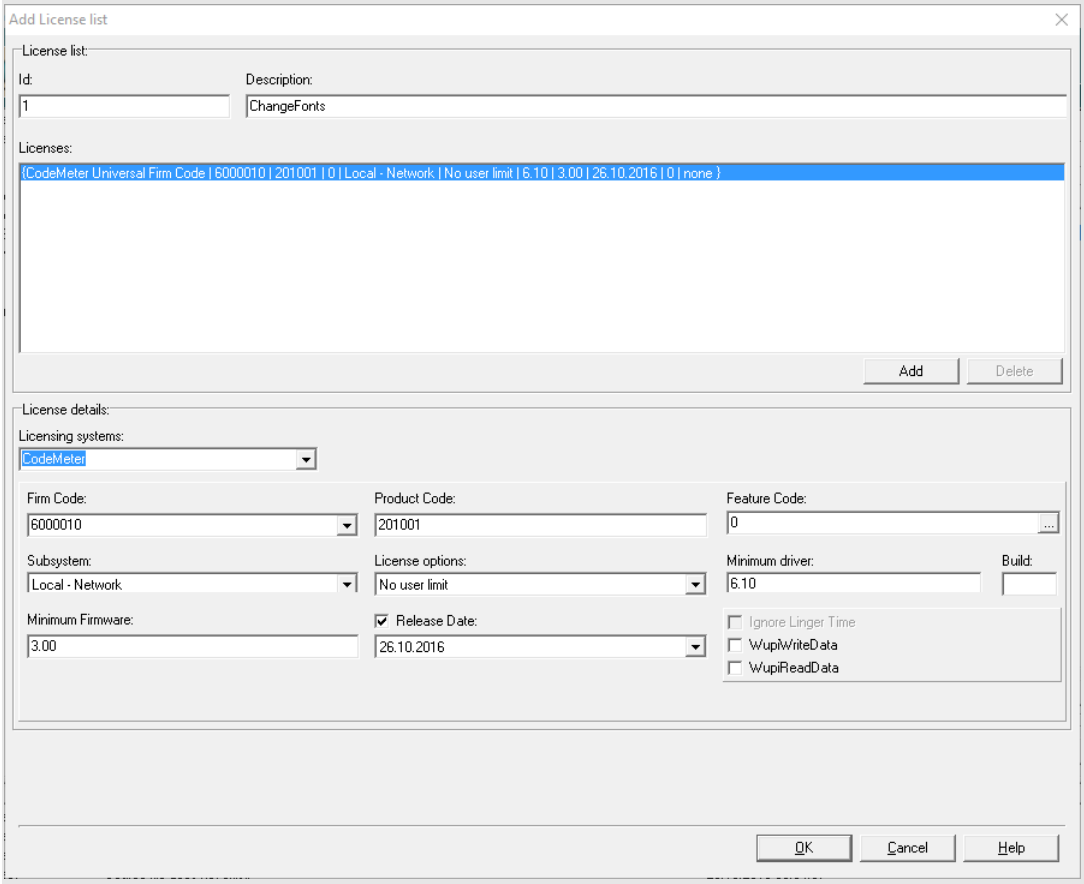


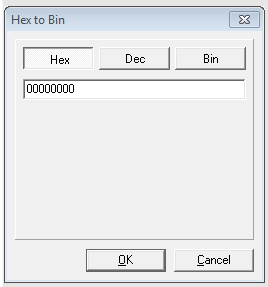
Figure 108: *AxProtector* Java - "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 109: *AxProtector* Java - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 283. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. If `LicenseList 0` contains an "*IP Protection*" license and the `-jip` option is not explicitly set, all operating system-specific runtime components, i.e.default DLLs for `win-32`, `win-64`, `mac-64`, `lin-32` and `lin-64` are added to the JAR archive (equivalent to the `-jip:std` option). However, you can use the field 165 "Extended Commandline Options" to add further DLLs. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) — **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle

Element	Description
	Firm Code CodeMeter Software Development Kit (SDK) | Licensing system `5010` *CmActLicense* Evaluation *Firm Code* | CmActLicense Commandline option see here 264.
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264.
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 110: *Feature Map* Input Commandline option see here 264.
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). Element | Description Local | This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network | This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network | This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local | This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). Element | Description Normal user limit | Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share | Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode | Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode | Here a protected application can be started only once on a PC. No user limit | Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) | Version `6000010; >= 6.000.000` (*Universal Firm Code*) | 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) | 4.20 When setting the minimum driver version to 3.20 the session handling for terminal

Element	Description
	Firm Codes (licensing system) / Version servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) — 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) — 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55 ". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

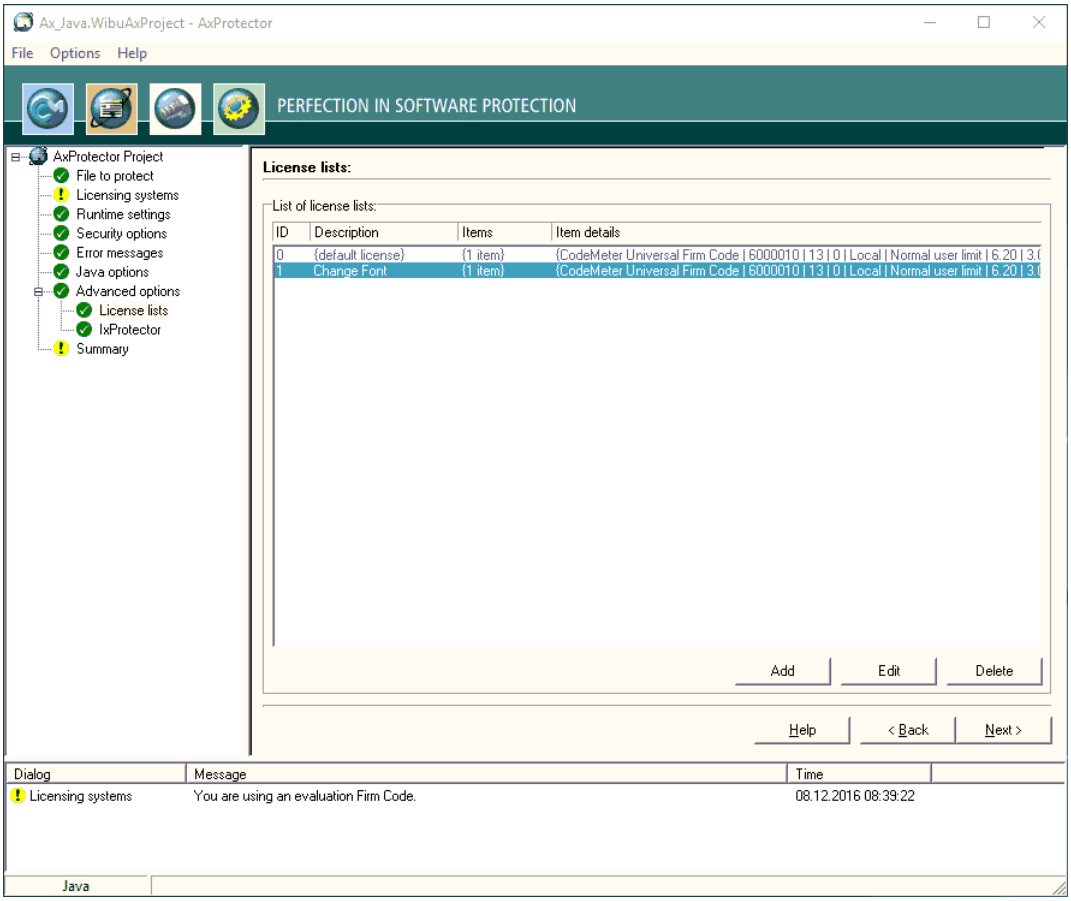


Figure 111: *AxProtector* Java - "Completed License Lists"

7.4.5.7.2 IxProtector

Using this menu item allows you to separately define single encryption types for single elements.

In the case you activated the checkbox **"IxProtector"** in the menu item **"Advanced options"** the source application file is loaded and displayed in a tree view making available all packages, classes, and methods.

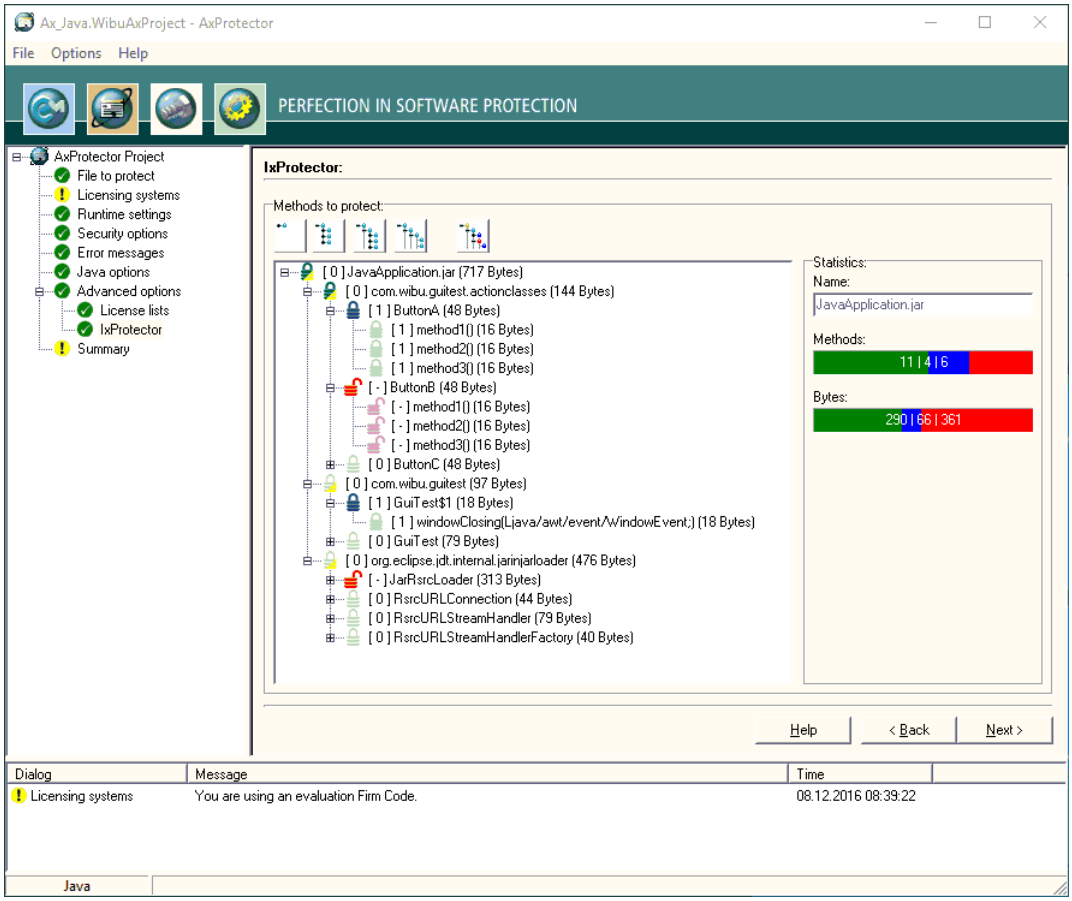


Figure 112: *AxProtector* Java - "IxProtector"

Click the different buttons in the upper "IxProtector" area to select from different views.

Views

Buttons	Description
	Closes all levels of the tree structure.
	Expands the package level of the application file.
	Expands the class level of the application file.
	Expands the method level of the application file.
	Expands all parent levels of the application file. In this view see all levels where modifications have been made.

The area "Statistics" on the right shows you more encryption details depending on the selection you have made for the tree view.

Element	Description
Name	This field refers to the name of the element you have marked in the tree view.
Methods	Using different colors the bar 'Methods' shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted methods for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.
Bytes	Using different colors the bar 'Instructions' also shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted instructions for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using AxProtector).
	Blue: Shows that the method will be encrypted using IxProtector.
	Red: Shows that the method in not encrypted.

You also have the option to separately assign the protection technologies *AxProtector* and *IxProtector* to single elements, or exclude single elements from encrypting. To assign a protection technology by using the secondary menu, please proceed as follows:

1. In the left tree view, select the favored element (package, class, or method).
2. Click the right mouse button.
   The secondary menu opens.
3. Assign the favored encryption types by using symbols.

The License List IDs you are prompted are automatically transferred from the entries you added to the license list.

Symbol	Description
	Excludes the selected element from encryption.
	Encrypts the selected element using *AxProtector* (License List ID with a value of 0, i.e. default license).
	Encrypts the selected element using *IxProtector* (License List ID with a value unequal to 0, i.e. according to existing license list entries).
	This icon marks methods that are excluded from encryption due to the number of instructions included. The instruction threshold can be set on the page 'Advanced Options' in the group Optimizing
EntryPoint	Sets entry point. This allows the direct external call of a class / method (e.g. as library).
Check Integrity	Checks for code integrity. Available only, if respective option 160 is checked.
Apply Obfuscation	Applies obfuscation. Available only, if respective option 161 is checked.

The modifications you made instantly display in the left area.

7.4.5.8 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

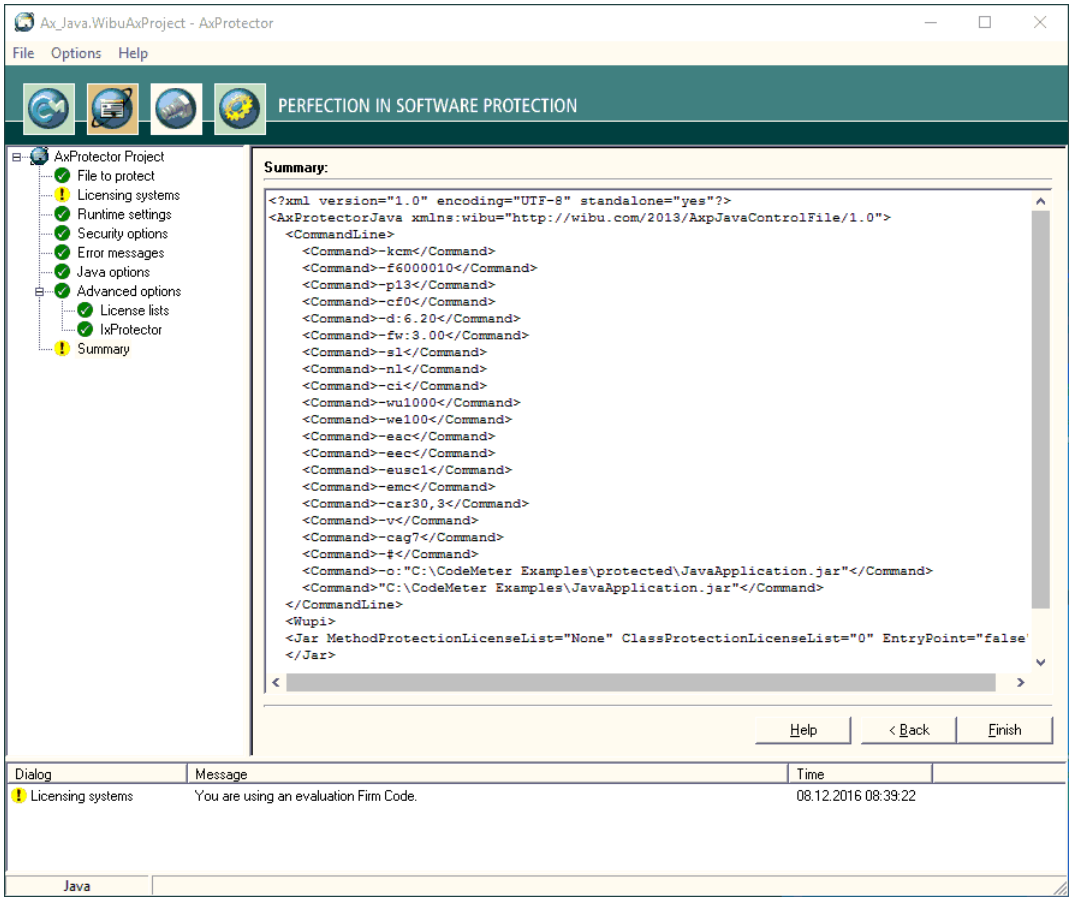


Figure 113: *AxProtector* - Java "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

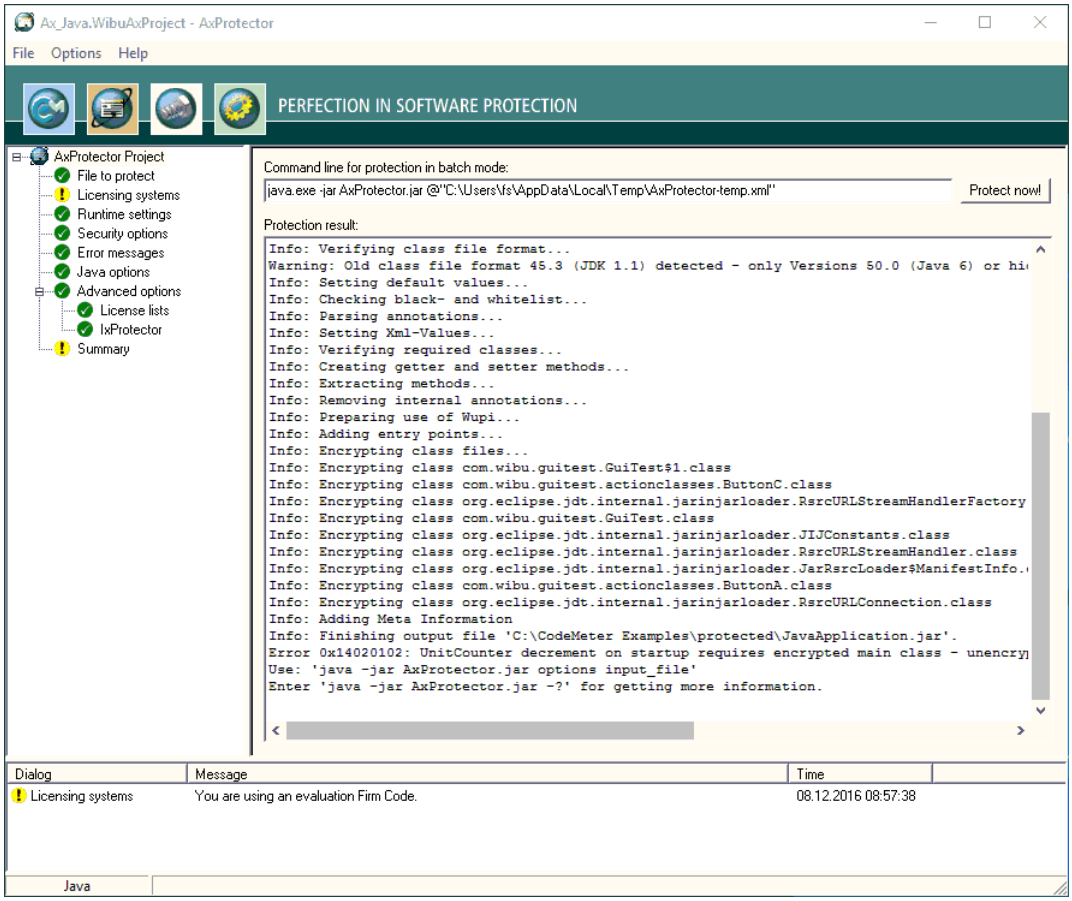


Figure 114: *AxProtector* - Java "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.4.6 Linux Application or Shared Object

Using this *AxProtector* project type works analog to the previous project types. For a detaiiled view and description of the singe navigation menu items an online variant of this guide is available for download at the Wibu-System website (www.wibu.com/en/manuals-guides.html). A complete version is also included in the help documentation you find in the *CodeMeter* SDK.

This project type covers encrypting executables in the standard binary format for executable programs (ELF, Executable and Linking Format) and program libraries (shared objects `*.so`).The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
Linux Application or Shared Object	AxProtector Linux	√	Windows commandline [263] In a separate commandline for Linux, running on Linux operating systems, you are also able to insert encryption parameter [195].

The following menu items are available in the navigation windows:

- File to protect [175]
- Licensing Systems [175]
- Runtime Settings [181]
- Security Options [184]
- Error Messages [187]
- Advanced Options [188]
    - License Lists [188]

- IxProtector 192
- Summary

7.4.6.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

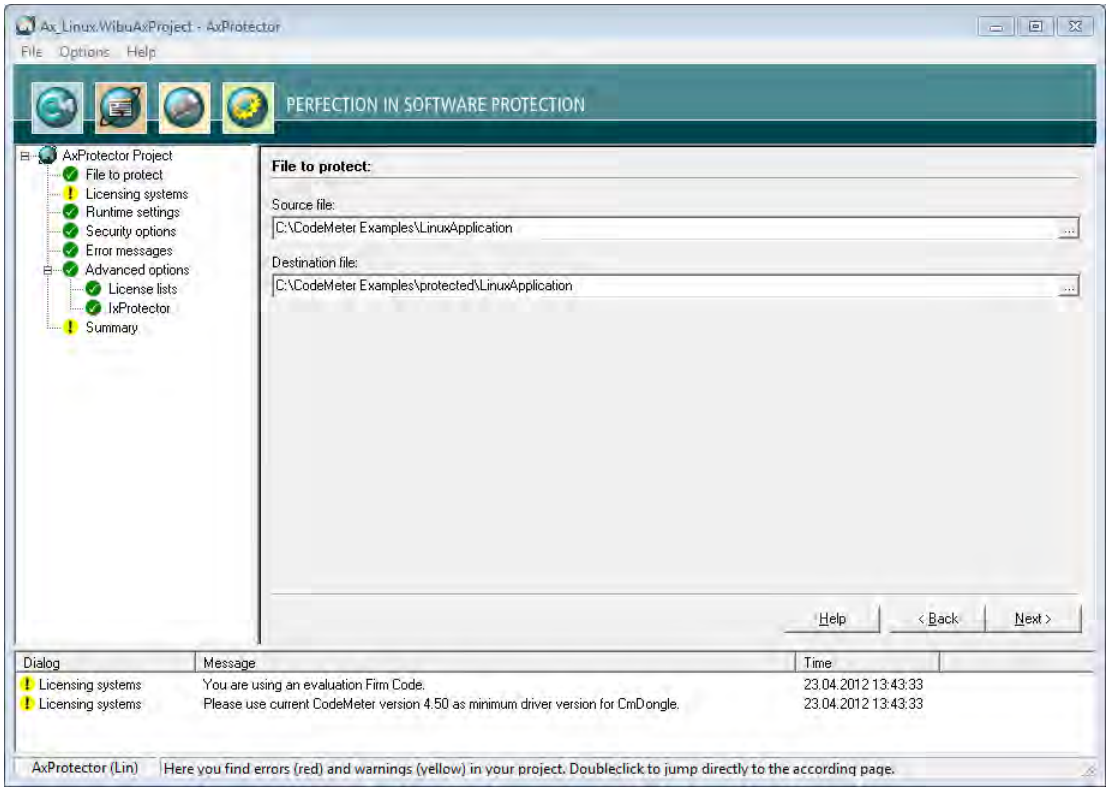


Figure 115: *AxProtector* - Linux "File to Protect"

File to Protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.4.6.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

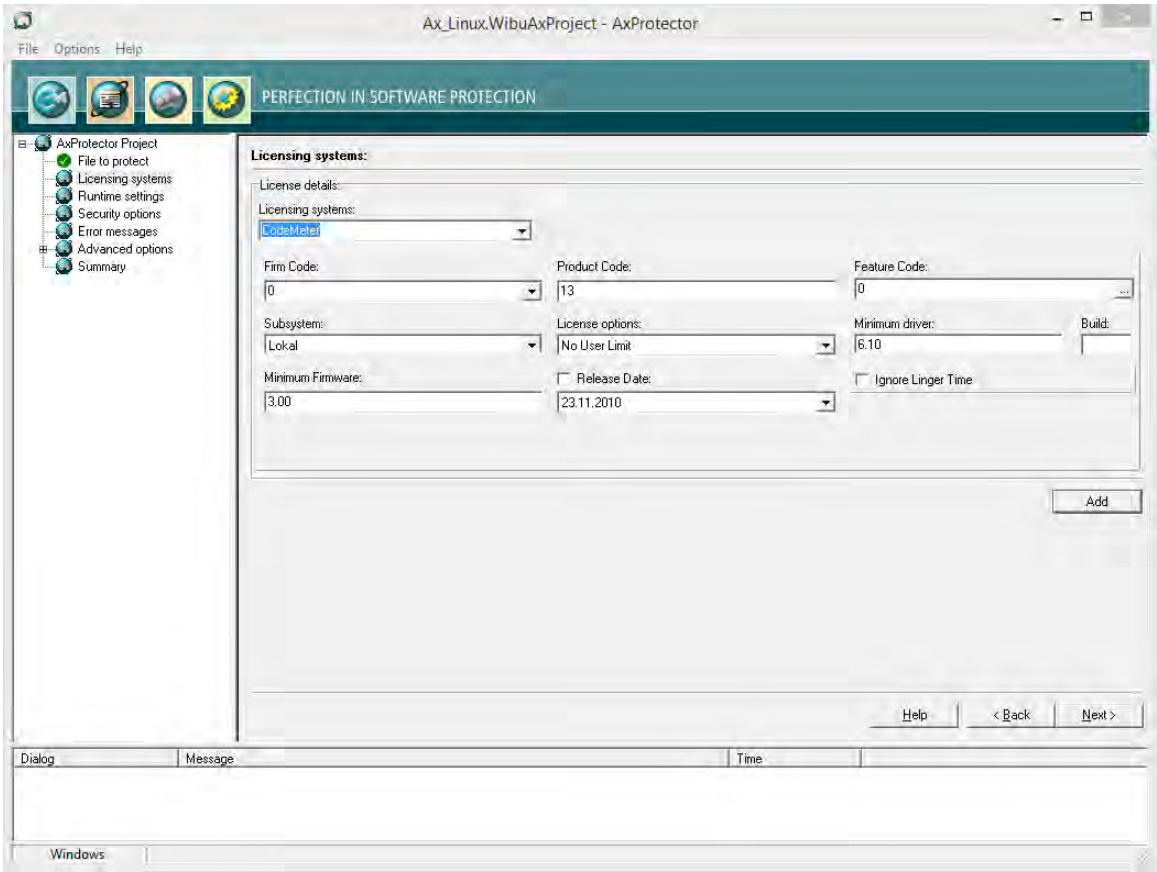


Figure 116: *AxProtector* - Linux "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

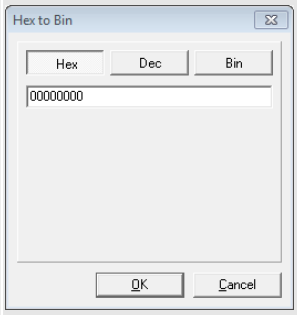
Element	Description
Licensing systems	Selecting the desired licensing system to be applied:

Entry	Description
`CodeMeter`	Applying the licensing system *CodeMeter*.
`IP Protection`	Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.
`WibuKey`	Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Element	Description
Firm Code	Specify the ***Firm Code*** to be used for encrypting the software. As a registered licensor, you will be issued your own unique ***Firm Code***(s). The following default settings exist:

Firm Code CodeMeter Software Development Kit (SDK)	Licensing system
`6000010` Evaluation ***Universal Firm Code***	*CodeMeter*
`10` *CmDongle* Evaluation ***Firm Code***	*CmDongle*
`5010` *CmActLicense* Evaluation ***Firm Code***	*CmActLicense*

Commandline option see here 264 .

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 117: *AxProtector - Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Firm Codes (licensing system)	Version
`6000010; >= 6.000.000` (*Universal Firm Code*)	6.10 This supports the License Transfer feature.
`10, 100.000-4.999.999` (*CmDongle Firm Code*)	4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses.

Element	Description
	Firm Codes (licensing system) / Version Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*): 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*): 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*): 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

7.4.6.2.1 Licensing Systems - Add licenses

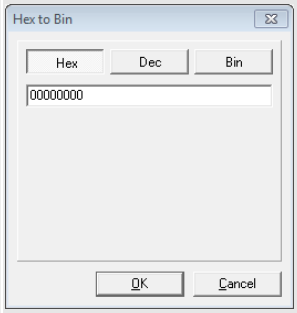
Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description

Licensing systems	Select from the dropdown control the desired licensing system. Available are the following entries:
	`CodeMeter`
	`WibuKey`
	For setting *WibuKey* options, see the separate "WibuKey Developer Guide".
	If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.
	In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist:

Element	Description
	Firm Code CodeMeter Software Development Kit (SDK) / Licensing system `6000010` Evaluation *Universal Firm Code* / *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* / *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* / *CmActLicense* Commandline option see here 264 .
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 118: *AxProtector - Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265).

Element	Description
Local	This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM.
Network	This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server.
Local - Network	This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network.
Network - Local	This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.

License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
Normal user limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
WibuKey Compatibility Mode	Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.
Exclusive Mode	Here a protected application can be started only once on a PC.
No user limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist:

Element	Description
	Firm Codes (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) — 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*) — 1.14 In order to use the *Product Item* Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

7.4.6.3 Runtime Settings

This input window lets you define the application's runtime settings, e.g. license checks for *CmContainer*, issue warnings, etc.

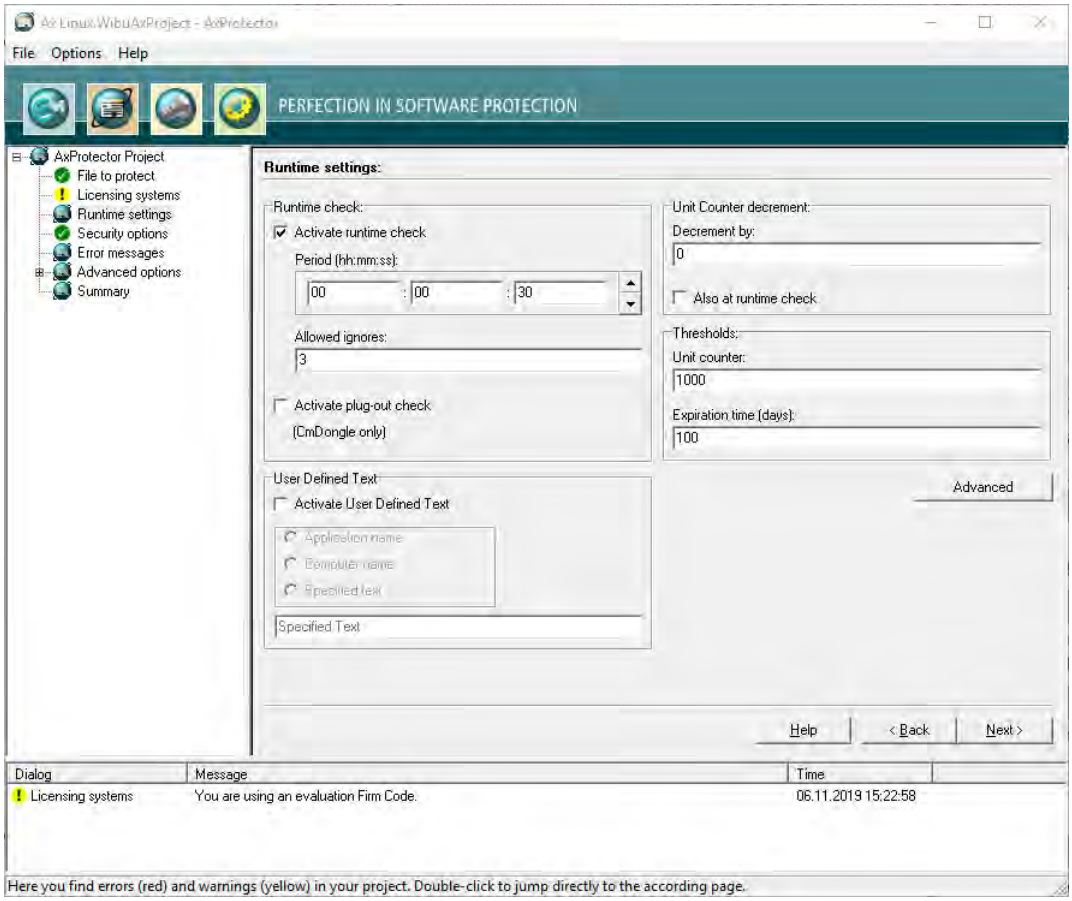


Figure 119: *AxProtector* - macOS "Runtime Settings"

Runtime Check

In this group you define whether and how often the protected application checks the license at runtime.

Element	Description
Activate Runtime Check	Activates or deactivates the check at runtime of the protected application. Commandline options see here 270 .
Period	Defines the period between two checks. You specify this time interval in the format: `hours: minutes: seconds`.
Max. Allowed Ignores	Defines how often the end-user is able to ignore a failed check If the connection to a *CmContainer* 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.
Activate Plug-out Check (only CmDongle)	This option closes the protected application if the *CmDongle* is removed while the application is running. Immediately, an error message is issued. This option is valid for *CmDongle* only. Commandline option see here 267 .

Unit Counter Decrement

Decrementing an Unit Counter can serves to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see here 276).

Element	Description
Decrement by	Defines the value by which the *Unit Counter* is decremented. This option causes a decrement of the counter when the protected application starts. If the "Also at Runtime Check" option is activated and the specifications are set as shown in the figure above, every 30 seconds (see the defined period) a set *Unit Counter* is decremented by a value of 1.
Also at Runtime Check	Decrements the *Unit Counter* also at runtime of the protected application. This option works only when the "Also at Runtime Check" option in the "Runtime" group is activated.

Thresholds

In this group you define when a message is issued to give information on the validity of a license.

For customizing the messages texts see here[137] .

Element	Description
Unit Counter	If the defined threshold falls short, a warning message is issued. Commandline option see here[277] .
Expiration Time (days)	When the specified *Expiration Time* (in days) is achieved within the defined threshold, a warning message is issued. Commandline option see here[277] .

User Defined Text

In this group you can use a User Defined Text, which is then stored as text entries in the *AxEngine* (***CmAccess***) license access structure. These entries then overwrite the texts that are set by a Message DLL. For the commandline option see here[279] .

Element	Description
Activate User Defined Text	Activates or deactivates the use of User Defined Text. The following text entries can be used. **Element** / **Description** Application name — uses the application name. Computer name — uses the computer name. Specified text — uses the specified text in the field of the same name.

7.4.6.3.1 Advanced Runtime Settings

This input window lets you define further settings at the runtime of an encrypted application.

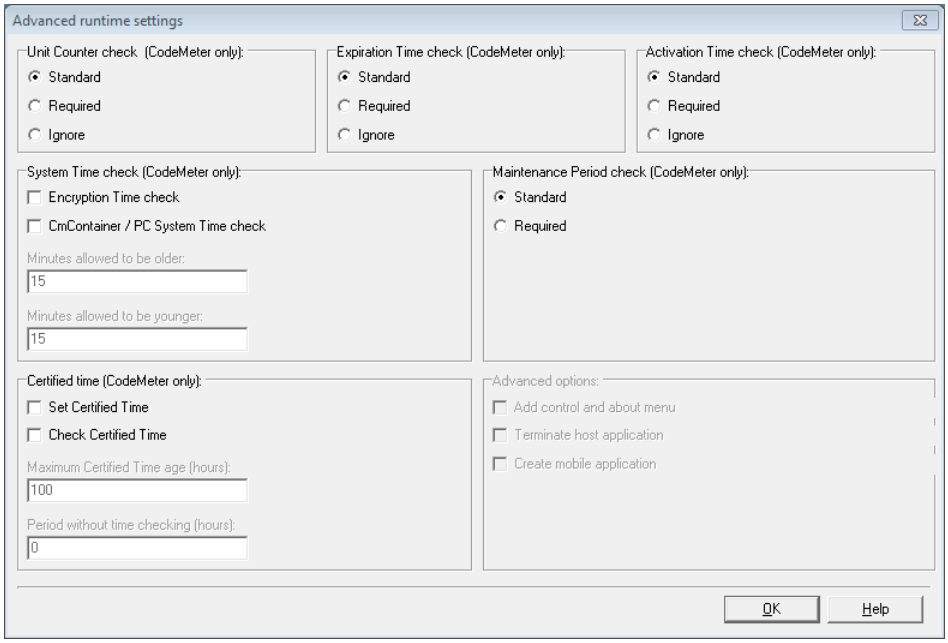


Figure 120: *AxProtector* - Linux "Advanced Runtime Settings"

For checking the options *Unit Counter*, Expiration Time, Activation Time defined in a license the following handling is valid.

Status	Standard	Required	Ignored
= 0	X	X	√
< > 0	√	√	√
not specified	√	√	√

Unit Counter

Defines the handling of a *Unit Counter* set in a license (commandline option see here[276]).

Element	Description
Standard	Decrements at runtime and/or start time an existing *Unit Counter* entry in a license by the value defined on the previous page. If the *Unit Counter* reaches 0 (null) the encrypted application does not start.
Required	A Unit Counter entry < > 0 in a license is required. Without such an entry the encrypted application does not start at all.
Ignore	An existing *Unit Counter* entry in the license is ignored. The application does not decrement the *Unit Counter*. The application will start with a *Unit Counter* entry set to 0.

Expiration Time

Defines the handling of an *Expiration Time* set in a license (commandline option see here[275]).

Element	Description
Standard	Checks for an existing *Expiration Time* entry in a license. However, the application also starts when no Expiration Time entry exists, or the current date precedes the Expiration Time.
Required	An *Expiration Time* entry in a license is required. Without such an entry the encrypted application does not start.
Ignore	An existing *Expiration Time* entry in a license is ignored. Also, when the current date exceeds the Expiration Time.

Activation Time

Defines the handling of an Activation Time set in a license (commandline option see here[275]).

Element	Description
Standard	Checks for an existing Activation Time entry in a license. However, the application also starts when no Activation Time exists, or the certified time[357] is later than the Activation Time.
Required	An Activation Time entry in a license is required. Without such an entry the encrypted application does not start. Please note that in that case, an Internet connection for getting the certified time is also required.
Ignore	An existing Activation Time entry in a license is ignored. Also, when the current date precedes the Activation Time.

Maintenance Period

Defines the handling of a *Maintenance Period* saved to the license. 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 performed if the date is within the defined period (commandline option see here[276]).

> The option is available only, if you activated the checkbox *Release Date* on the page "Licensing systems[178].

Two checking options exist:

Element	Description
Standard	At runtime of the protected application a *Release Date* check is performed only if a *Maintenance Period* exists. This corresponds to the default setting, even when on the page "Licensing systems" the checkbox *Release Date* has not been activated.
Required	At runtime of the protected application a *Release Date* check is mandatory performed. The PIO *Maintenance Period* must exist.

Certified Time

Each *CmContainer* has an integrated clock which advances when the *CmContainer* is connected with the computer or activated. If the *CmContainer* is connected or activated, the clock's time synchronizes forward. Otherwise, the time last saved applies.

If desired, the *Certified Time* can be updated by synchronizing with any *CodeMeter*® Time Server. The Time Servers are spread globally by Wibu-Systems and provide a *certified time*. On updating the *Certified Time* the internal *CmContainer* time is synchronized and updated as well (commandline option see here[270]).

> For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see here[357]..

Element	Description
Set Certified Time	This option attempts to update the *Certified Time* in a *CmDongle*. The *Certified Time* is requested from the Time Server. This option requires a connection to the Internet.
Check Certified Time	This option checks to see if the *Certified Time* is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.
Maximum Certified Time Age (hours)	If you select the option "Check", you are able to define here the Maximum Certified Time Age in hours. The age is calculated by the difference between the running System Time and the *Certified Time*.
Period without time checking (hours)	Specifies the period (in hours) when no check of the *Certified Time* certificate is performed. If this period is not reached, a check is not performed. If the *Certified Time* certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the *Certified Time* certificate is performed. If this is not successful, however, the application continues running until the 'Maximum Certified Time Age' is reached. Not until this happens, is an update of the *Certified Time* certificate required.

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see here[267]).

Element	Description
Encryption Time check	This option saves the time when the encryption takes place (PC Time) in the protected application. Then the application runs on the user PC only when the *CmContainer* System Time is newer than the encryption time.

Element	Description
	Requires at least *CodeMeter*® 4.10.
CmContainer / PC System Time check	When activated these options define a time corridor in which a difference between *CmContainer* System Time and PC Time is allowed. If the PC Time does not fall into this defined time corridor, the protected application will not run on the user PC.
Minutes to be allowed older	States in minutes how much the PC Time is allowed to be older than the *CmContainer* System Time.
Minutes to be allowed younger	States in minutes how much PC Time is allowed to be younger than the *CmContainer* System Time.

7.4.6.4 Security Options

This input window lets you select from different mechanisms and methods for protecting your application. You are able to scale the degree of security for yourself, for example, search intensity for debugger or whether a *CmContainer* is locked.

When the options you set here turn out to be incompatible with your protected application, you are also able to separately deactivate single security options.

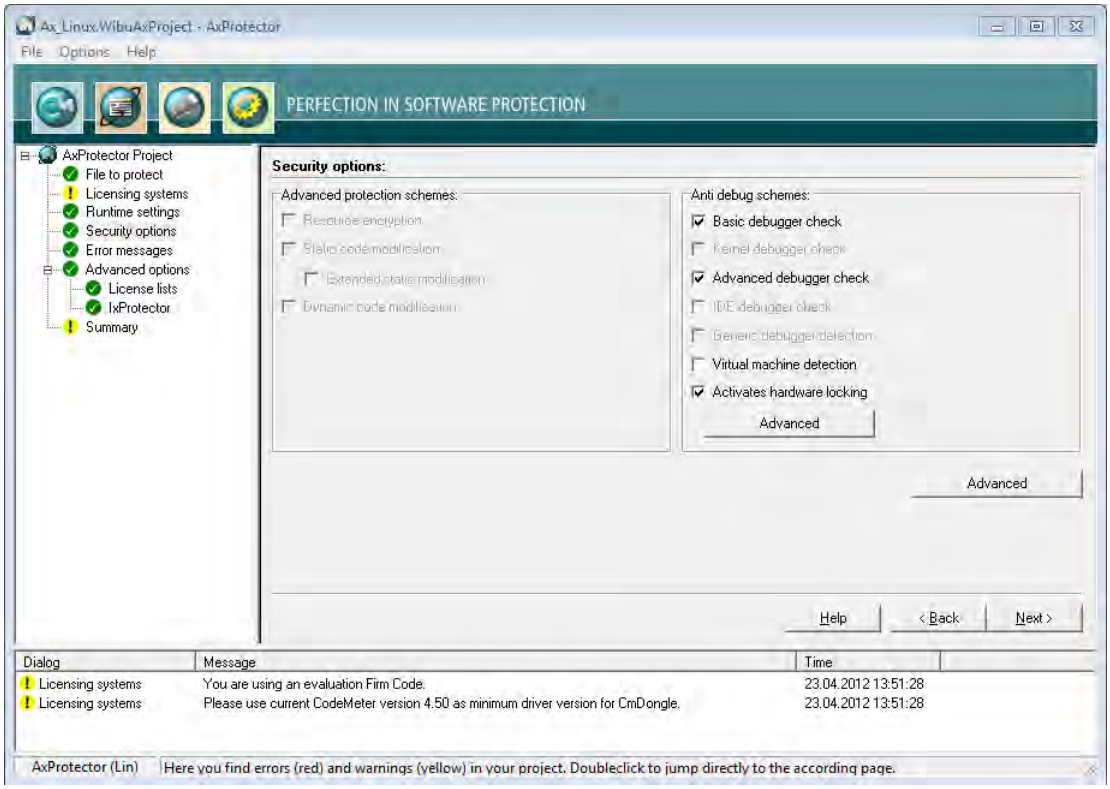


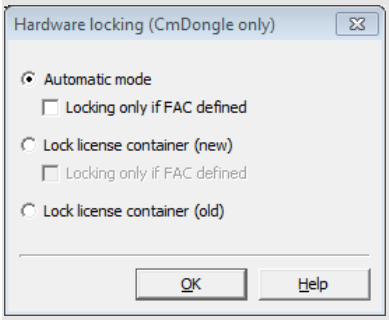
Figure 121: *AxProtector* - Linux "Security Options"

Anti-Debug Schemes

Debugger programs serve an honest role in searching for error and finding bugs. But they may also be used by hackers to analyze software. In this group you determine how to react to debugger programs (commandline options see here 267).

Element	Description
Basic Debugger Check	The 'Basic Debugger Check', checks to see if a debugger is attached to your application. When a debugger is found, your application will not be started or exited.
Advanced Debugger Check	Checks in an advanced search for debugger programs which may run parallel to your application, also cracker tools, such as, ImpREC, are detected. In the case a debugger is found, your application will not be started.
Virtual Machine Detection	Detects if the application is to be started on a virtual machine and prevents this.
Activate license access lock	This option locks the license access to the used Firm Item in a *CmContainer* as soon as a debugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the **"Configuration"** button. This button is activated only for *CodeMeter.*
Configuration	If the option **"Activate license access lock"** is activated, you are able to define further settings in the dialog which opens by clicking the **"Configuration"** button:

Element	Description
	Depending on the Firmware used this dialog allows to define separate locking scenarios (for more detailed information see separate CodeMeter Developer Guide, section "Advanced CodeMeter Features | Locking a CmContainer").

Locking Scenario	Description
immediate locking	is performed starting with Firmware Version 1.14 as soon as a debugger is detected.
prepared locking	is performed by checking the *Firm Access Counter* (FAC). The *Firm Access Counter* locates 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`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked. The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.  Figure 122: *AxProtector* -"Security Options - Hardware Locking" The following settings are available:

Option	Description
"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher, the *Firm Item* is immediately locked. For compatibility reasons this represents the default setting.
"Automatic Mode" activated and "Locking only if FAC defined" activated	If the Firmware version is smaller than 1.14 and a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. If the Firmware version is 1.14 or higher and a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (new)" activated and "Locking only if FAC defined" not activated	This option requires a Firmware version 1.14 or higher. The *Firm Item* is immediately locked. Seen from a security point of view this is the recommended option. This, however, requires that all *CmContainer* in the field must have a Firmware Version 1.14 an higher.
"Lock License Container (new)" and "Locking only if FAC defined" activated	This option requires a Firmware version 1.14 or higher. If a *Firm Access Counter* unequal `65535` has been programmed, the *Firm Item* is immediately locked.
"Lock License Container (old)" activated	Independent from the Firmware version, if a *Firm Access Counter* unequal `65535` has been programmed, the counter will be decremented by a value of `1`. This holds for all Firmware versions. If 'prepared locking' is programmed, the *Firm Access Counter* is decremented by a value of `1`.

7.4.6.4.1 Advanced Security Options

This input window lets you define further settings.

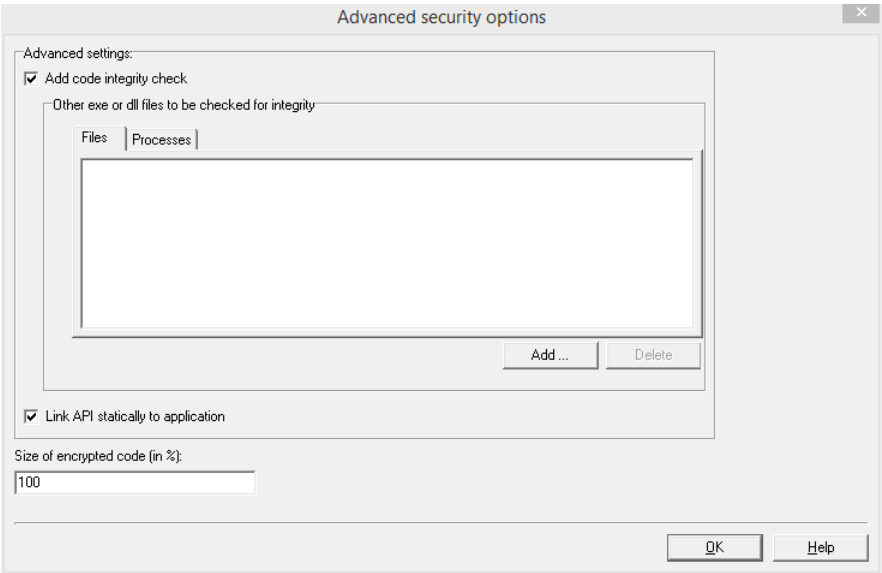
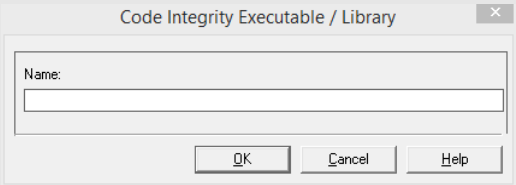
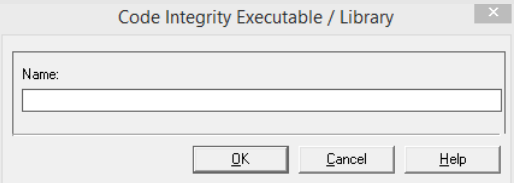


Figure 123: *AxProtector* - Linux "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

Element	Description
Add code integrity check	The protected application is checked for code integrity using asymmetric authentication 47 mechanisms, if you check this box (commandline options see here 270). On code integrity check first a check sum (hash value) of the application is created and signed with the private key of the Individual Software Vendor (ISV). The hash value and the signature are added to the application. The recalculation and the integrity check of the hash value and thus of the application is performed at runtime check using the public key located in the software (AxEngine). Alternatively to the default private key you can also apply the commandline option -sig 266 to use an entry of a *Hidden* or *Secret Data* field to define another private key. Moreover, the code integrity check may also cover several executable files / libraries. Then each file is able to check all other files for integrity. Each file then requires the public key of the ISV: The hash value of the files to be checked then is recalculated and compared to the hash value signed with the private key. To add other files for performing an integrity check, please proceed as follows. **1.** Set focus to tab "**Files**". **2.** Click the "**Add**" button. The dialog for adding displays.  **2.** Add a single or several executable files / libraries by completing the "**Name**" field. The sequence of the specified files does not matter. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button. Moreover, on encrypting a DLL also a list of applications can be transferred allowed to load these libraries. On loading the DLL then it is checked whether the process name includes one of the names specified in tab "**Files**". If not, an error message displays and subsequently the application closes. To add processes please proceed as follows: **1.** Set focus to tab "**Processes**". **2.** Click the "**Add**" button. The dialog for adding displays.

Element	Description
	 **3.** Add one or more processes which include one or more application names listed in tab "Files" by completing the field "**Name**". The sequence of the specified files does not matter. If the same application names are also specified in the list of tab "Files" also their code integrity is checked. Specifying the file extensions is optional. If using `*.wbc` files across several platforms, omitting the file extensions is recommended. **4.** Confirm each specification using the "**OK**" button.
Link API statically to Application	The *CodeMeter Core API* is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here 263).
Size of encrypted Code (in %)	Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here 270).

7.4.6.5 Error Messages

This input window lets you define the messages displayed if errors occur.

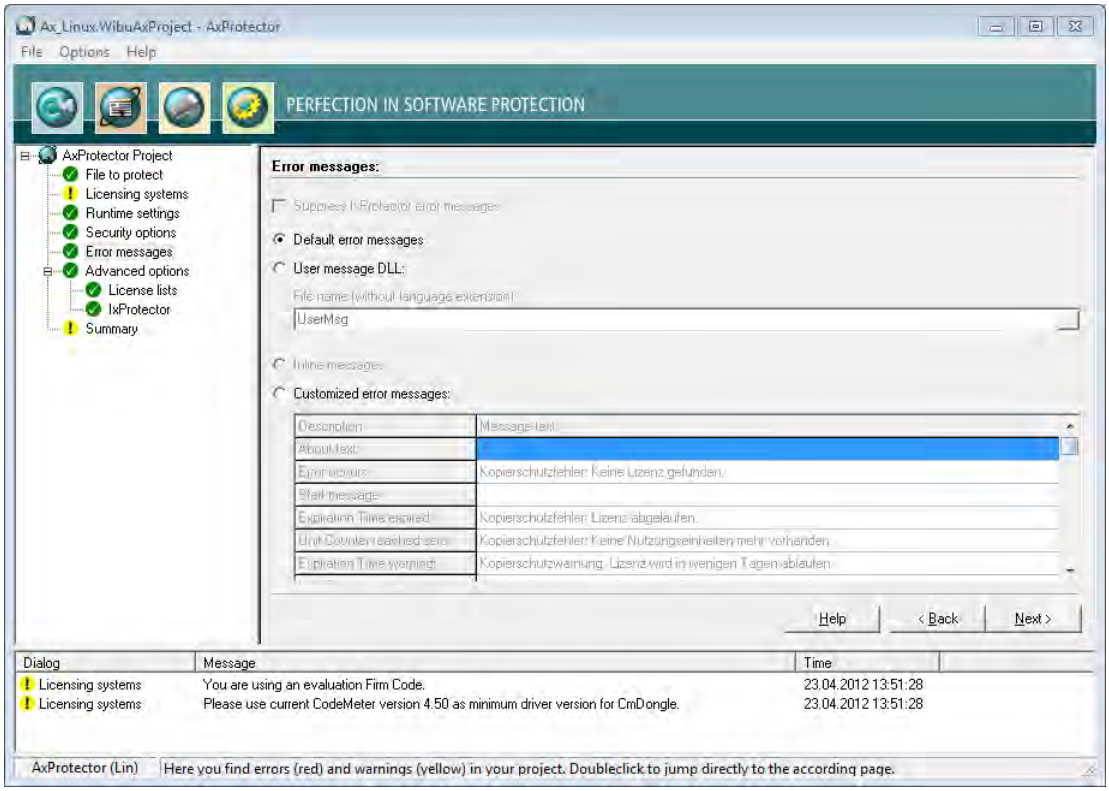


Figure 124: *AxProtector* - Linux "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.4.6.6 Advanced Options

This input window lets you set further encryption options.

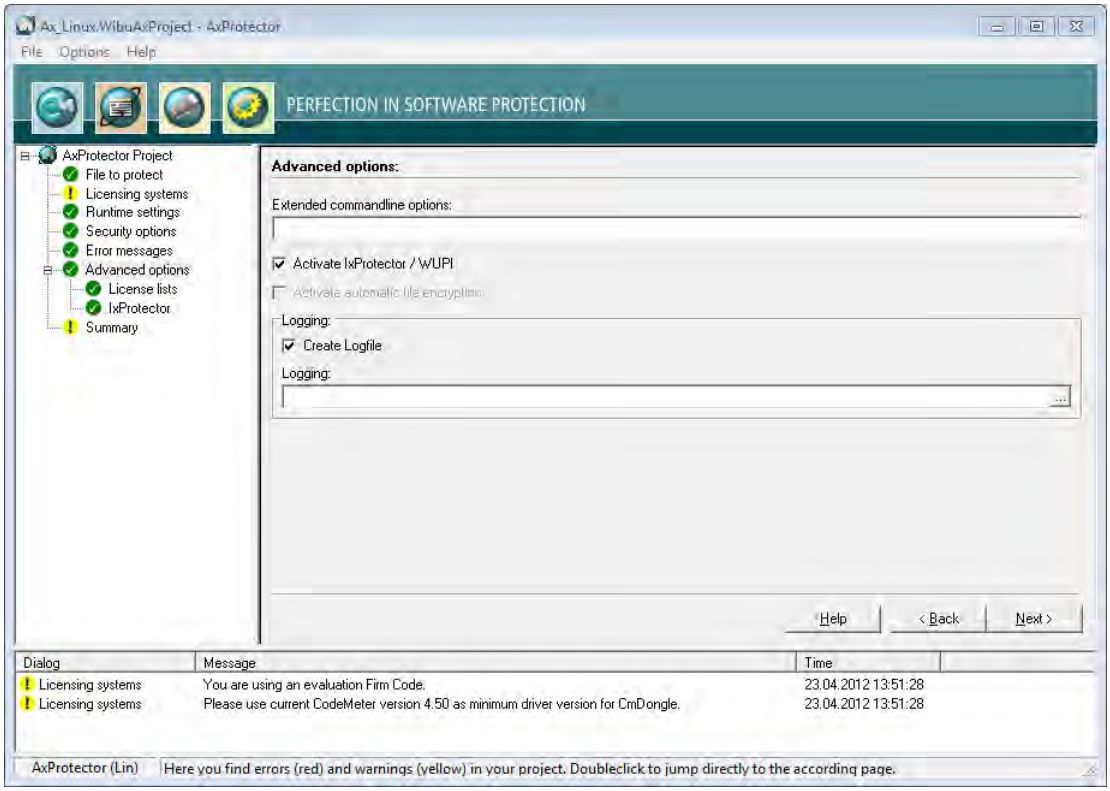


Figure 125: *AxProtector* - Linux "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector / WUPI	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (commandline option see here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.4.6.6.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289. License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290.

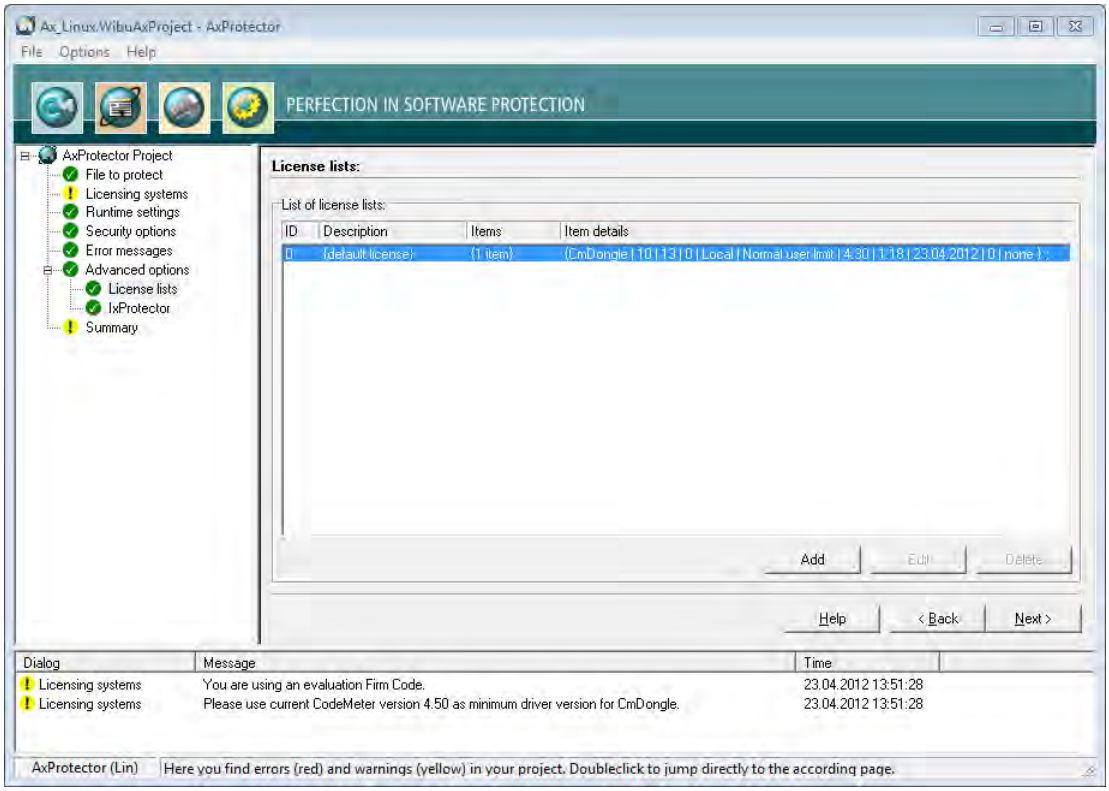
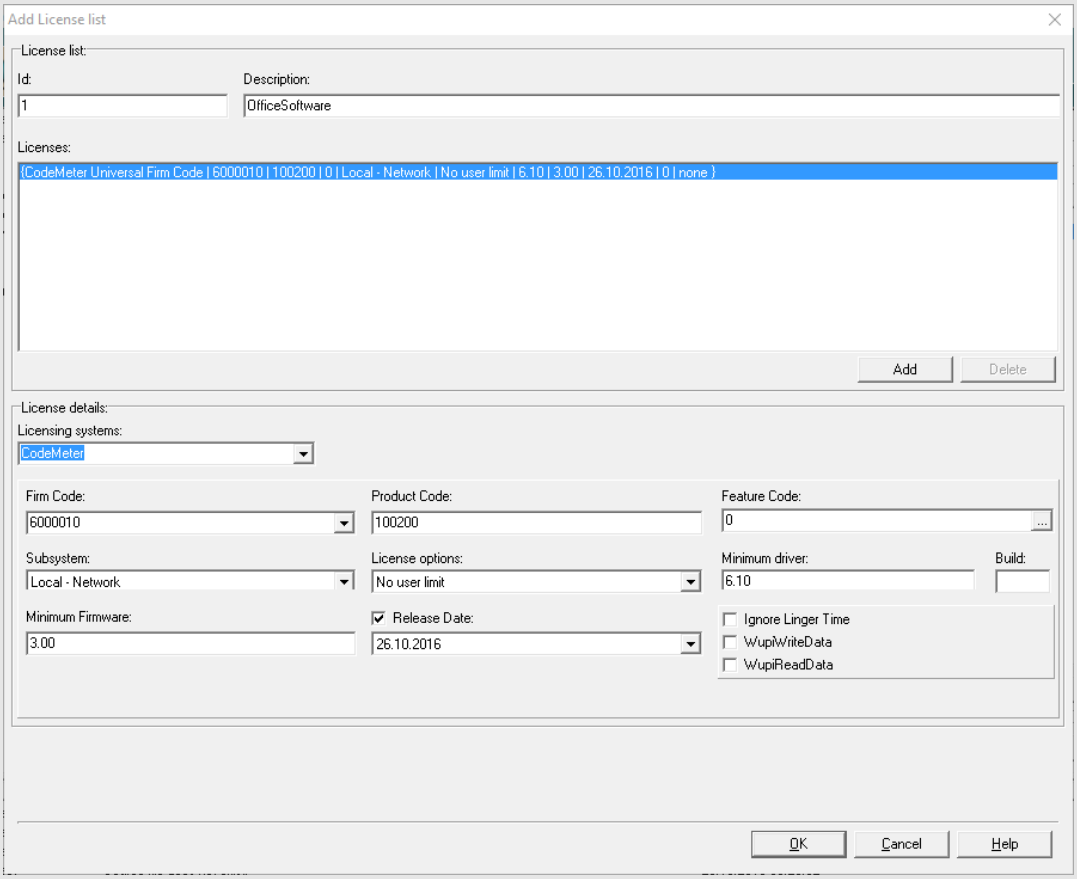


Figure 126: *AxProtector* Linux - "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **ID**s starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 127: *AxProtector* - Linux - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code** *CodeMeter* Software Development Kit (SDK) / **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle `5010` *CmActLicense* Evaluation *Firm Code* — CmActLicense Commandline option see here 264 .

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description** Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description** Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. • You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: ***Firm Codes*** **(licensing system)** / **Version** `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. `5010, 5.000.000-5.999.999` (*CmActLicense*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264 .
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist:

Element	Description	
	Firm Codes (licensing system)	Version
	`6000010; >= 6.000.000` (*Universal Firm Code*)	3.00 This supports the License Transfer feature.
	`10, 100.000-4.999.999` (*CmDongle*)	1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	`5010, 5.000.000-5.999.999` (*CmActLicense*)	1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.
	Commandline option see here 265.	
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55".	
	Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.	
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.	
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.	

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

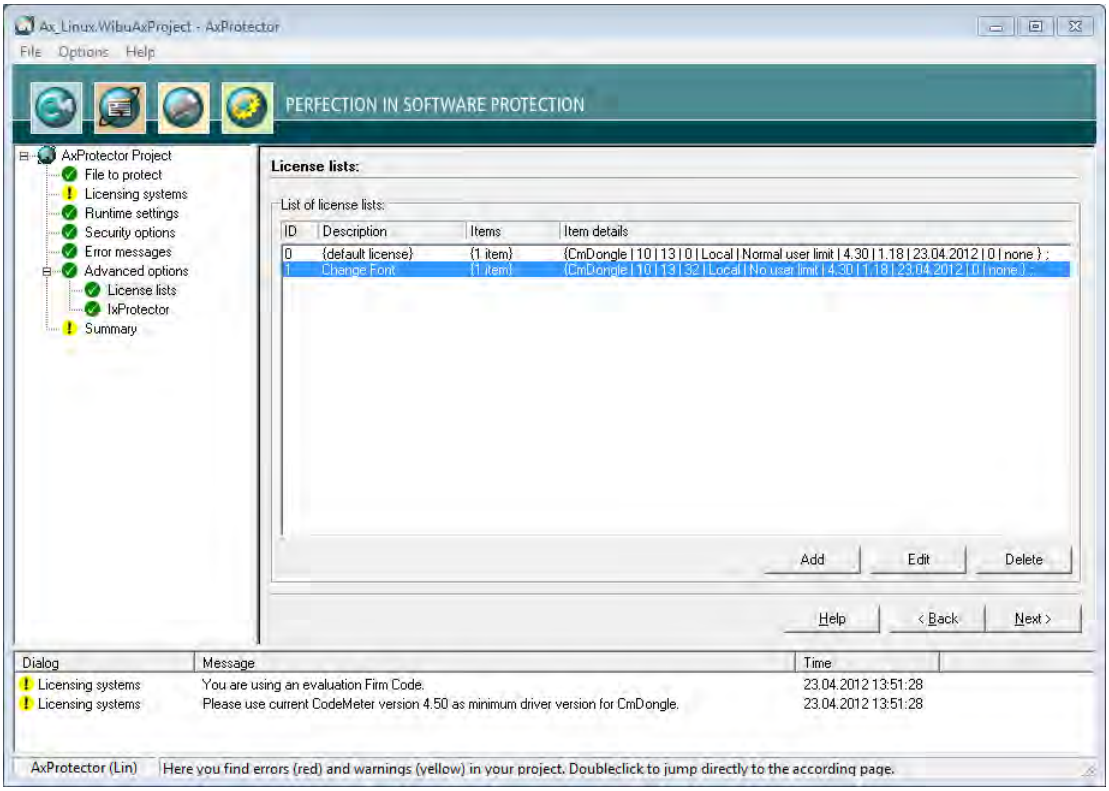


Figure 128: *AxProtector* - Linux - "Completed License Lists"

7.4.6.6.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

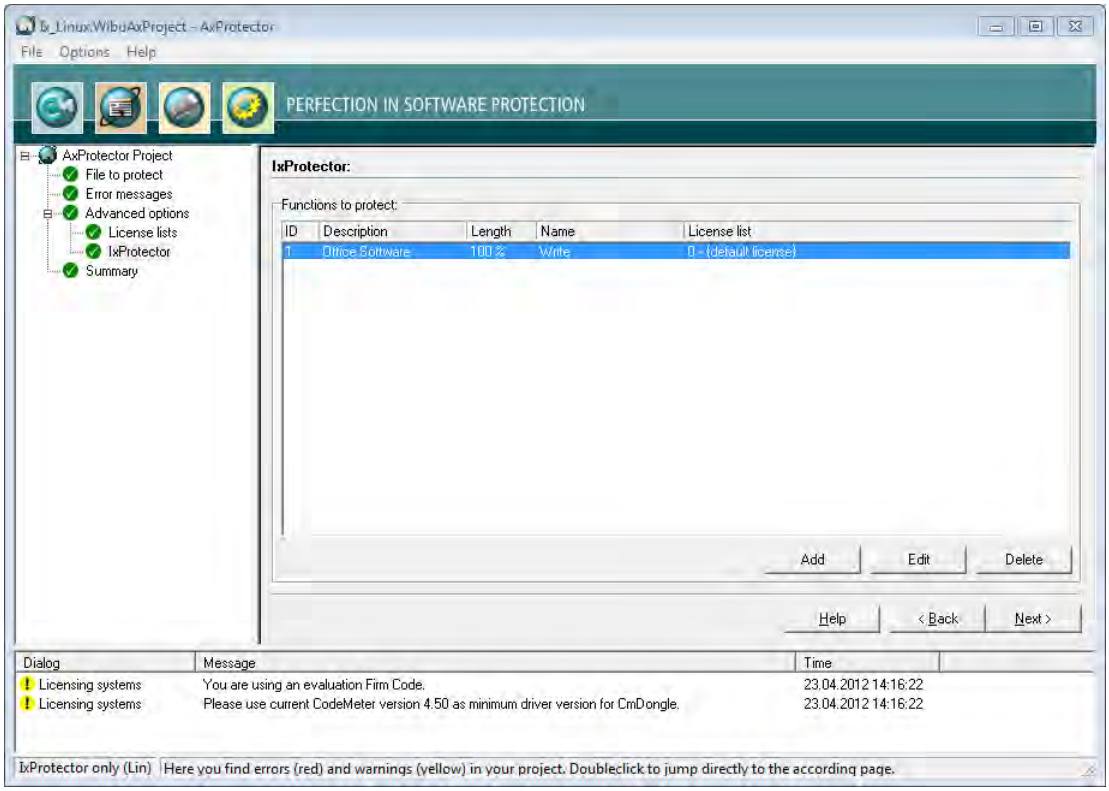
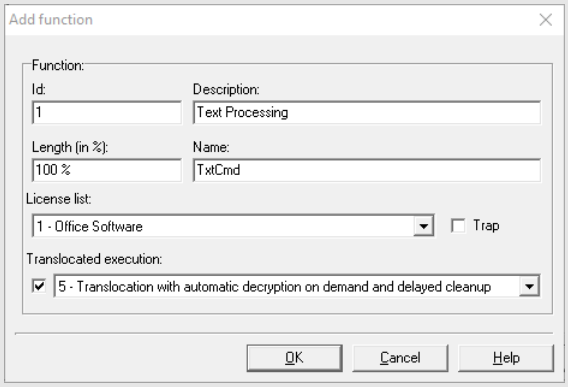


Figure 129: *AxProtector* - Linux - "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: 1. Click the **"Add"** button in the group "IxProtector Options". 2. Define the function by completing the fields in the "Function" group.  Figure 130: *AxProtector* - Linux - "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291.
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td>License List</td><td>Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.</td></tr><tr><td>Trap</td><td>Activates the trap function for the function.</td></tr><tr><td>Translocated execution</td><td>Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.<table><tr><th>Option</th><th>Description</th></tr><tr><td>1</td><td>Translocation with automatic decrpytion on demand and cleanup.</td></tr><tr><td>2</td><td>Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).</td></tr><tr><td>5</td><td>Translocation with automatic decrpytion on demand and delayed cleanup. (Default)</td></tr><tr><td colspan="2">Command line option see here 286 .</td></tr></table></td></tr></table>
	3. Click the **"OK"** button. The new functions are added to the function list.

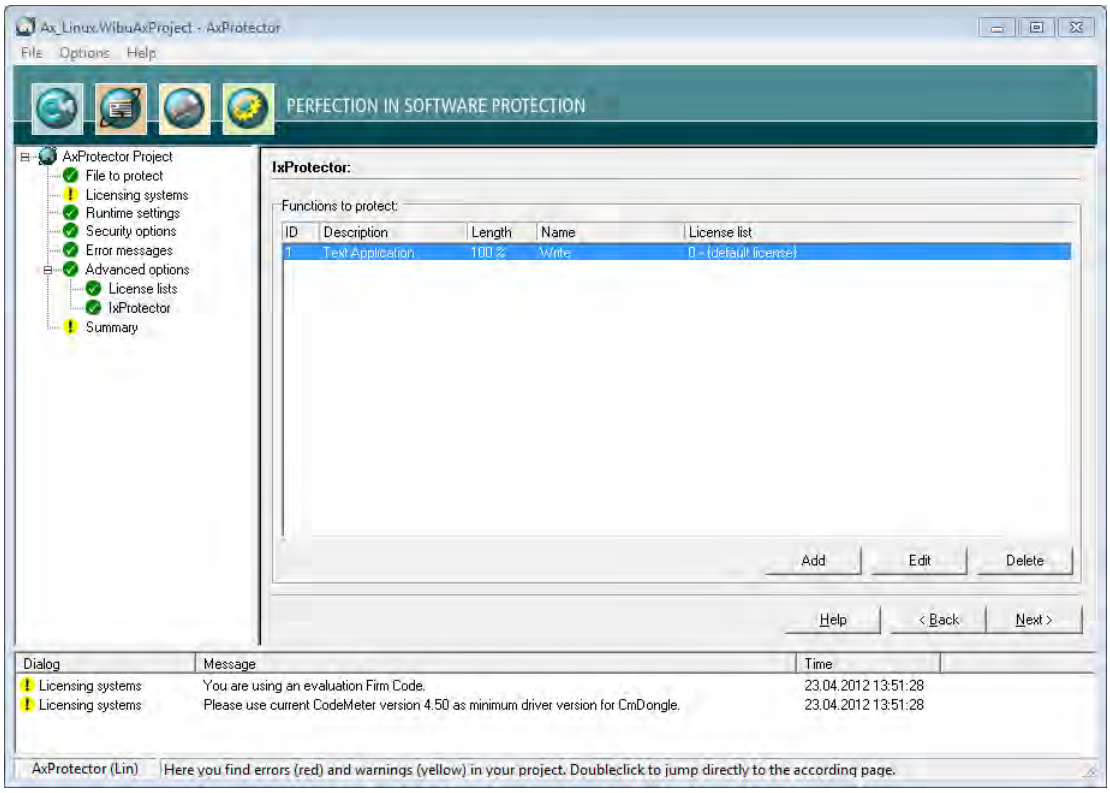


Figure 131: *AxProtector* - Linux - "Completed Function List"

7.4.6.7 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

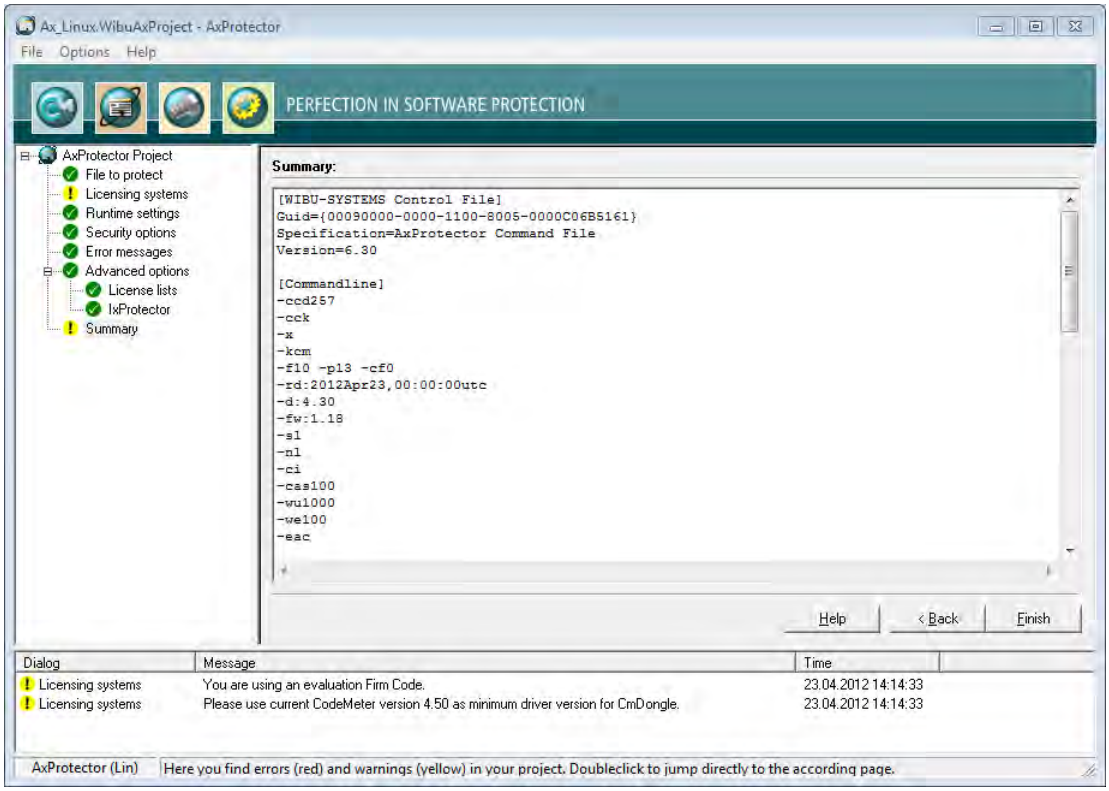


Figure 132: *AxProtector* - Linux "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

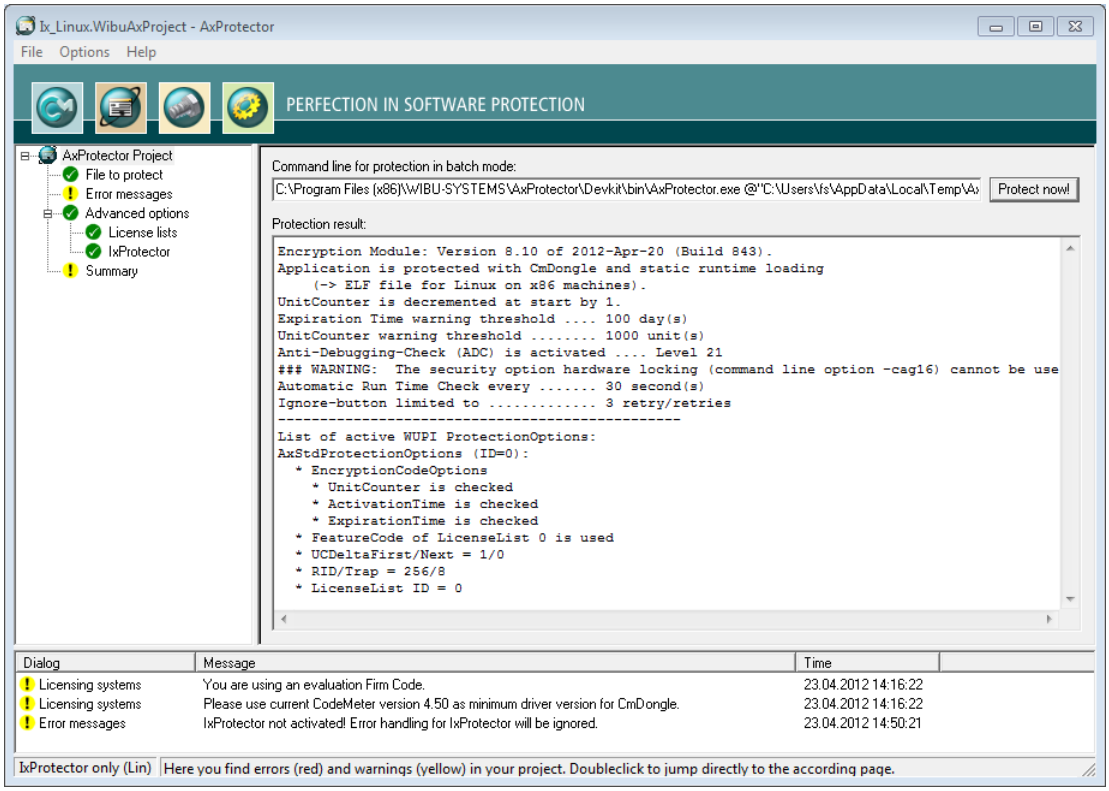


Figure 133: *AxProtector* - Linux "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode.

Element	Description
	You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.5 IxProtector Tab

This tab offers you the selection of the following project types:

	Windows Application or DLL
	.NET Assembly
	.NET Standard 2.0 Assembly
	Linux Application or Shared Object
	macOS Application or Dylib

7.5.1 Windows Application or DLL

When you want to encrypt specified functions of an application using an index-based list, you select this project type. However, then the complete application is not additionally protected with *AxProtector*.

Wibu-Systems recommends to use *IxProtector* within *AxProtector* if no other special requirements exist.

Then *IxProtector* finds the respective code areas and encrypts them. But even when you choose the project type increased security is fact, since *IxProtector* uses static code to be integrated later when the protected application is executed.

The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
Windows Application or DLL	IxProtector Windows	√	Windows commandline

The following menu items are available in the navigation windows:

- File to protect 197
- Error Messages 198
- Advanced Options 199
  - License Lists 199
  - IxProtector 203
- Summary

7.5.1.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

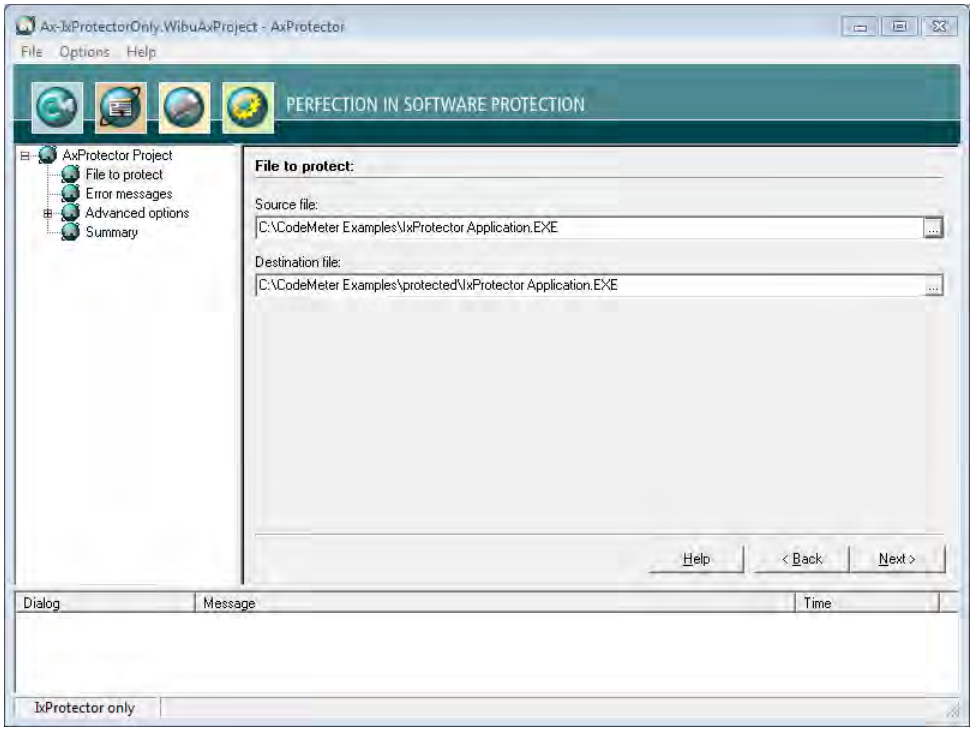


Figure 134: *AxProtector* - *IxProtector* only Windows "File to Protect"

File to protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.5.1.2 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

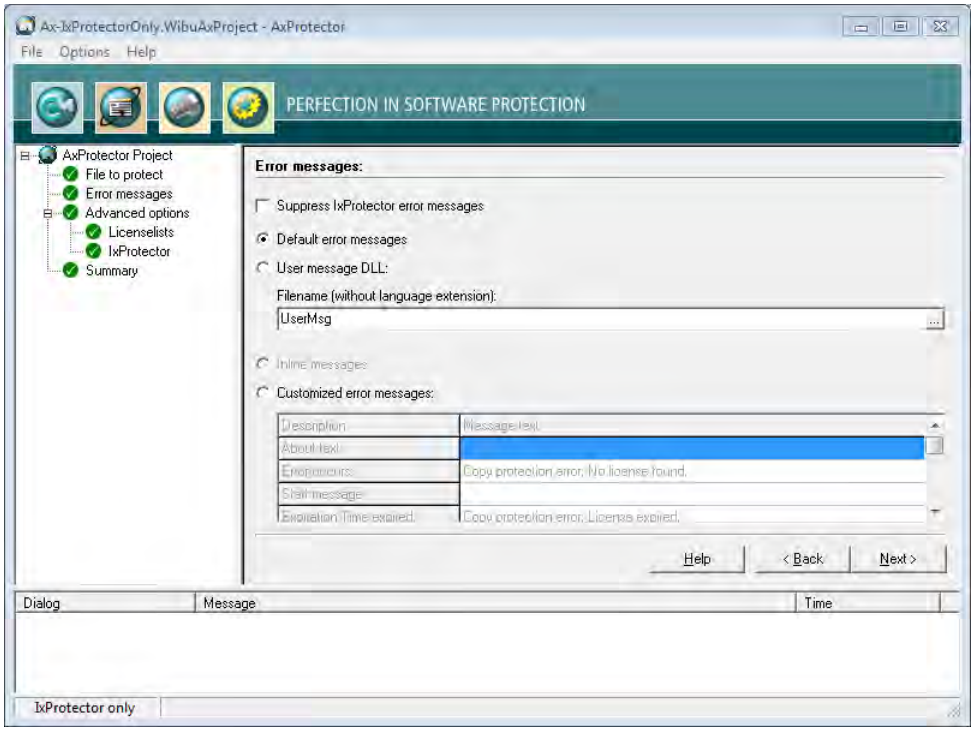
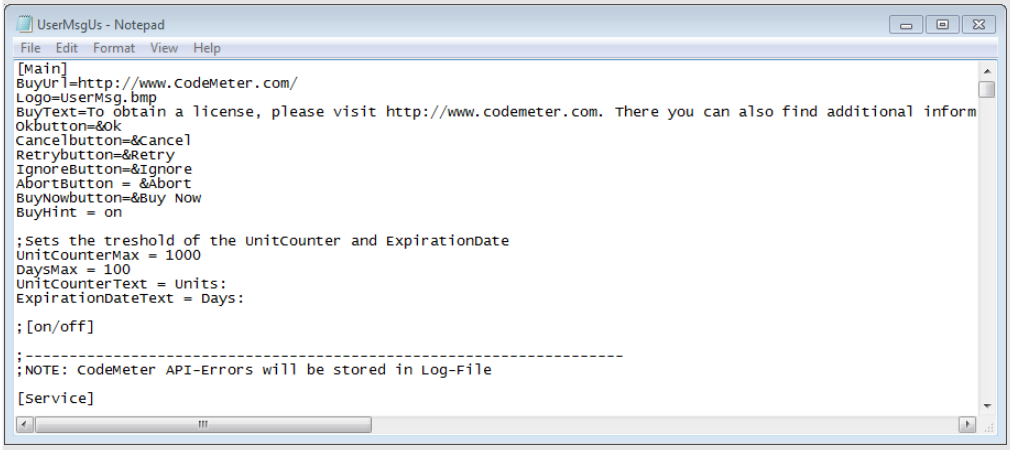


Figure 135: *AxProtector - IxProtector* only Windows "Error Messages"

Error Messages

Element	Description
Suppress IxProtector Error Messages	The output of *IxProtector* error messages is suppressed (commandline option see here 273). If you do not activate this option, when using *IxProtector* errors, additional message windows are displayed along with the messages you program in the project.
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text (commandline option see here 278). The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.  Figure 136: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension. The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.

```
[Main]
BuyUrl=http://www.CodeMeter.com/
Logo=UserMsg.bmp
BuyText=To obtain a license, please visit http://www.codemeter.com. There you can also find additional inform
Okbutton=&Ok
Cancelbutton=&Cancel
Retrybutton=&Retry
IgnoreButton=&Ignore
AbortButton = &Abort
BuyNowbutton=&Buy Now
BuyHint = on

;Sets the treshold of the UnitCounter and ExpirationDate
UnitCounterMax = 1000
DaysMax = 100
UnitCounterText = Units:
ExpirationDateText = Days:

;[on/off]

;-----------------------------------------------------------------
;NOTE: CodeMeter API-Errors will be stored in Log-File

[Service]
```

Element	Description
Inline Messages	Links for .NET projects, with an inline assembly which can also be configured by `*.ini` files. This option is available for the encryption of .NET applications only.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.5.1.3 Advanced Options

This input window lets you set further encryption options.

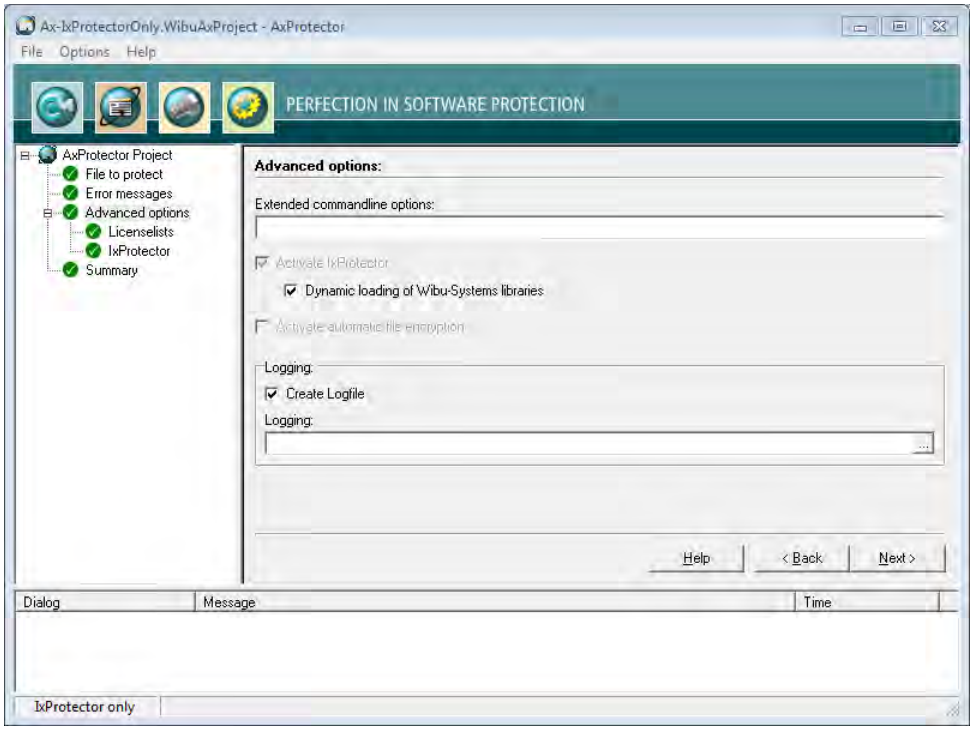


Figure 137: *AxProtector - IxProtector* only Windows "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Dynamic loading of Wibu-Systems libraries	When activated this checkbox results in a special, more time-intensive process. This when VB6 applications or dynamic loading of Wibu-Systems libraries are involved (commandline option see here [273])
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector.*
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.5.1.3.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* [289] .
License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* [290] .

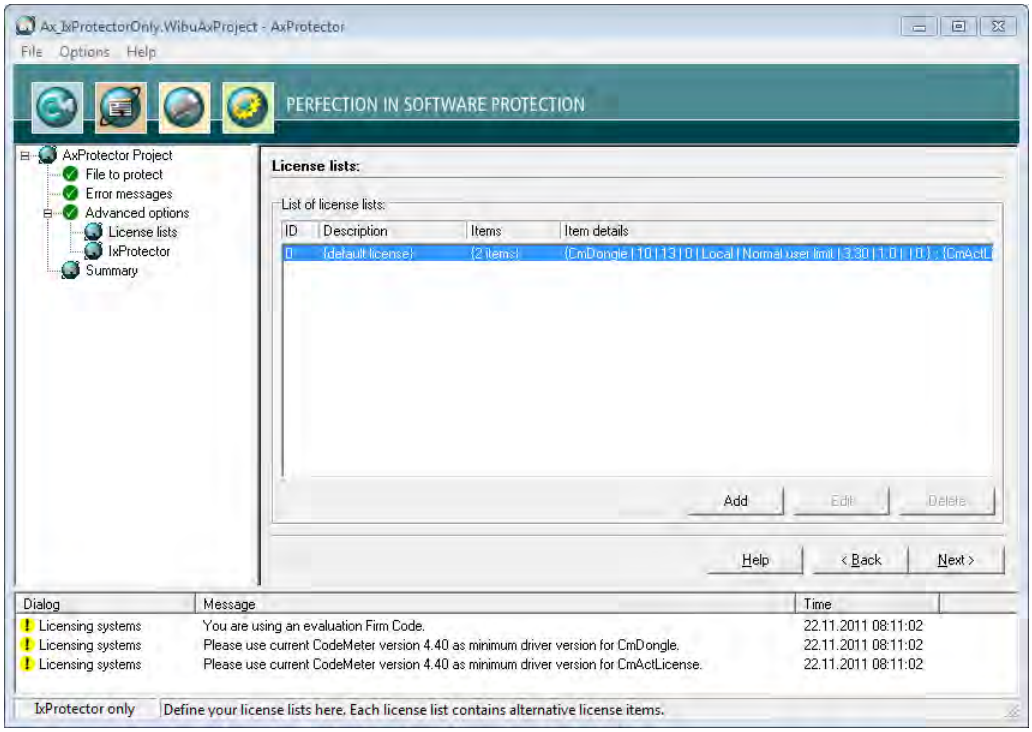
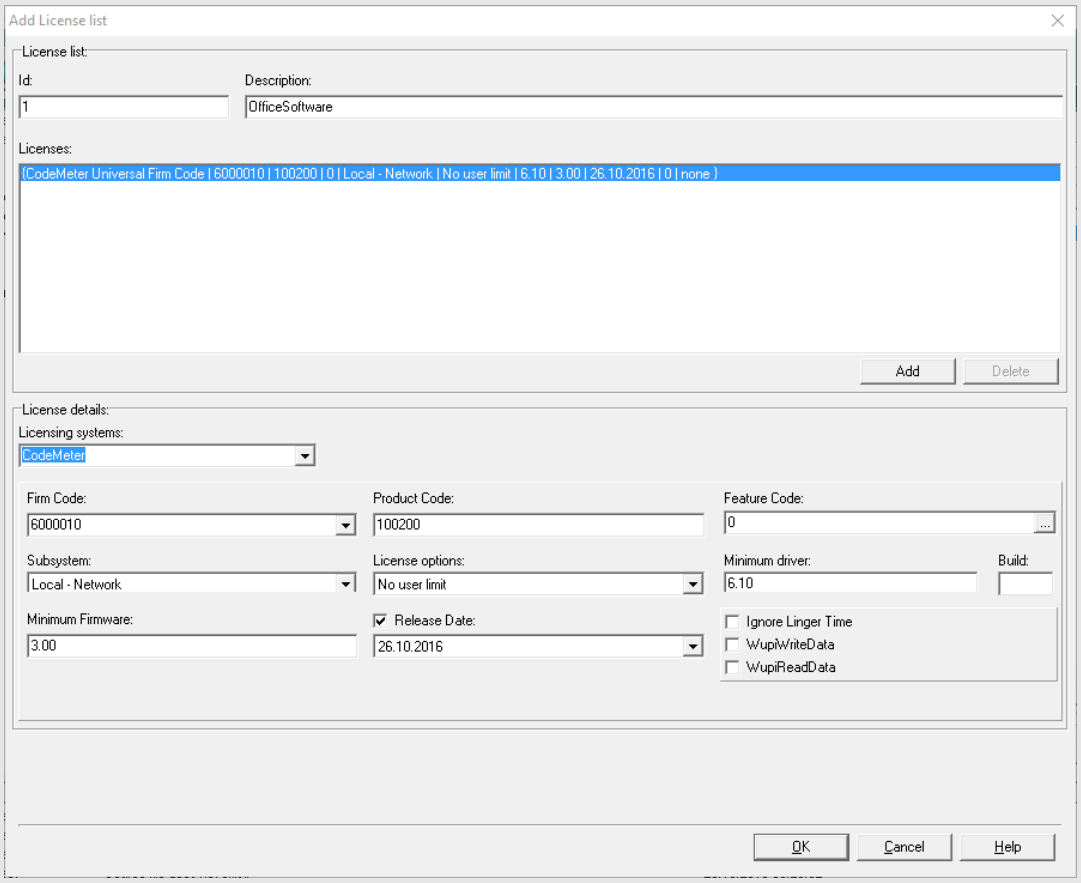


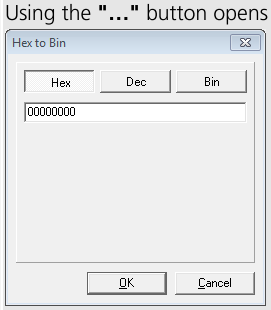
Figure 138: *AxProtector - IxProtector* only Windows "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 139: *AxProtector - IxProtector* only Windows "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Enter the Firm Code used for the protection of the license.
Product Code	Enter the Product Code used for the protection of the license.
Feature Code	Enter the Feature Code used, for example, to encrypt different versions of your application.

Element	Description
	Using the **"..."** button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. 
Subsystem	Select the subsystem in which the protected application is to search (local or network), and define the search order. **License Options** Select the options for license allocation: • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • *No User limit*
Minimum Driver Version	Specify the required minimum driver version for the protected application.
Release Date	Starting with Firmware version 1.18 *CodeMeter®* supports the *Product Item Option Maintenance Period*. In the PIO two date values are stored: a start and an end value. This allows you to implement license models which map the granting of support and services when using the software. Then the use of a license is limited to software versions, corrections, and extension 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. In the case the *Release Date* is not within the *Maintenance Period*, the use of the software is not covered by the license. To store the *Release Date*, please proceed as follows: **1.** Activate the **"Release Date"** checkbox to type in the *Release Date*. The current date is preset. **2.** Change if desired the date either directly in the field located below or use the calendar element which opens via the arrow button at the left margin of the field.
Minimum Firmware	Specify the minimum firmware version required. In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.
Ignore Linger Time	Activate this option to ignore a programmed *LingerTime*. This license option allows to define an allocation time of the license after a protected application has been released or the protected application has been closed (more Information in the *CodeMeter* Developer Guide).
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.

5. Click the **"OK"** button. The new license data is added to the license list.

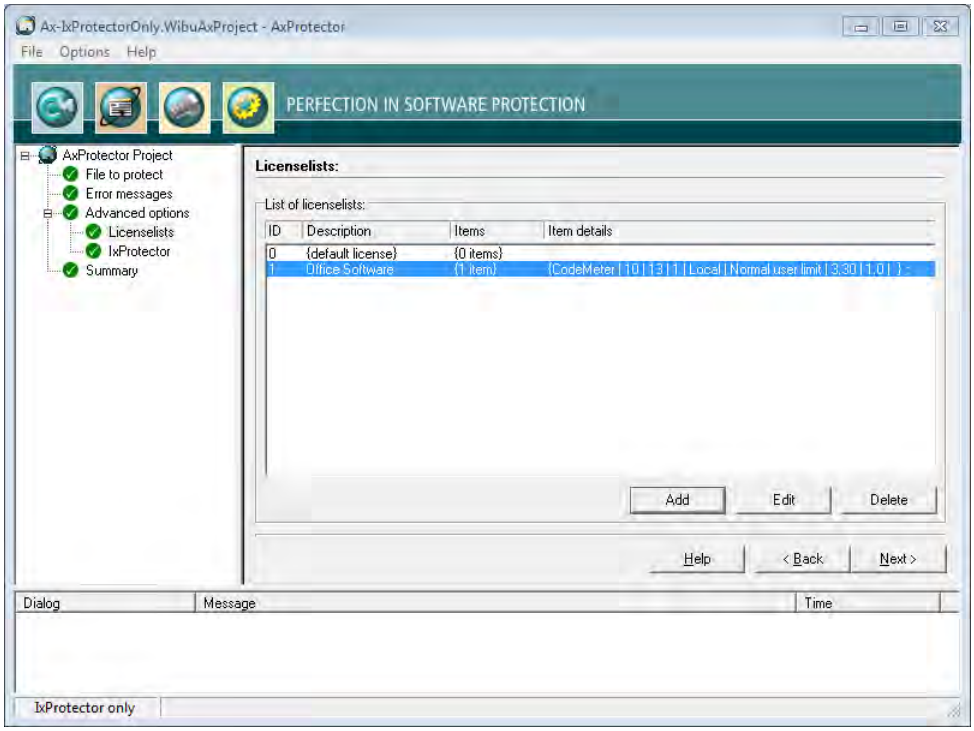


Figure 140: *AxProtector - IxProtector* only Windows "Completed License Lists"

7.5.1.3.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

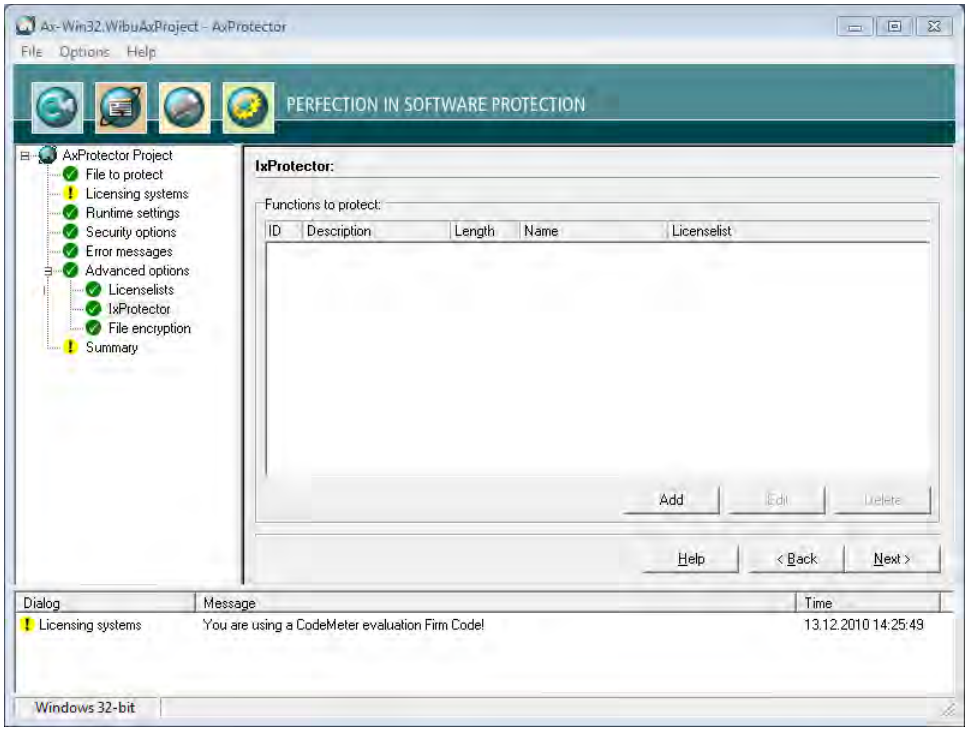


Figure 141: *AxProtector - IxProtector* only Windows "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: **1.** Click the **"Add"** button in the group "IxProtector Options".

Element	Description
	2. Define the function by completing the fields in the "Function" group.

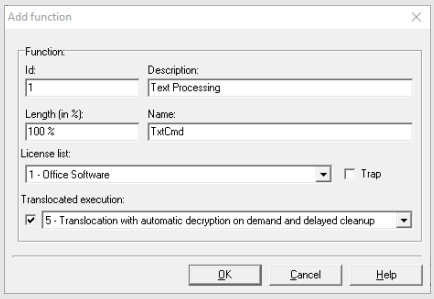


Figure 142: *AxProtector* - *IxProtector* only Windows "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291 .
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.
License List	Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.
Trap	Activates the trap function for the function.
Translocated execution	Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.

Option	Description
1	Translocation with automatic decrpytion on demand and cleanup.
2	Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).
5	Translocation with automatic decrpytion on demand and delayed cleanup. (Default)

Command line option see here 286 .

3. Click the **"OK"** button. The new functions are added to the function list.

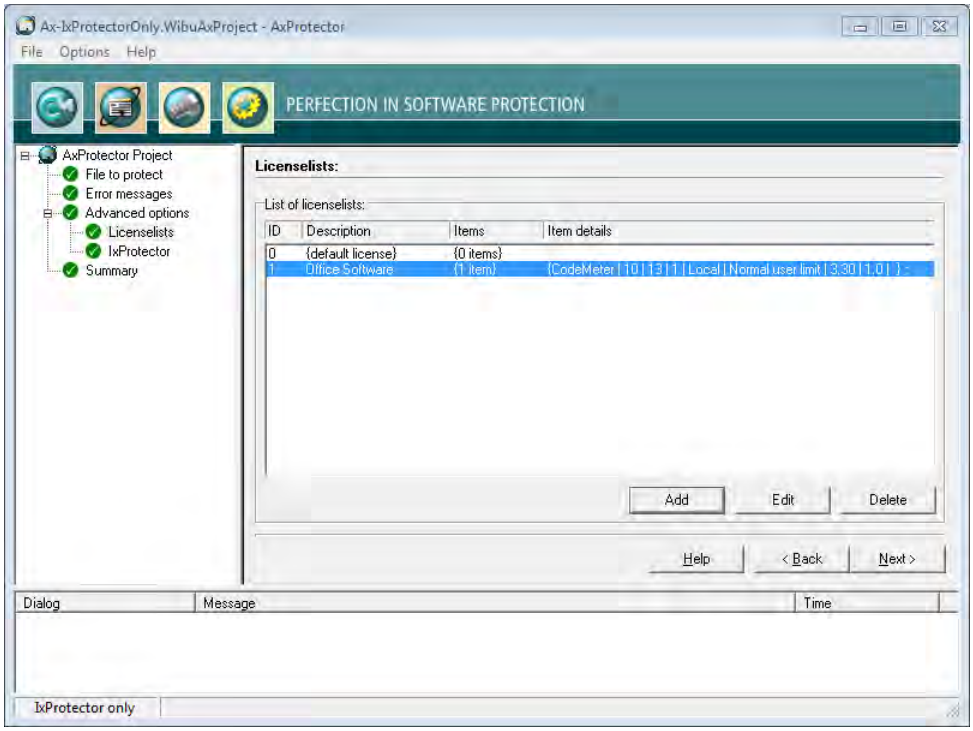


Figure 143: *AxProtector - IxProtector* only Windows "Completed Function List"

7.5.1.4 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

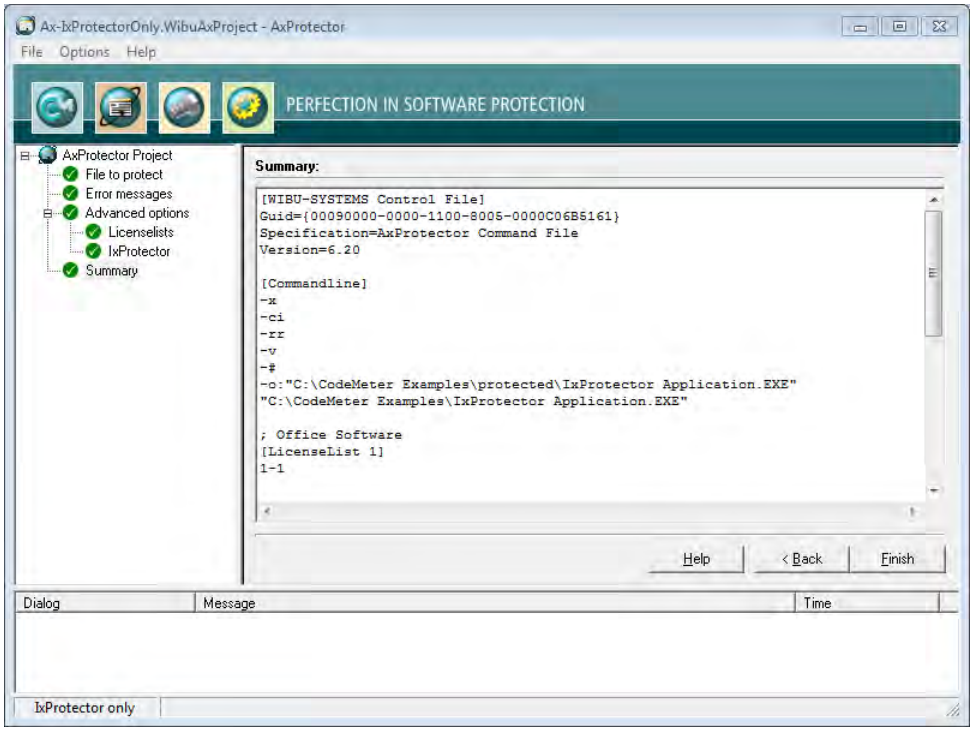


Figure 144: *AxProtector - IxProtector* only Windows "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.

Element	Description
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

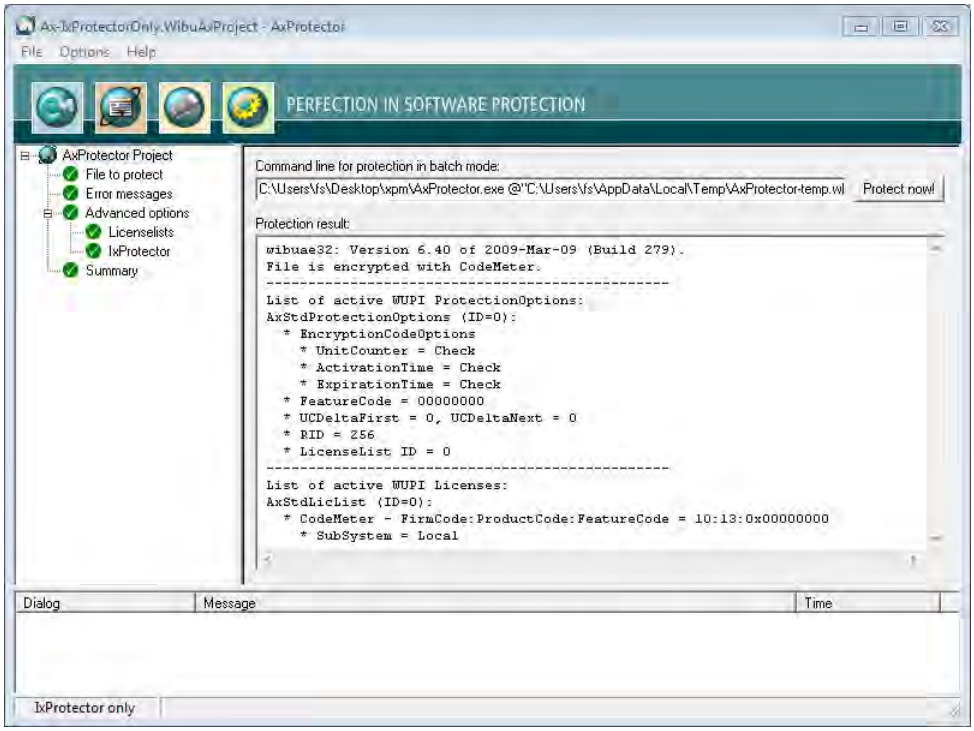


Figure 145: *AxProtector* - *IxProtector* only "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.5.2 .NET Assembly

If you want to encrypt specified functions of an application using an index-based list, you select this project type. However, then the complete application is not additionally protected with *AxProtector*.

Wibu-Systems recommends to use *IxProtector* within *AxProtector* if no other special requirements exist.

Then *IxProtector* finds the respective code areas and encrypts them. But even when you choose the project type increased security is fact, since *IxProtector* uses static code to be integrated later when the protected application is executed. The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
.NET Assembly	IxProtector.NET	√	.NET commandline

The following menu items are available in the navigation windows:

- File to protect 207
- Error Messages 208
- Advanced Options 210
  - License Lists 210
  - IxProtector 214
- Summary

7.5.2.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

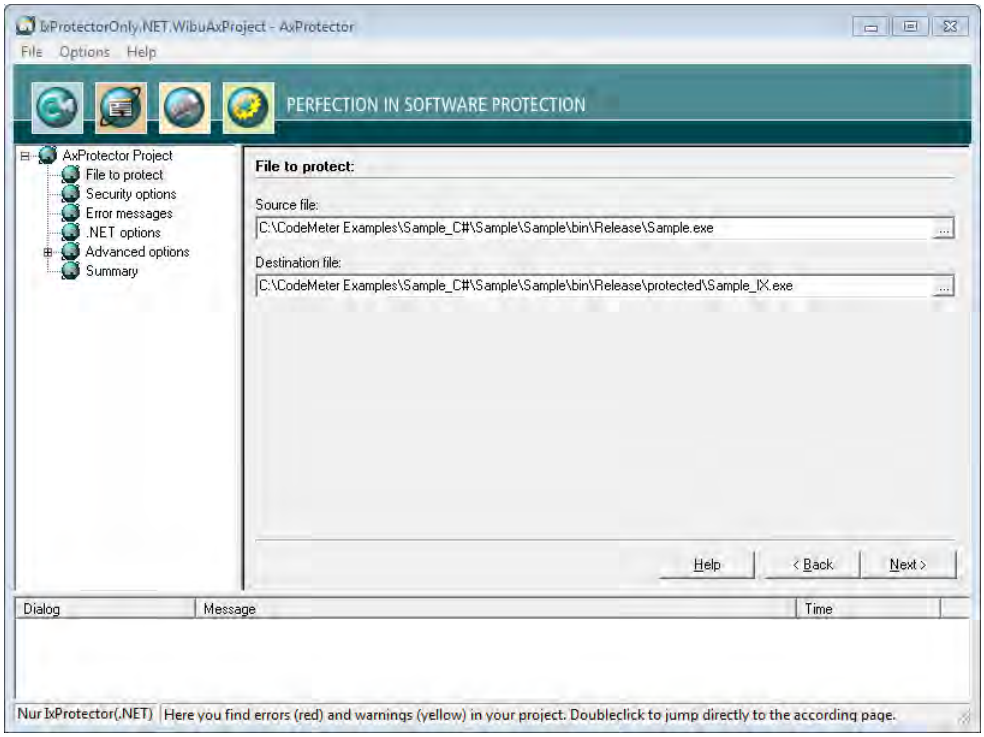


Figure 146: *AxProtector - IxProtector* only (.NET) "File to Protect"

File tom protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279.

7.5.2.2 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

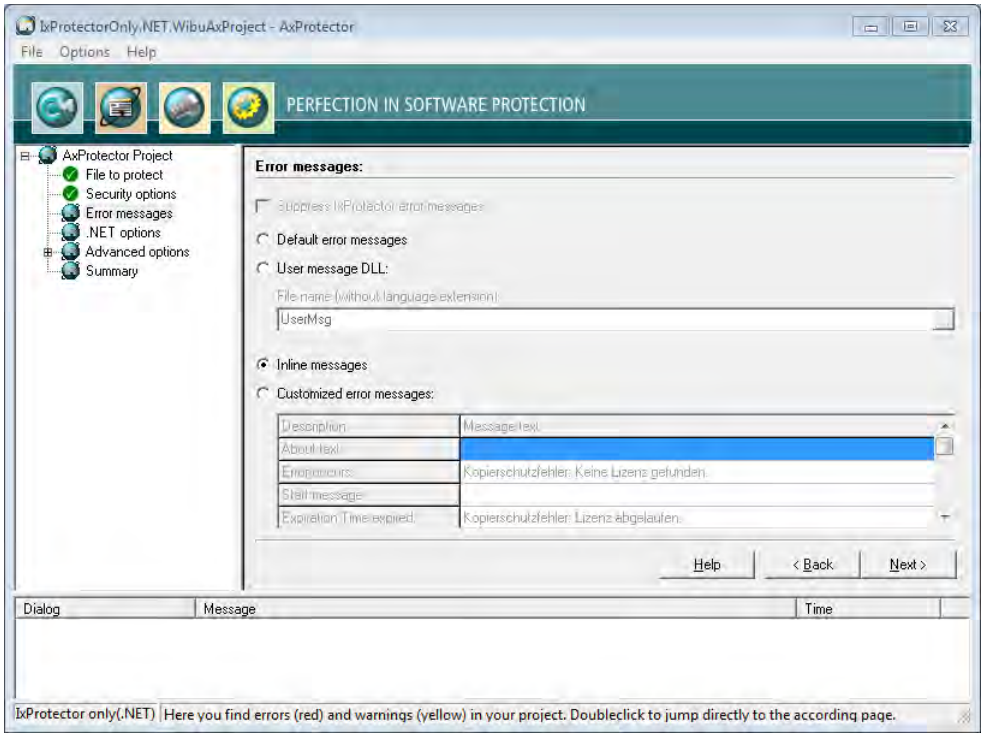
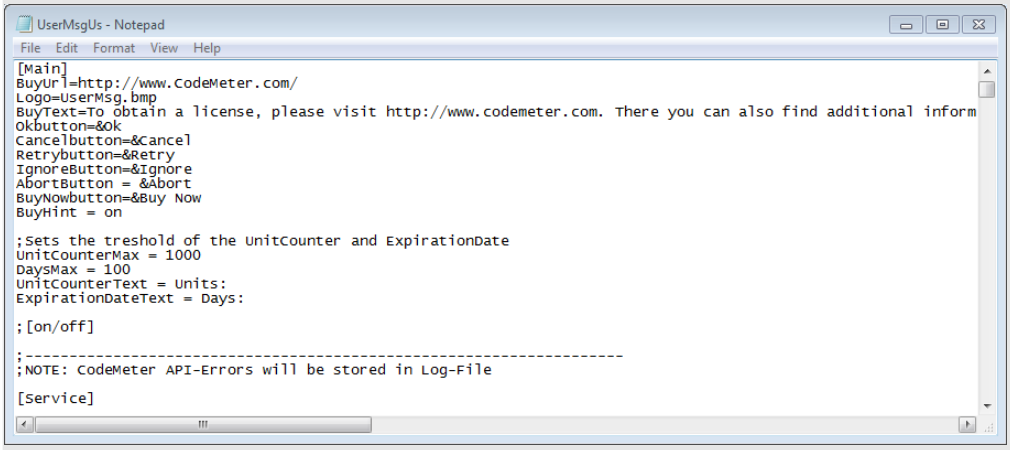


Figure 147: *AxProtector - IxProtector* only (.NET) "Error Messages"

Error messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here[277]).
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text (commandline option see here[279]). The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.  Figure 148: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension. The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.
Inline Messages	Links for .NET projects, with an inline assembly, can also be configured by `*.ini` files (commandline option see here[278]). When using Inline UserMessages the logging is saved to the directory "`%CommonApplicationData%`". When you want to specify another path specify the parameter `LogPath=<Pfad>` in the `*.ini` file.

Element	Description
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.5.2.3 .NET Options

This page allows you to specify further .NET settings.

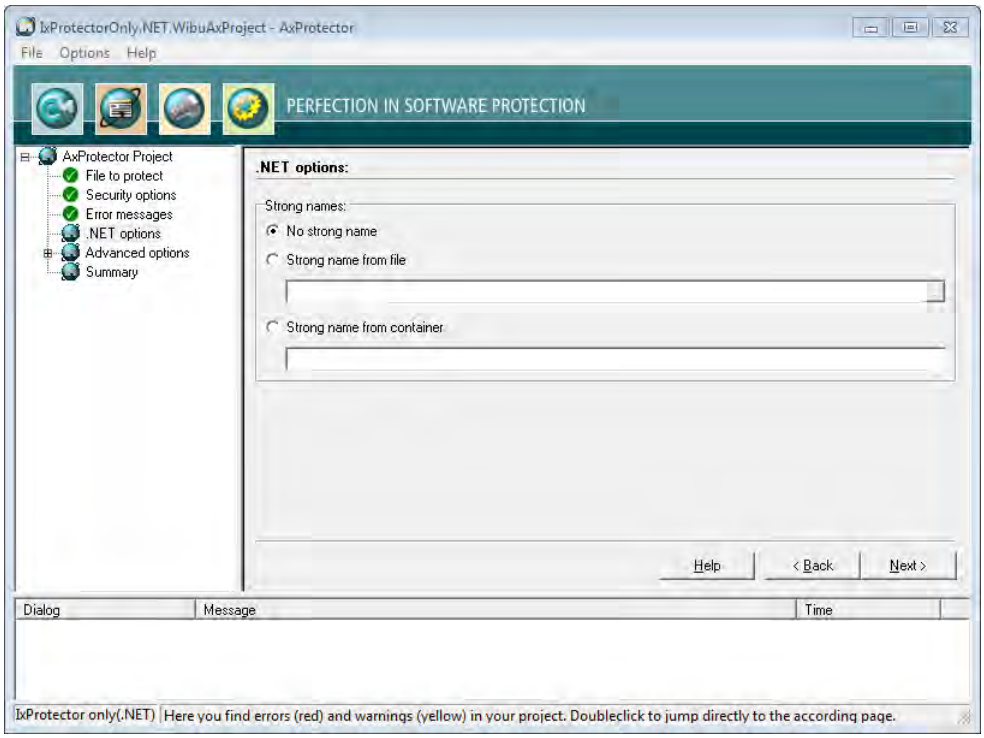


Figure 149: *AxProtector - IxProtector* only (.NET) ".NET Options"

Here you are able to specify whether your assembly is signed by *AxProtector*.

Element	Description
No Strong Name	Activate this checkbox to not sign your assembly.
Strong Name from File	Activate this checkbox to use a source file to sign the program class. Then specify a file holding the key pair to generate a strong name (commandline option see here 279).
Strong Name from Container	Activate this checkbox to use a container file to sign the program class (commandline option see here 279).

7.5.2.4 Advanced Options

This input window lets you set further encryption options.

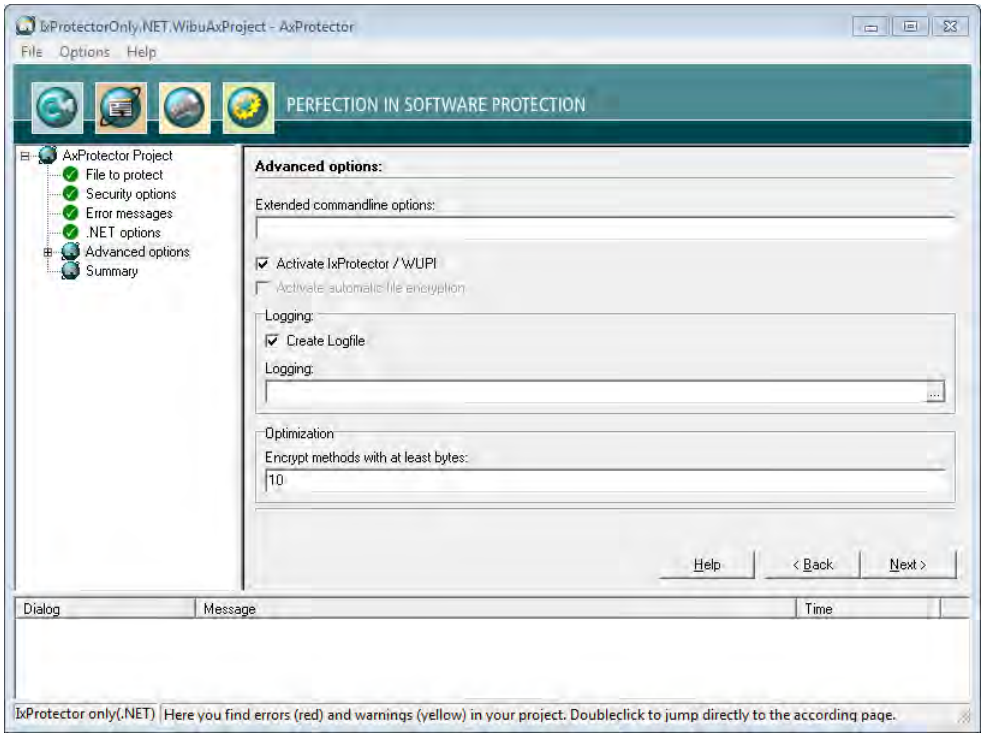


Figure 150: *AxProtector - IxProtector* only (.NET) "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (commandline option here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.
Optimizing	For an optimized performance specify here the minimum size for assemblies to be encrypted. The default setting is 10 bytes. This way you are able to exclude methods from encryption which are smaller than the number of bytes you specify here. By setting a value of 0 this feature is deactivated. Commandline option see here 275).

7.5.2.4.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

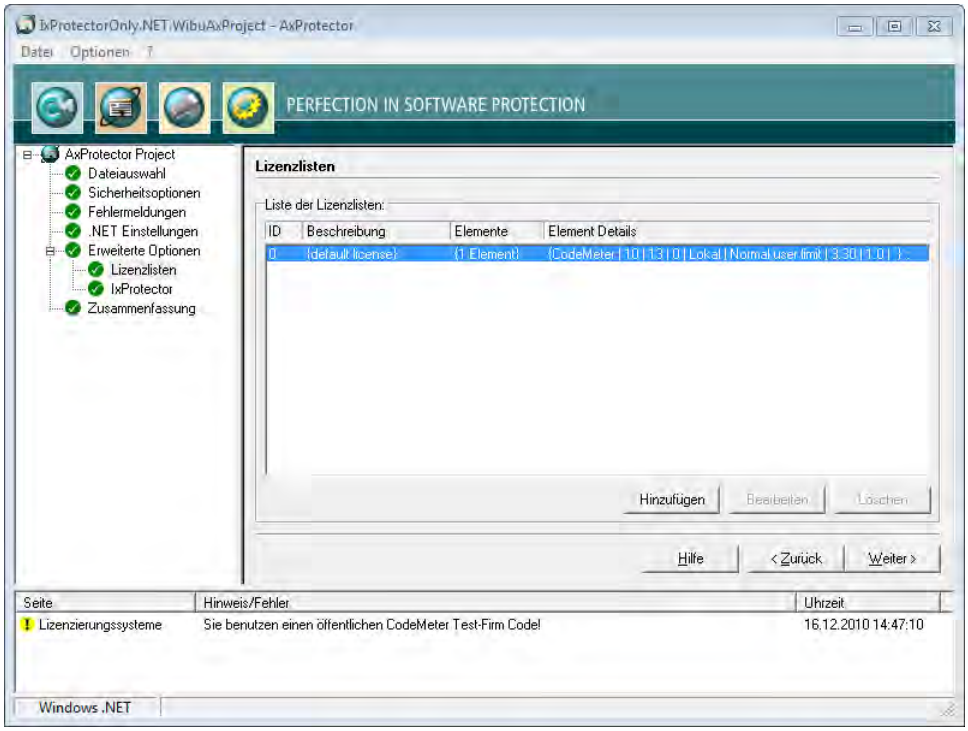
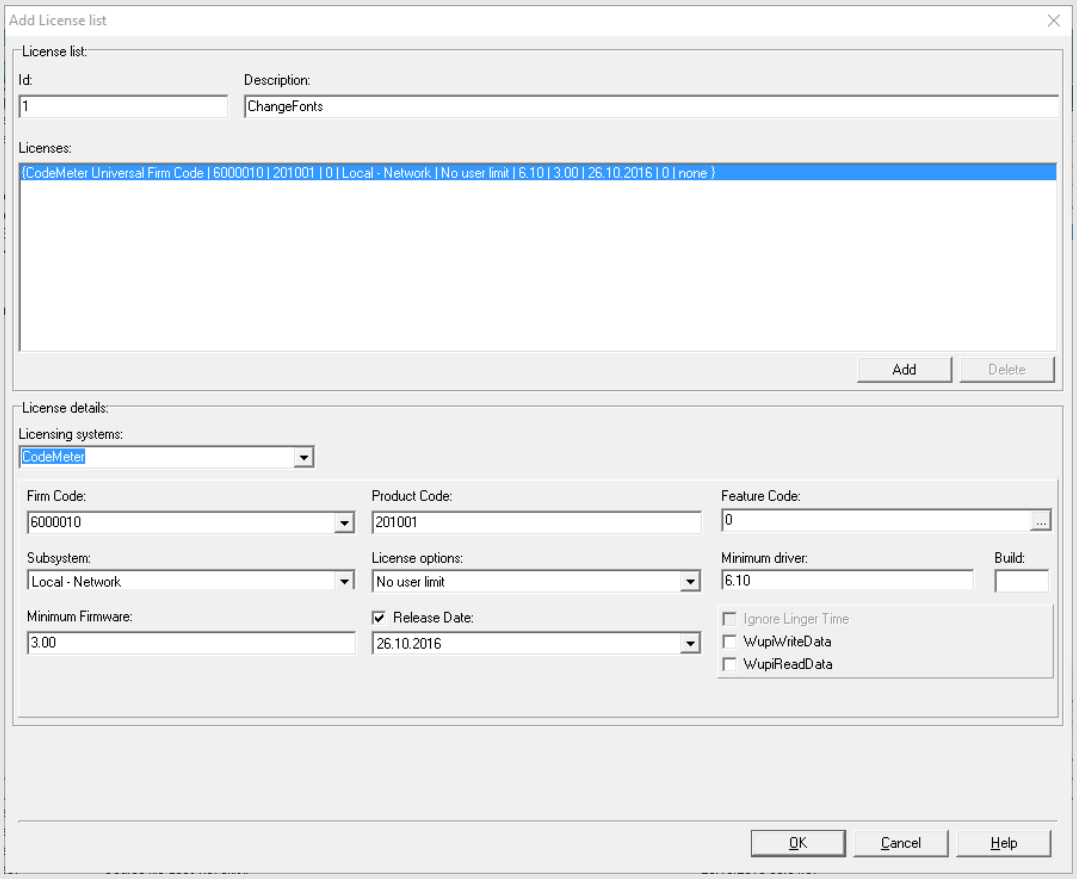


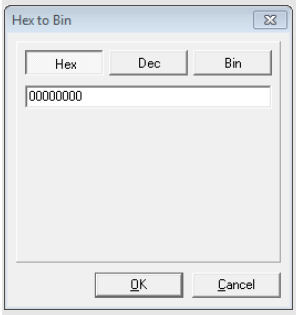
Figure 151: *AxProtector - IxProtector* only (.NET) "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 152: *AxProtector - IxProtector* Only (.NET)' - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . • Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". • If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. • In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Enter the Firm Code used for the protection of the license.
Product Code	Enter the Product Code used for the protection of the license.
Feature Code	Enter the Feature Code used, for example, to encrypt different versions of your application.

Element	Description
	Using the **"..."** button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. 
Subsystem	Select the subsystem in which the protected application is to search (local or network), and define the search order. **License Options** Select the options for license allocation: • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • *No User limit*
Minimum Driver Version	Specify the required minimum driver version for the protected application.
Release Date	Starting with Firmware version 1.18 *CodeMeter®* supports the *Product Item Option Maintenance Period*. In the PIO two date values are stored: a start and an end value. This allows you to implement license models which map the granting of support and services when using the software. Then the use of a license is limited to software versions, corrections, and extension 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. In the case the *Release Date* is not within the *Maintenance Period*, the use of the software is not covered by the license. To store the *Release Date*, please proceed as follows: **1.** Activate the **"Release Date"** checkbox to type in the *Release Date*.The current date is preset. **2.** Change if desired the date either directly in the field located below or use the calendar element which opens via the arrow button at the left margin of the field.
Minimum Firmware	Specify the minimum firmware version required. In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.
Ignore Linger Time	Activate this option to ignore a programmed *LingerTime*. This license option allows to define an allocation time of the license after a protected application has been released or the protected application has been closed (more Information in the *CodeMeter* Developer Guide).
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.

5. Click the **"OK"** button. The new license data is added to the license list.

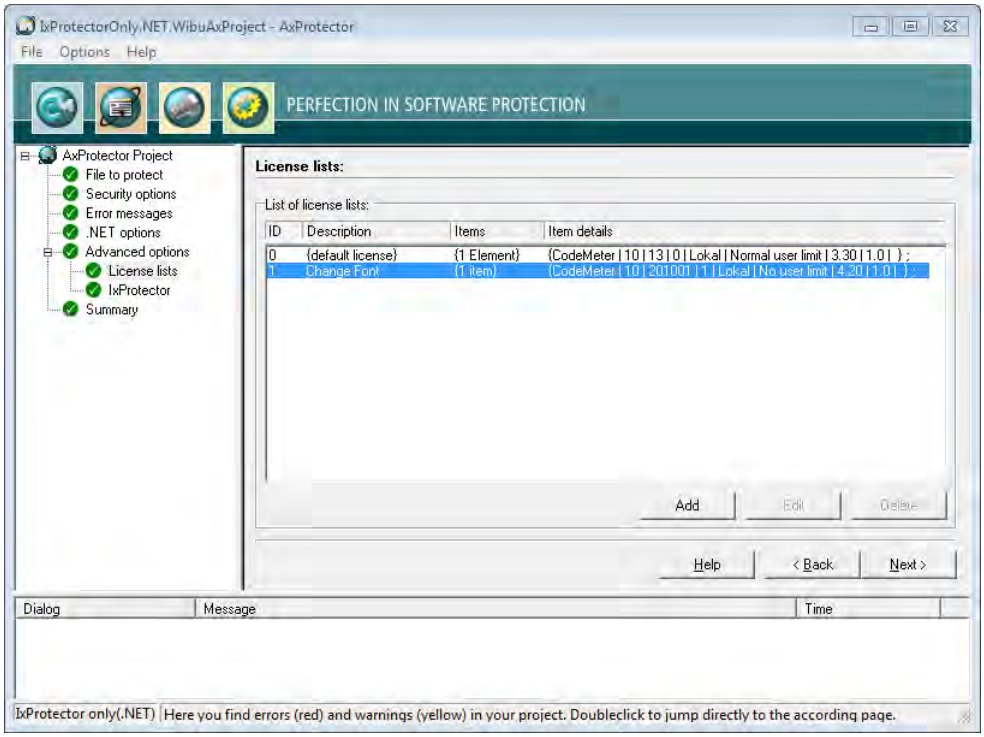


Figure 153: *AxProtector - IxProtector* only (.NET) - "Completed License List"

7.5.2.4.2 IxProtector

Using this menu item allows you to separately define single encryption types for single assembly elements.

In the case you activated the checkbox **"IxProtector"** in the menu item **"Advanced options"** the source assembly is loaded and displayed in a tree view making available all name spaces, classes, and modules.

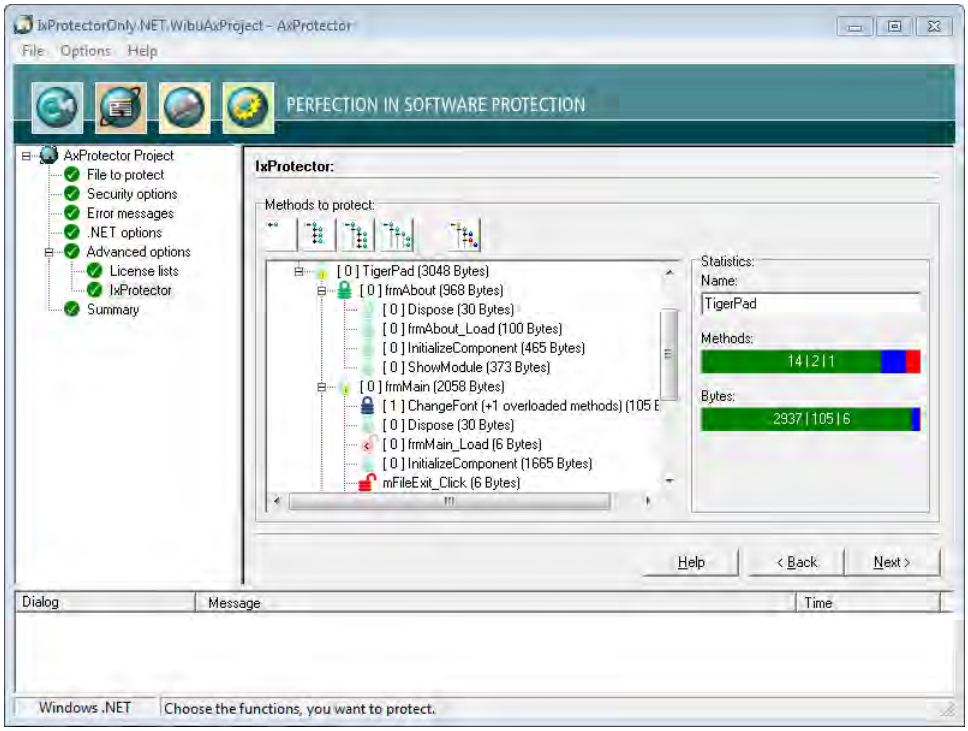


Figure 154: *AxProtector - IxProtector* only (.NET) " IxProtector"

Click the different buttons in the upper "IxProtector" area to select from different assembly views.

Views

Button	Description
	Closes all assembly levels of the tree structure.
	Expands the name space level of the assembly.

Button	Description
	Expands the class level of the assembly.
	Expands the method level of the assembly.
	Expands all parent levels of the assembly. In this view see all levels where modifications have been made.

The area "Statistics" on the right shows you more encryption details depending on the selection you have made for the tree view.

Element	Description
Name	This field refers to the name of the element you have marked in the tree view.
Methods	Using different colors the bar 'Methods' shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted methods for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.
Bytes	Using different colors the bar 'Bytes' also shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted bytes for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.

You also have the option to separately assign the protection technologies *AxProtector* and *IxProtector* to single assembly elements, or exclude single elements from encrypting. To assign a protection technology by using the secondary menu, please proceed as follows:

1. In the left tree view, select the favored assembly element (name space, class, or method).
2. Click the right mouse button.
   The secondary menu opens.
3. Assign the favored encryption types by using symbols.

The License List IDs you are prompted are automatically transferred from the entries you added to the license list.

Symbol	Description
	Excludes the selected element from encryption.
	Encrypts the selected element using *AxProtector* (License List ID with a value of 0, i.e. default license).
	Encrypts the selected element using *IxProtector* (License List ID with a value unequal to 0, i.e. according to existing license list entries).
	Marks methods that are excluded from encryption due to the size of the method. The threshold can be set on the page 'Advanced Options' in the area optimizing.

The modifications you made instantly display in the left area.

7.5.2.5 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

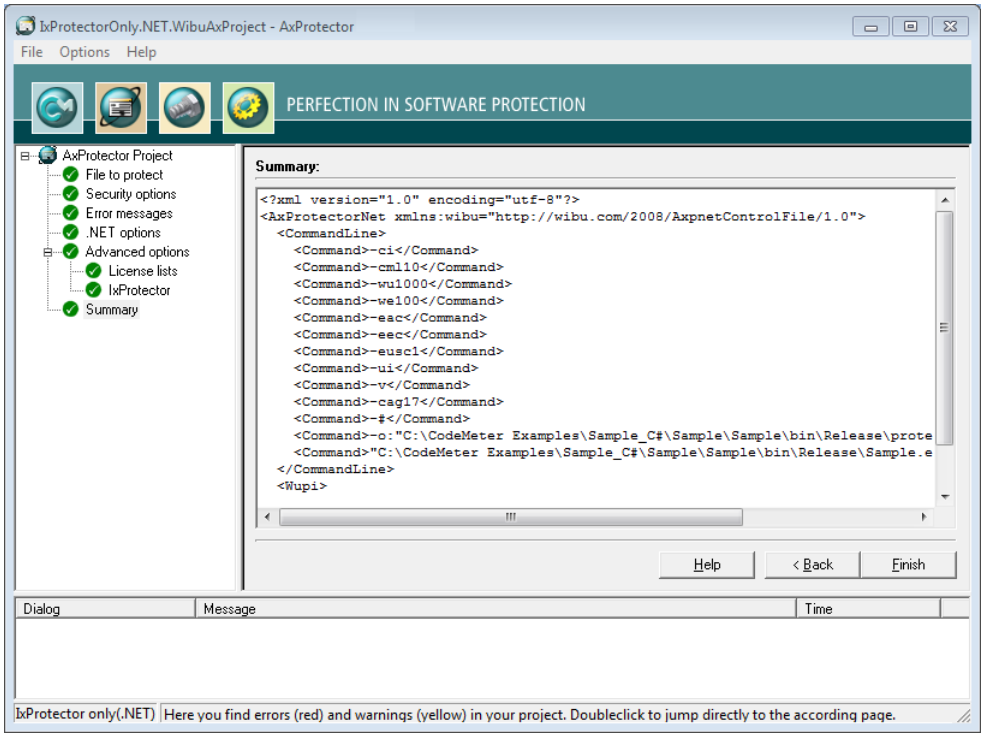


Figure 155: *AxProtector - IxProtector* only (.NET) "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

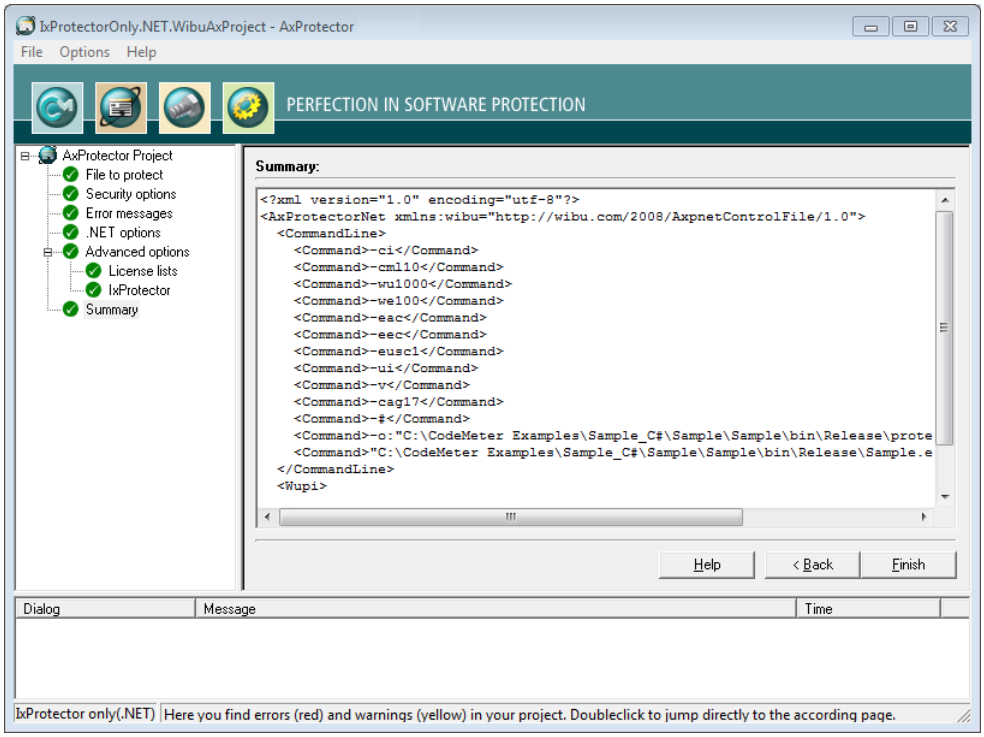


Figure 156: *AxProtector - IxProtector* only (.NET) "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.5.3 .NET Standard 2.0 Assembly

If you want to encrypt specified functions of an application using an index-based list, you select this project type. However, then the complete application is not additionally protected with *AxProtector*.

> Wibu-Systems recommends to use *IxProtector* within *AxProtector* if no other special requirements exist.

Then *IxProtector* finds the respective code areas and encrypts them. But even when you choose the project type increased security is fact, since *IxProtector* uses static code to be integrated later when the protected application is executed. The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
.NET Standard Assembly	IxProtector .NET Standard	√	.NET commandline [263] to be found in directory: `C:\Program Files (x86)\WIBU-SYSTEMS\AxProtector\Devkit\bin\net standard2.0`

The following menu items are available in the navigation windows:

- File to protect [217]
- Error Messages [218]
- Advanced Options [220]
  - License Lists [220]
  - IxProtector [224]
- Summary

7.5.3.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

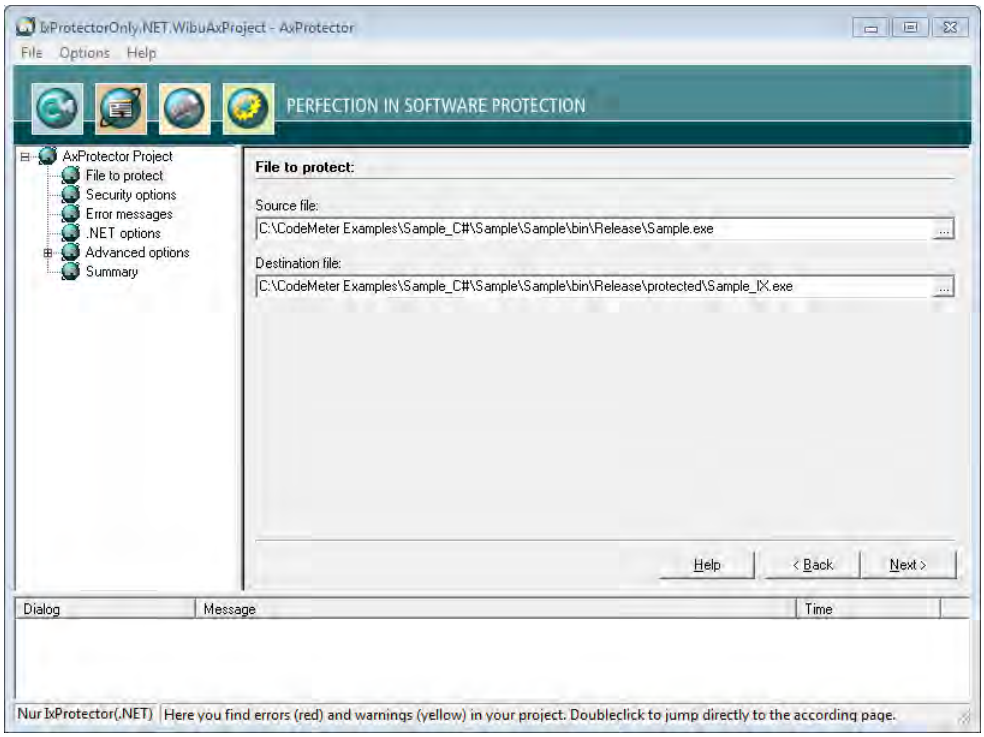


Figure 157: *AxProtector* - *IxProtector* only (.NET Standard) "File to Protect"

File tom protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application.

Element	Description
	Commandline option see here[279].

7.5.3.2 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

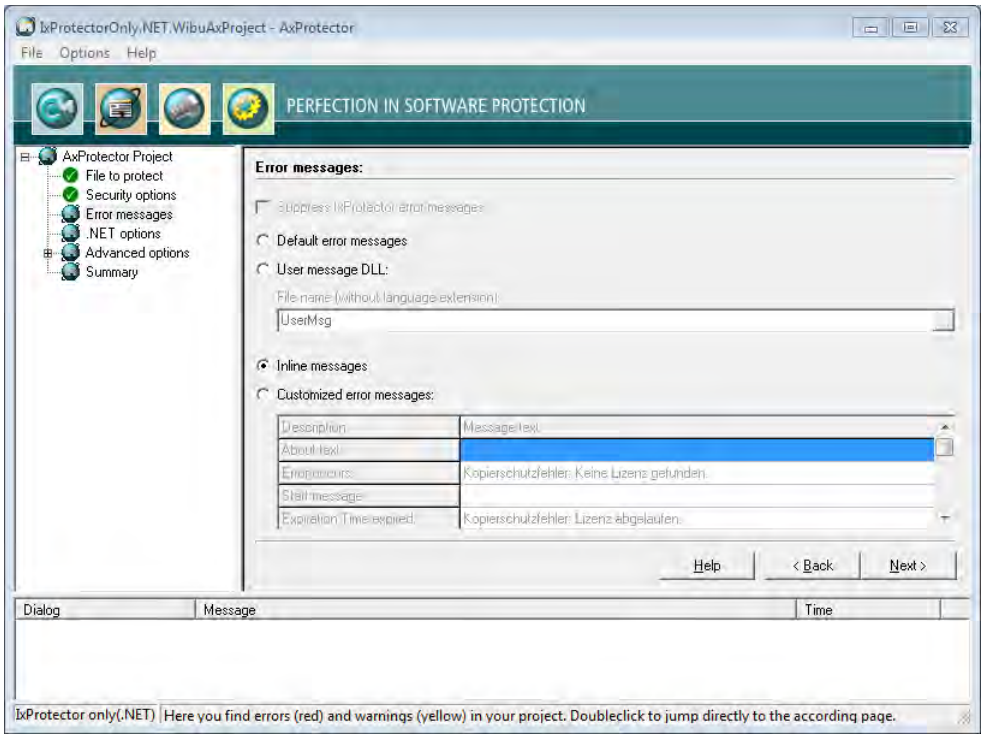
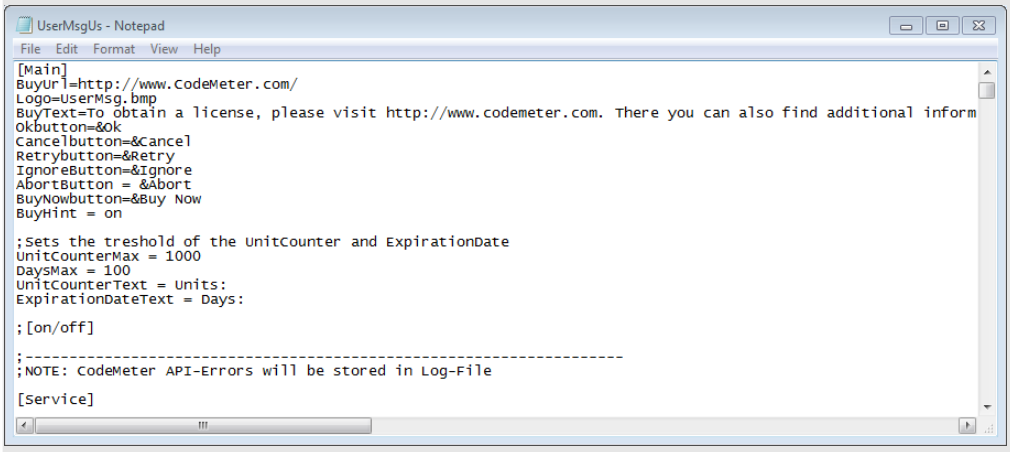


Figure 158: *AxProtector* - *IxProtector* only (.NET Standard) "Error Messages"

Error messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here[277]).
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text (commandline option see here[279]). The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.  Figure 159: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension. The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.

```
[Main]
BuyUrl=http://www.CodeMeter.com/
Logo=UserMsg.bmp
BuyText=To obtain a license, please visit http://www.codemeter.com. There you can also find additional inform
Okbutton=&Ok
Cancelbutton=&Cancel
Retrybutton=&Retry
IgnoreButton=&Ignore
AbortButton = &Abort
BuyNowbutton=&Buy Now
BuyHint = on

;Sets the treshold of the UnitCounter and ExpirationDate
UnitCounterMax = 1000
DaysMax = 100
UnitCounterText = Units:
ExpirationDateText = Days:

;[on/off]

;-----------------------------------------------------------------
;NOTE: CodeMeter API-Errors will be stored in Log-File

[Service]
```

Element	Description
Inline Messages	Links for .NET projects, with an inline assembly, can also be configured by `*.ini` files (commandline option see here[278]). When using Inline UserMessages the logging is saved to the directory "`%CommonApplicationData%`". When you want to specify another path specify the parameter `LogPath=<Pfad>` in the `*.ini` file.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.5.3.3 .NET Options

This page allows you to specify further .NET Standard settings.

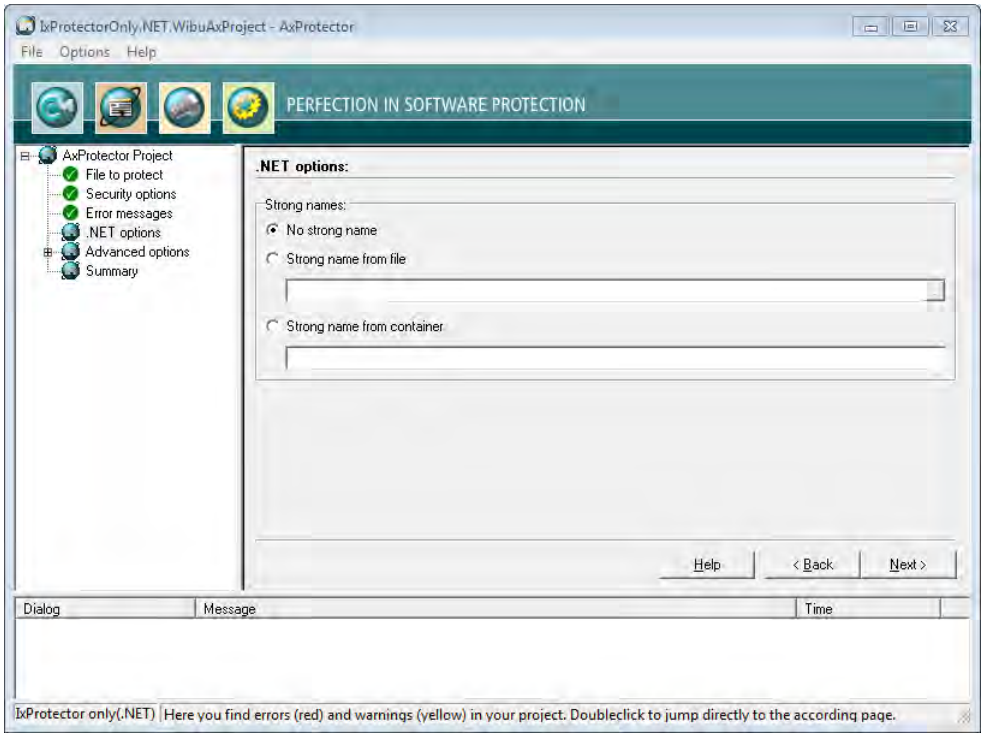


Figure 160: *AxProtector - IxProtector* only (.NET *Standard*) ".NET Options"

Here you are able to specify whether your assembly is signed by *AxProtector*.

Element	Description
No Strong Name	Activate this checkbox to not sign your assembly.
Strong Name from File	Activate this checkbox to use a source file to sign the program class. Then specify a file holding the key pair to generate a strong name (commandline option see here[279]).
Strong Name from Container	Activate this checkbox to use a container file to sign the program class (commandline option see here[279]).

7.5.3.4 Advanced Options

This input window lets you set further encryption options.

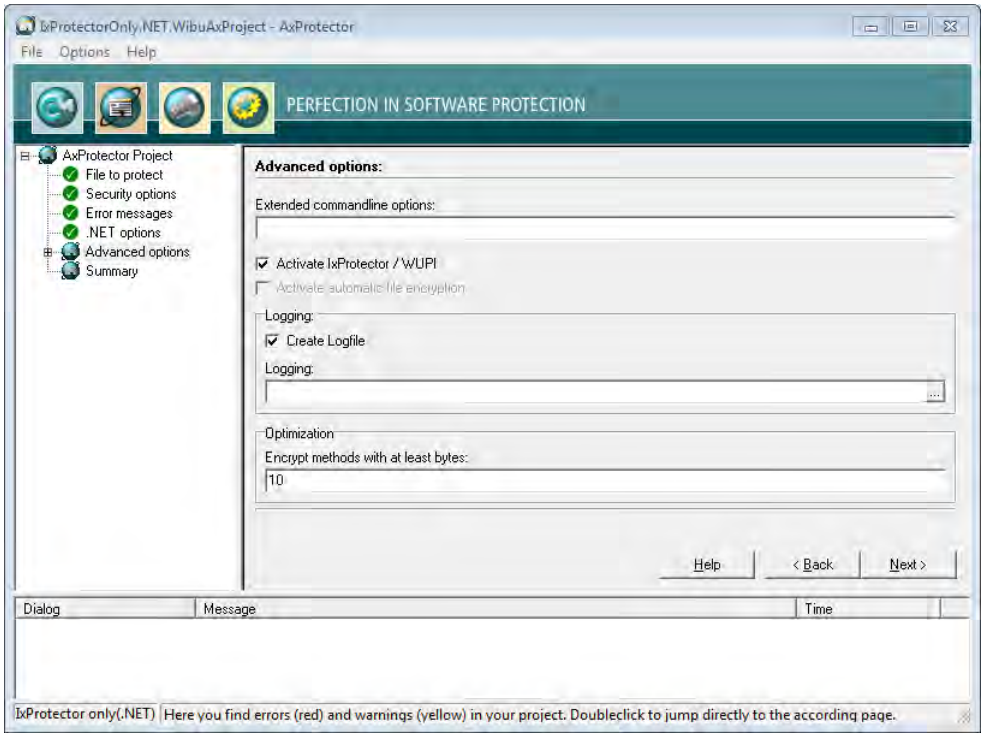


Figure 161: *AxProtector - IxProtector* only (.NET Standard) "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Activate IxProtector	Activate this checkbox to allow for the later creation and editing of license lists and function lists. These you need to protect using *IxProtector* via the *Software Protection-API* 289. (commandline option here 274).
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.
Optimizing	For an optimized performance specify here the minimum size for assemblies to be encrypted. The default setting is 10 bytes. This way you are able to exclude methods from encryption which are smaller than the number of bytes you specify here. By setting a value of 0 this feature is deactivated. Commandline option see here 275).

7.5.3.4.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .

License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

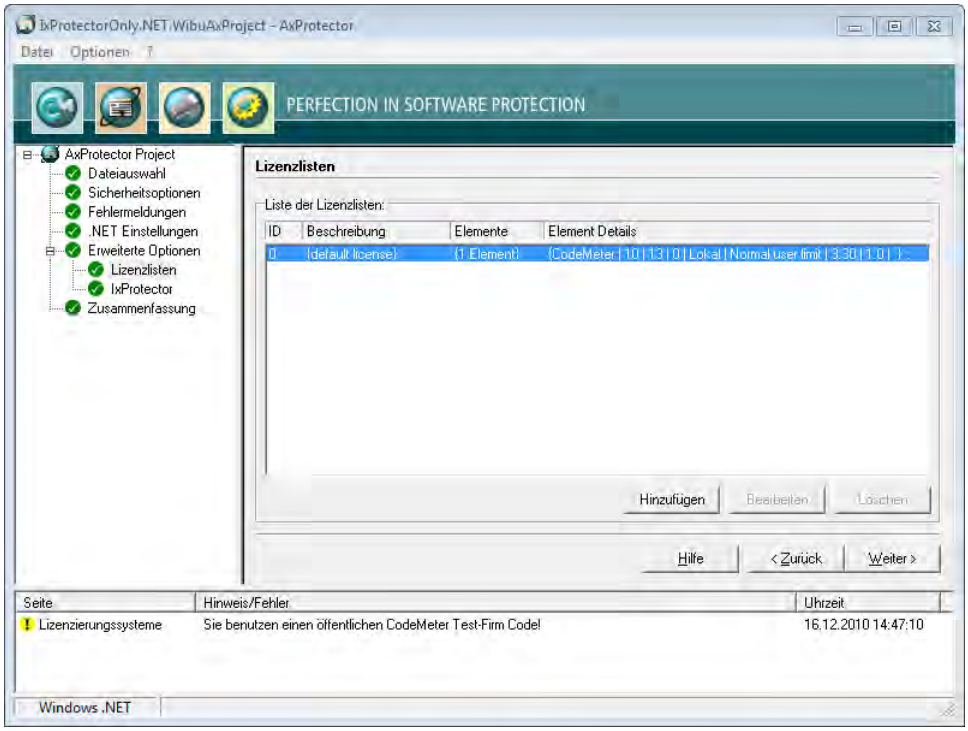
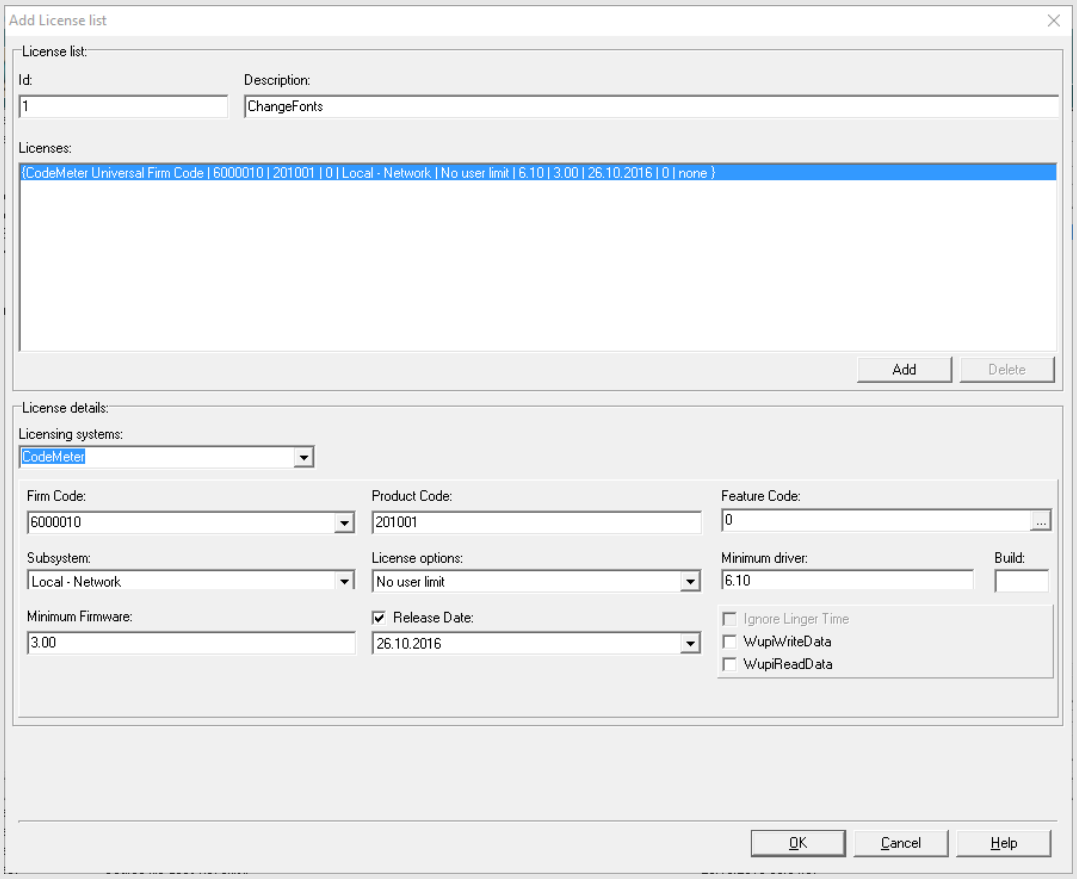


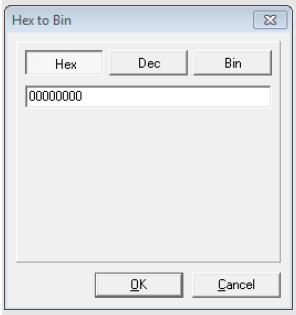
Figure 162: *AxProtector - IxProtector* only (.NET Standard) "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 163: *AxProtector - IxProtector* Only (.NET)' - "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** / **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . • Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". • If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. • In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Enter the Firm Code used for the protection of the license.
Product Code	Enter the Product Code used for the protection of the license.
Feature Code	Enter the Feature Code used, for example, to encrypt different versions of your application.

Element	Description
	Using the **"..."** button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. 
Subsystem	Select the subsystem in which the protected application is to search (local or network), and define the search order. **License Options** Select the options for license allocation: • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • *No User limit*
Minimum Driver Version	Specify the required minimum driver version for the protected application.
Release Date	Starting with Firmware version 1.18 *CodeMeter®* supports the *Product Item Option Maintenance Period*. In the PIO two date values are stored: a start and an end value. This allows you to implement license models which map the granting of support and services when using the software. Then the use of a license is limited to software versions, corrections, and extension 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. In the case the *Release Date* is not within the *Maintenance Period*, the use of the software is not covered by the license. To store the *Release Date*, please proceed as follows: **1.** Activate the **"Release Date"** checkbox to type in the *Release Date*.The current date is preset. **2.** Change if desired the date either directly in the field located below or use the calendar element which opens via the arrow button at the left margin of the field.
Minimum Firmware	Specify the minimum firmware version required. In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.
Ignore Linger Time	Activate this option to ignore a programmed *LingerTime*. This license option allows to define an allocation time of the license after a protected application has been released or the protected application has been closed (more Information in the *CodeMeter* Developer Guide).
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.

5. Click the **"OK"** button. The new license data is added to the license list.

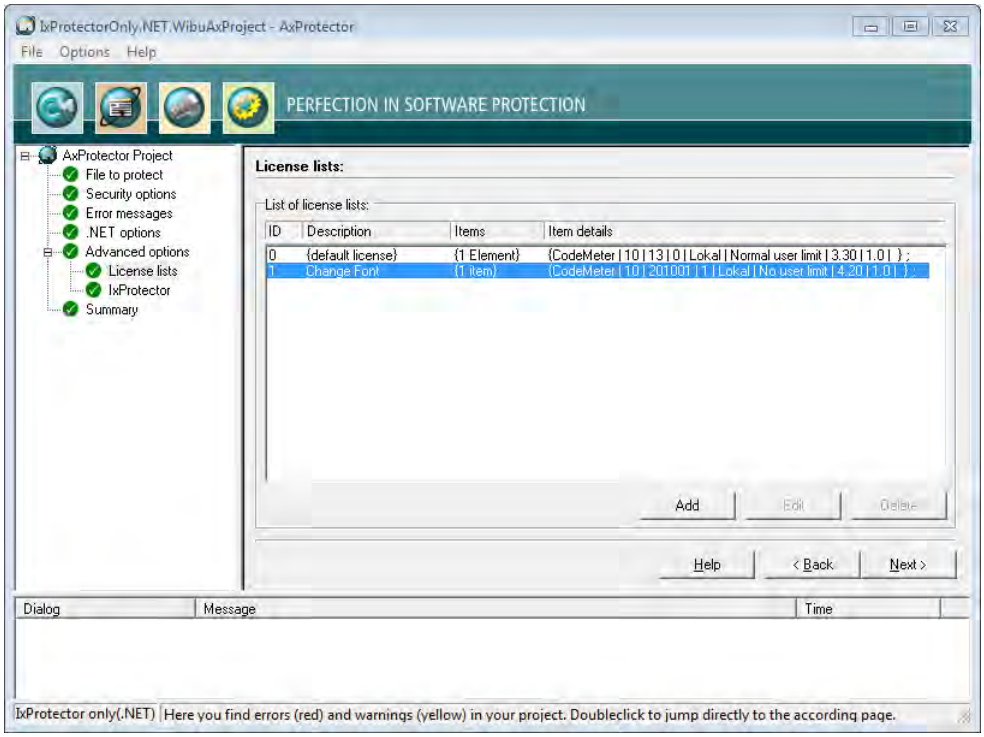


Figure 164: *AxProtector* - *IxProtector* only (.NET Standard) - "Completed License List"

7.5.3.4.2 IxProtector

Using this menu item allows you to separately define single encryption types for single assembly elements.

In the case you activated the checkbox **"IxProtector"** in the menu item **"Advanced options"** the source assembly is loaded and displayed in a tree view making available all name spaces, classes, and modules.

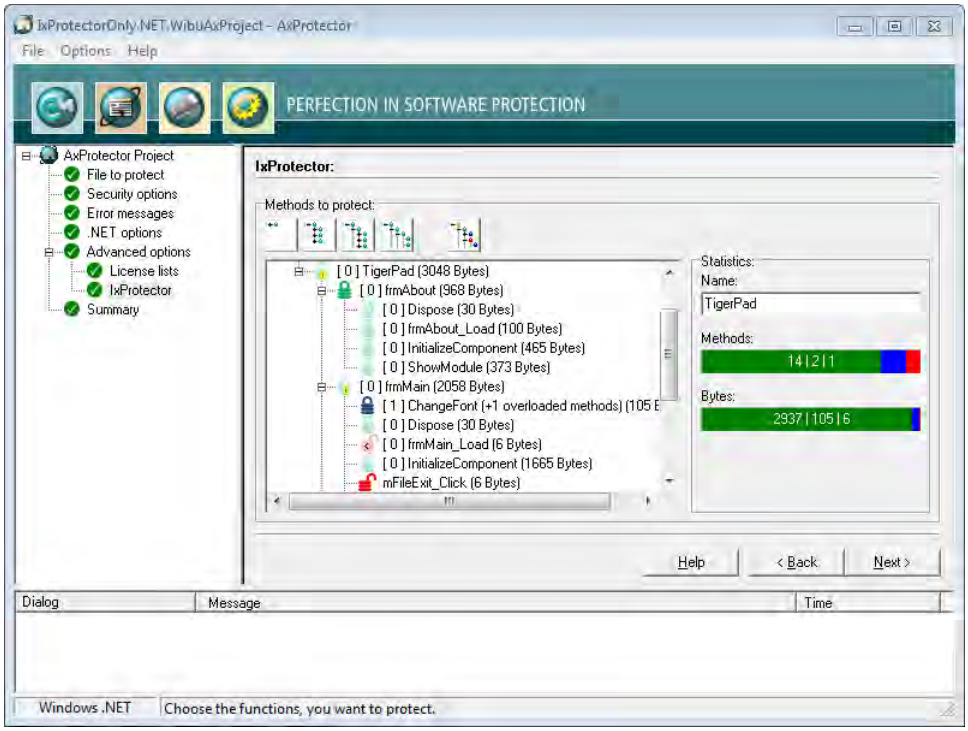


Figure 165: *AxProtector* - *IxProtector* only (.NET Standard) " IxProtector"

Click the different buttons in the upper "IxProtector" area to select from different assembly views.

Views

Button	Description
	Closes all assembly levels of the tree structure.
	Expands the name space level of the assembly.

Button	Description
	Expands the class level of the assembly.
	Expands the method level of the assembly.
	Expands all parent levels of the assembly. In this view see all levels where modifications have been made.

The area "Statistics" on the right shows you more encryption details depending on the selection you have made for the tree view.

Element	Description
Name	This field refers to the name of the element you have marked in the tree view.
Methods	Using different colors the bar 'Methods' shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted methods for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.
Bytes	Using different colors the bar 'Bytes' also shows you the protection technology used or not used when encrypting or not encrypting. At the same time, the displayed numbers inform you about the number of encrypted or non-encrypted bytes for each protection technology.
	Color / **Description**
	Green: Shows that the method will be encrypted using *AxProtector* and that the License List ID has a value of 0 (default license)
	Blue: Shows that the method will be encrypted using *IxProtector* and that the License List ID has a value unequal 0.
	Red: Shows that the method in not encrypted.

You also have the option to separately assign the protection technologies *AxProtector* and *IxProtector* to single assembly elements, or exclude single elements from encrypting. To assign a protection technology by using the secondary menu, please proceed as follows:

1. In the left tree view, select the favored assembly element (name space, class, or method).
2. Click the right mouse button.
   The secondary menu opens.
3. Assign the favored encryption types by using symbols.

The License List IDs you are prompted are automatically transferred from the entries you added to the license list.

Symbol	Description
	Excludes the selected element from encryption.
	Encrypts the selected element using *AxProtector* (License List ID with a value of 0, i.e. default license).
	Encrypts the selected element using *IxProtector* (License List ID with a value unequal to 0, i.e. according to existing license list entries).
	Marks methods that are excluded from encryption due to the size of the method. The threshold can be set on the page 'Advanced Options' in the area optimizing.

The modifications you made instantly display in the left area.

7.5.3.5 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

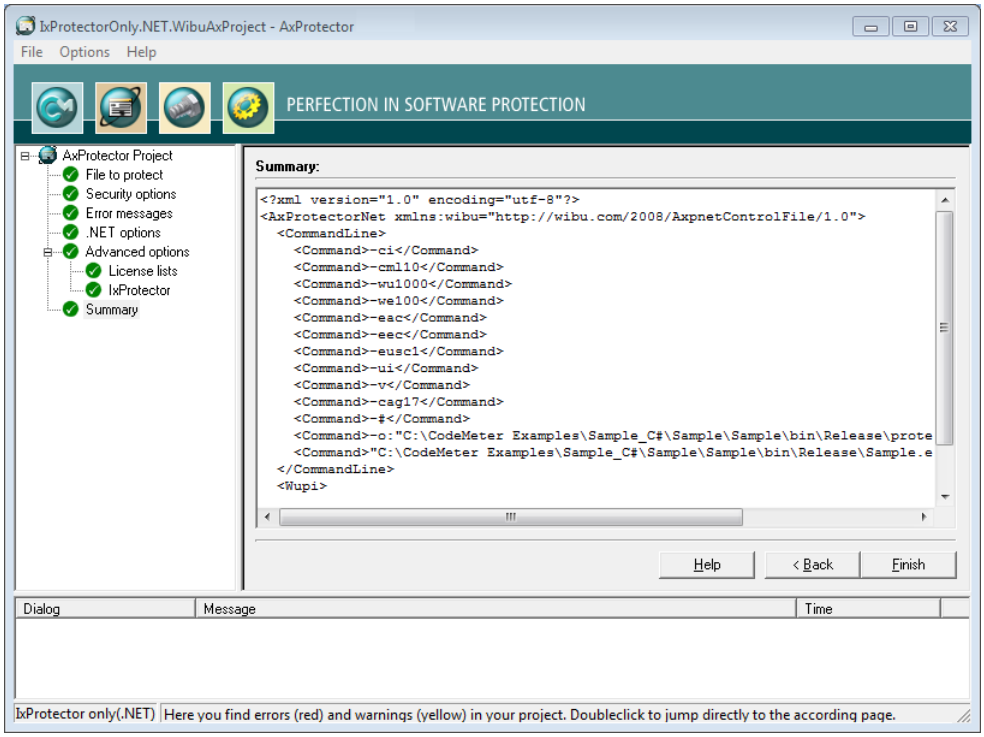


Figure 166: *AxProtector* - *IxProtector* only (.NET Standard) "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

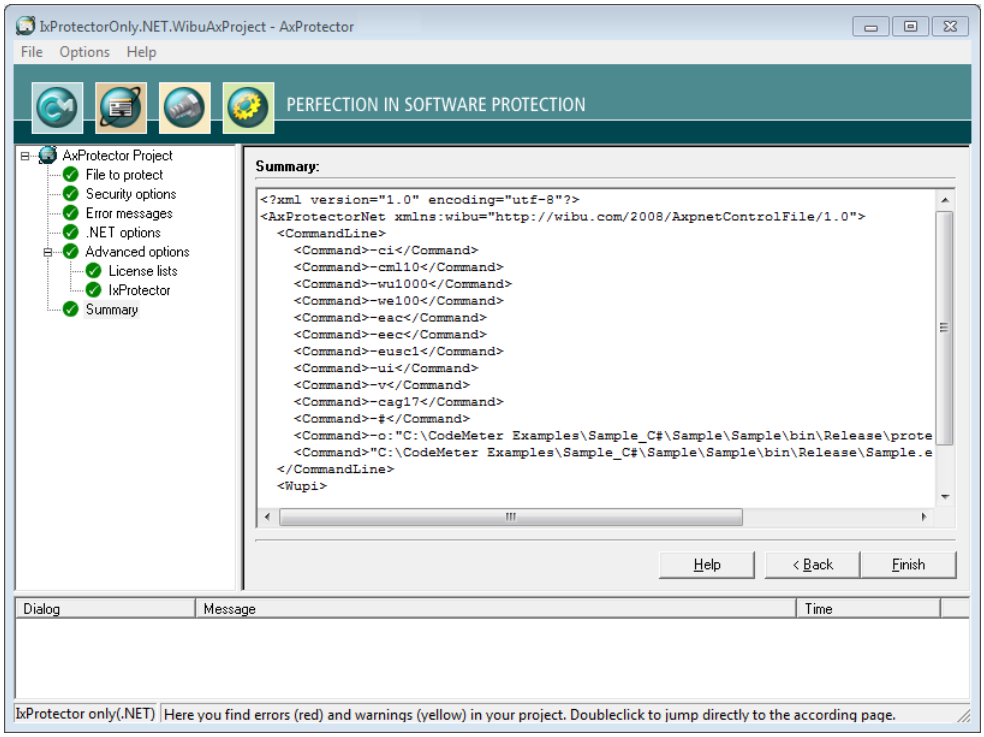


Figure 167: *AxProtector* - *IxProtector* only (.NET Standard) "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.5.4 macOS Application or Dylib

When you want to encrypt specified functions of an application using an index-based list, you select this project type. However, then the complete application is not additionally protected with *AxProtector*.

> Wibu-Systems recommends to use *IxProtector* within *AxProtector*, if no other special requirements exist.

Then *IxProtector* finds the respective code areas and encrypts them. But even when you choose the project type increased security is fact, since *IxProtector* uses static code to be integrated later when the protected application is executed.The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
macOS Application or Dylib	IxProtector macOS	√	Windows commandline [263] In a separate commandline for macOS, running on macOS operating systems, you are also able to insert encryption parameter [236] .

The following menu items are available in the navigation windows:

- File to protect [227]
- Error Messages [228]
- Advanced Options [229]
  - License Lists [229]
  - IxProtector [233]
- Summary

7.5.4.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

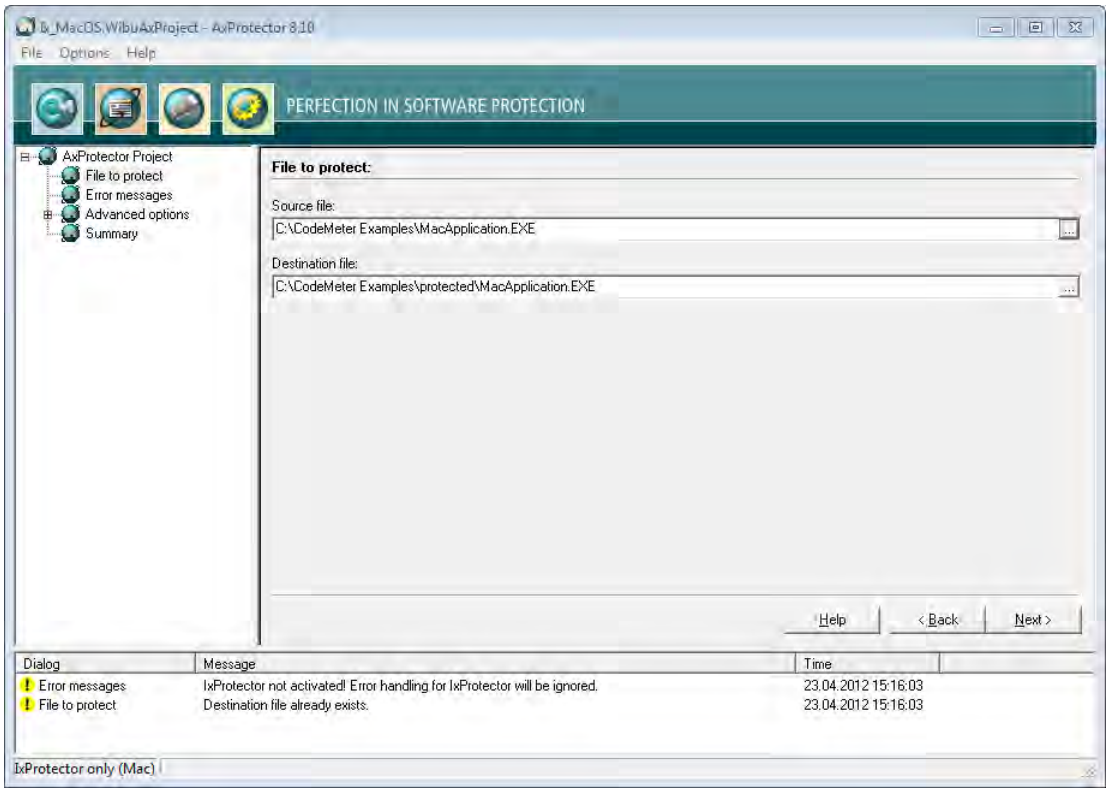


Figure 168: *IxProtector* Mac "File to Protect"

File to protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.

Element	Description
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279 .

7.5.4.2 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

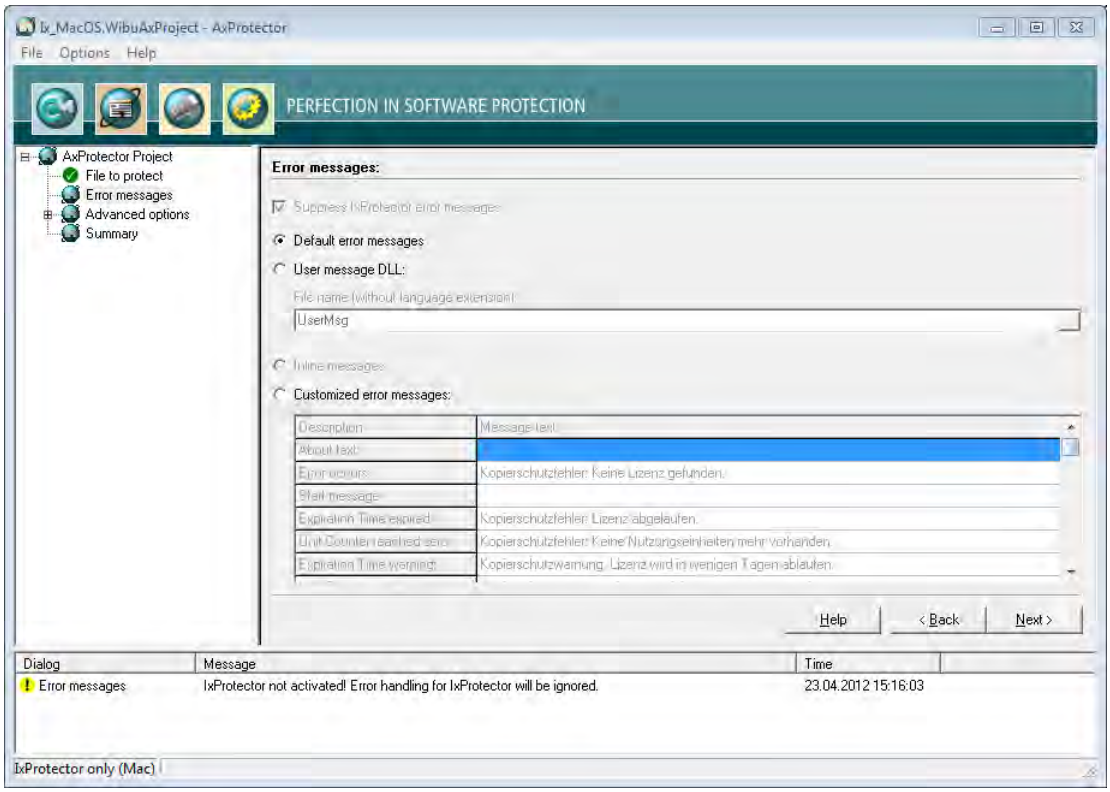
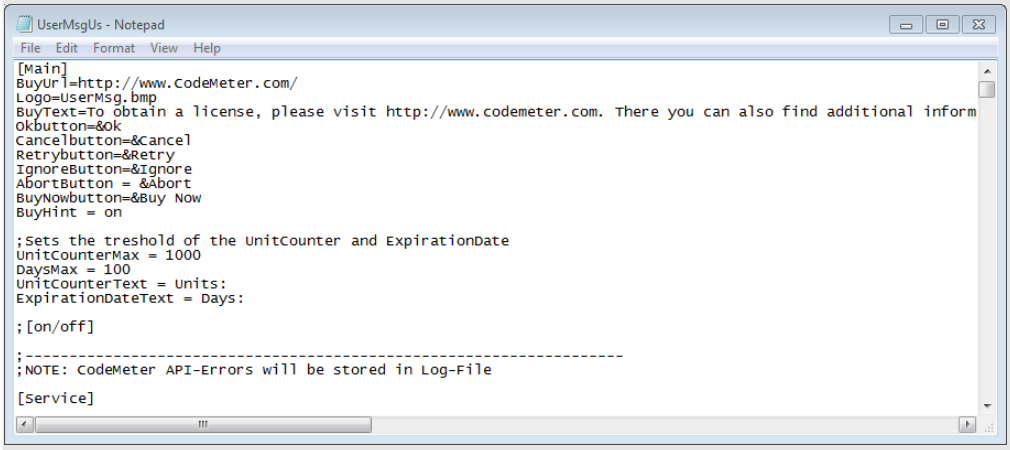


Figure 169: *IxProtector* Mac "Error Messages"

Error Messages

Element	Description
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages.
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text. The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.  Figure 170: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension.

```
[Main]
BuyUrl=http://www.CodeMeter.com/
Logo=UserMsg.bmp
BuyText=To obtain a license, please visit http://www.codemeter.com. There you can also find additional inform
Okbutton=&Ok
Cancelbutton=&Cancel
Retrybutton=&Retry
IgnoreButton=&Ignore
AbortButton = &Abort
BuyNowbutton=&Buy Now
BuyHint = on

;Sets the treshold of the UnitCounter and ExpirationDate
UnitCounterMax = 1000
DaysMax = 100
UnitCounterText = Units:
ExpirationDateText = Days:

;[on/off]

;-----------------------------------------------------------------
;NOTE: CodeMeter API-Errors will be stored in Log-File

[Service]
```

Element	Description
	The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.5.4.3 Advanced Options

This input window lets you set further encryption options.

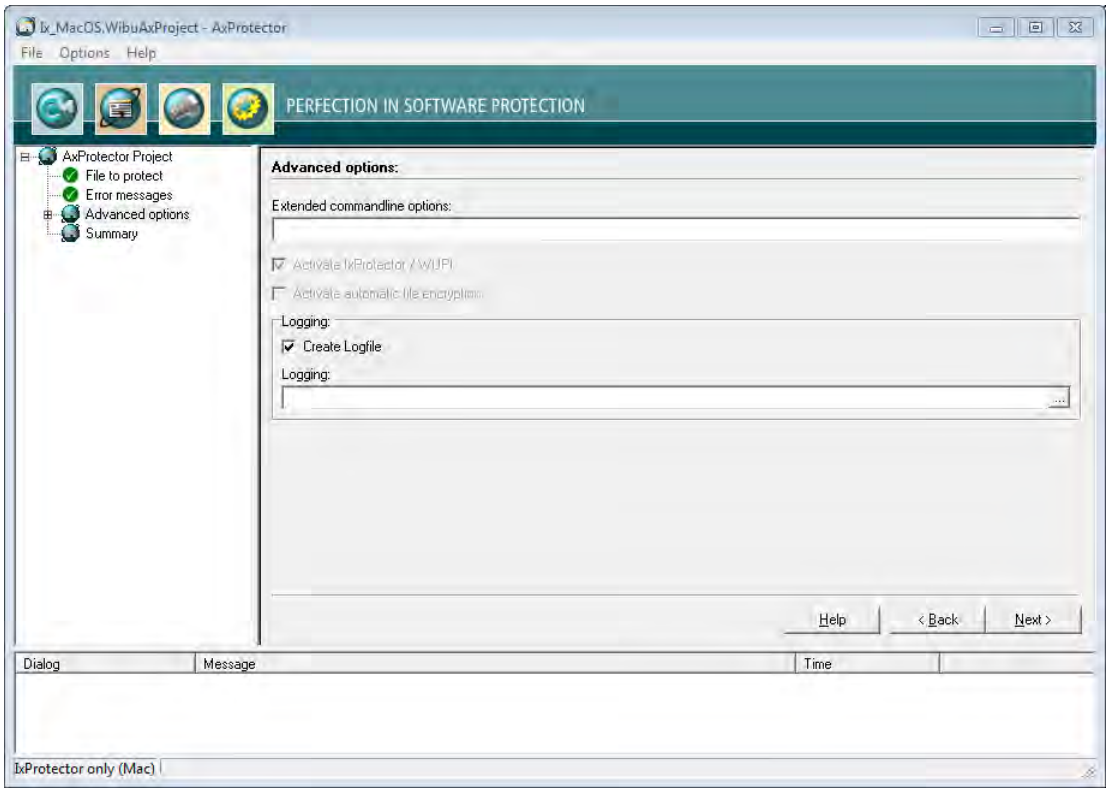


Figure 171: *IxProtector* Mac "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Dynamic loading of Wibu-Systems libraries	When activated this checkbox results in a special, more time-intensive process. This when VB6 applications or dynamic loading of Wibu-Systems libraries are involved.
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.5.4.3.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* 289 .
License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* 290 .

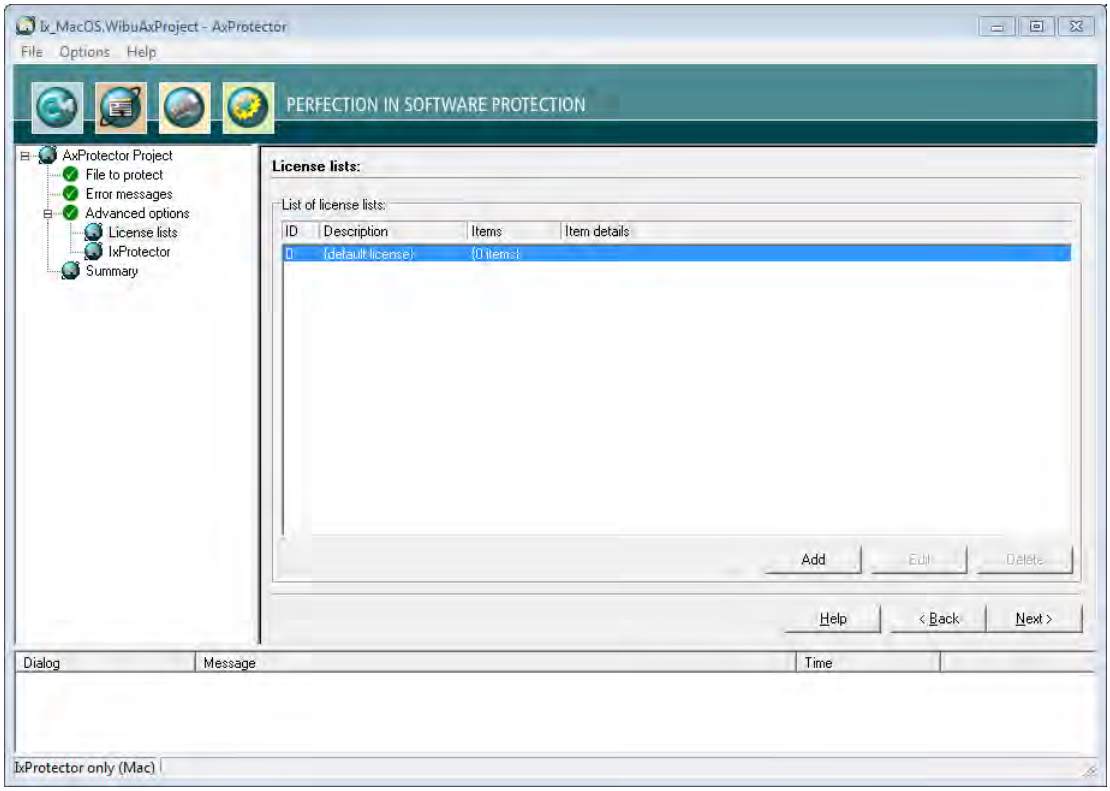
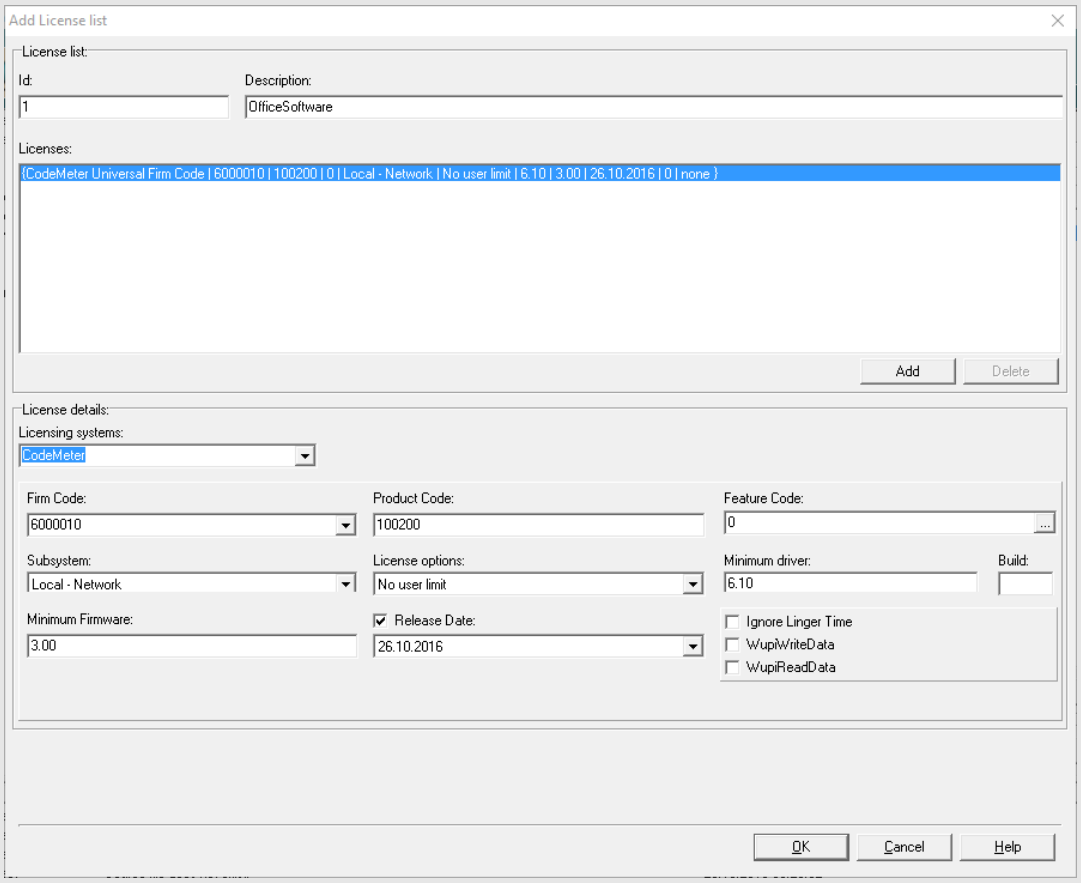


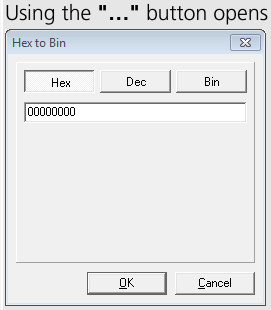
Figure 172: *IxProtector* Mac "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **ID**s starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 173: *IxProtector* Mac "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Enter the Firm Code used for the protection of the license.
Product Code	Enter the Product Code used for the protection of the license.
Feature Code	Enter the Feature Code used, for example, to encrypt different versions of your application.

Element	Description
	Using the **"..."** button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. 
Subsystem	Select the subsystem in which the protected application is to search (local or network), and define the search order. **License Options** Select the options for license allocation: • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • *No User limit*
Minimum Driver Version	Specify the required minimum driver version for the protected application.
Release Date	Starting with Firmware version 1.18 *CodeMeter®* supports the *Product Item Option Maintenance Period*. In the PIO two date values are stored: a start and an end value. This allows you to implement license models which map the granting of support and services when using the software. Then the use of a license is limited to software versions, corrections, and extension 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. In the case the *Release Date* is not within the *Maintenance Period*, the use of the software is not covered by the license. To store the *Release Date*, please proceed as follows: **1.** Activate the **"Release Date"** checkbox to type in the *Release Date*.The current date is preset. **2.** Change if desired the date either directly in the field located below or use the calendar element which opens via the arrow button at the left margin of the field.
Minimum Firmware	Specify the minimum firmware version required. In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.
Ignore Linger Time	Activate this option to ignore a programmed *LingerTime*. This license option allows to define an allocation time of the license after a protected application has been released or the protected application has been closed (more Information in the *CodeMeter* Developer Guide).
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.

5. Click the **"OK"** button. The new license data is added to the license list.

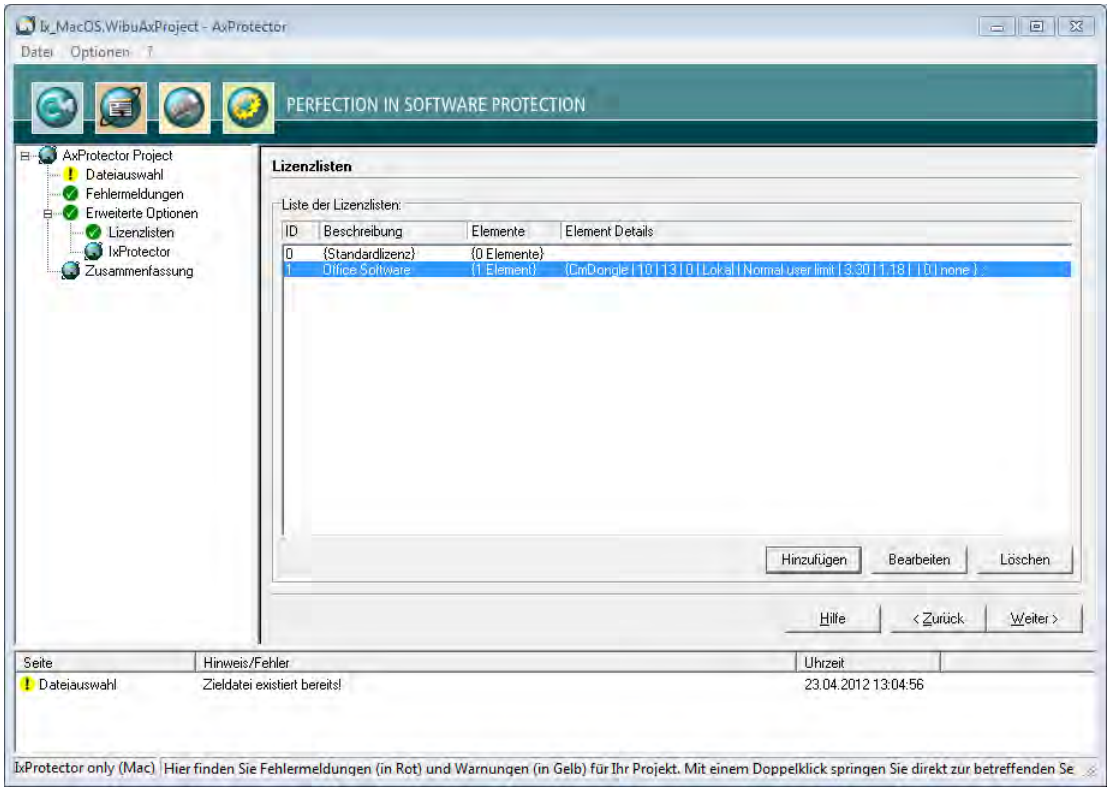


Figure 174: *IxProtector* Mac "Completed License Lists"

7.5.4.3.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

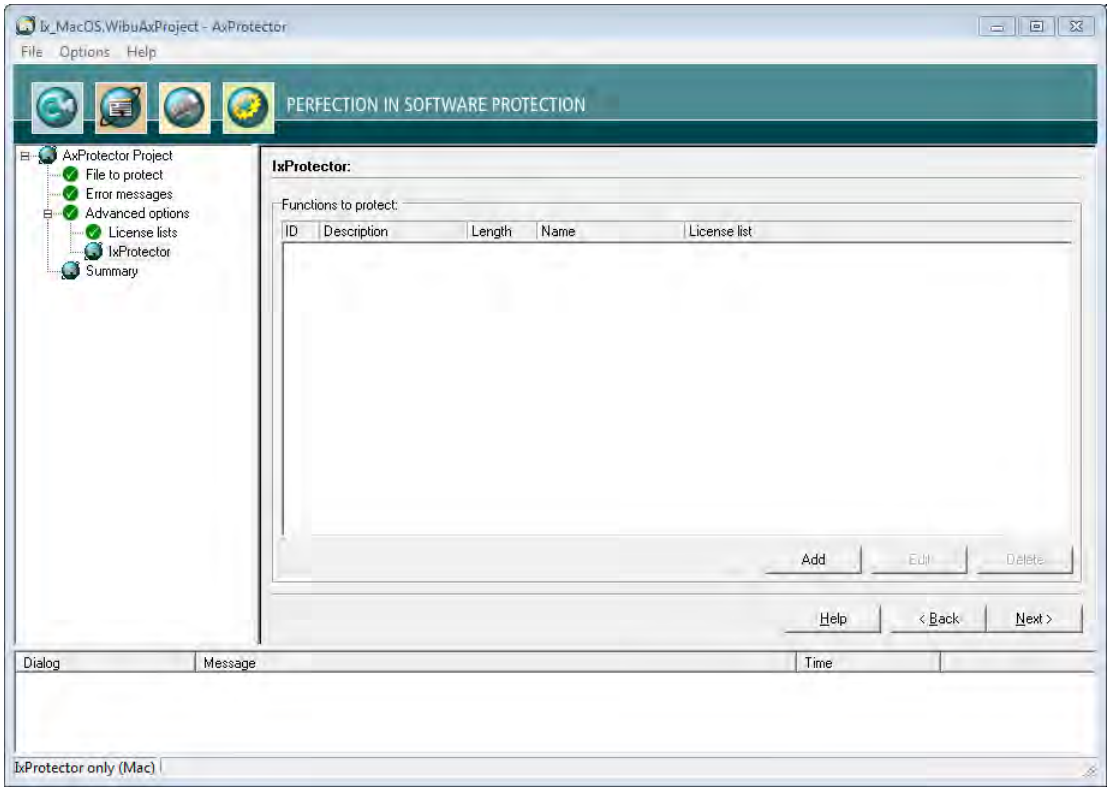
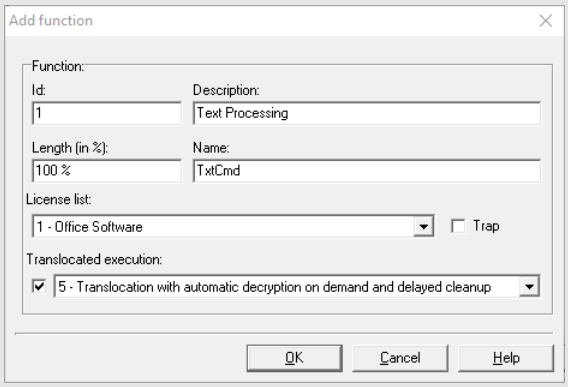


Figure 175: *IxProtector* Mac "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: **1.** Click the **"Add"** button in the group "IxProtector Options". **2.** Define the function by completing the fields in the "Function" group.  Figure 176: *IxProtector* Mac "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291 .
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. • Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td>License List</td><td>Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.</td></tr><tr><td>Trap</td><td>Activates the trap function for the function.</td></tr><tr><td>Translocated execution</td><td>Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.<table><tr><th>Option</th><th>Description</th></tr><tr><td>1</td><td>Translocation with automatic decrpytion on demand and cleanup.</td></tr><tr><td>2</td><td>Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).</td></tr><tr><td>5</td><td>Translocation with automatic decrpytion on demand and delayed cleanup. (Default)</td></tr></table>Command line option see here 286 .</td></tr></table> **3.** Click the **"OK"** button. The new functions are added to the function list.

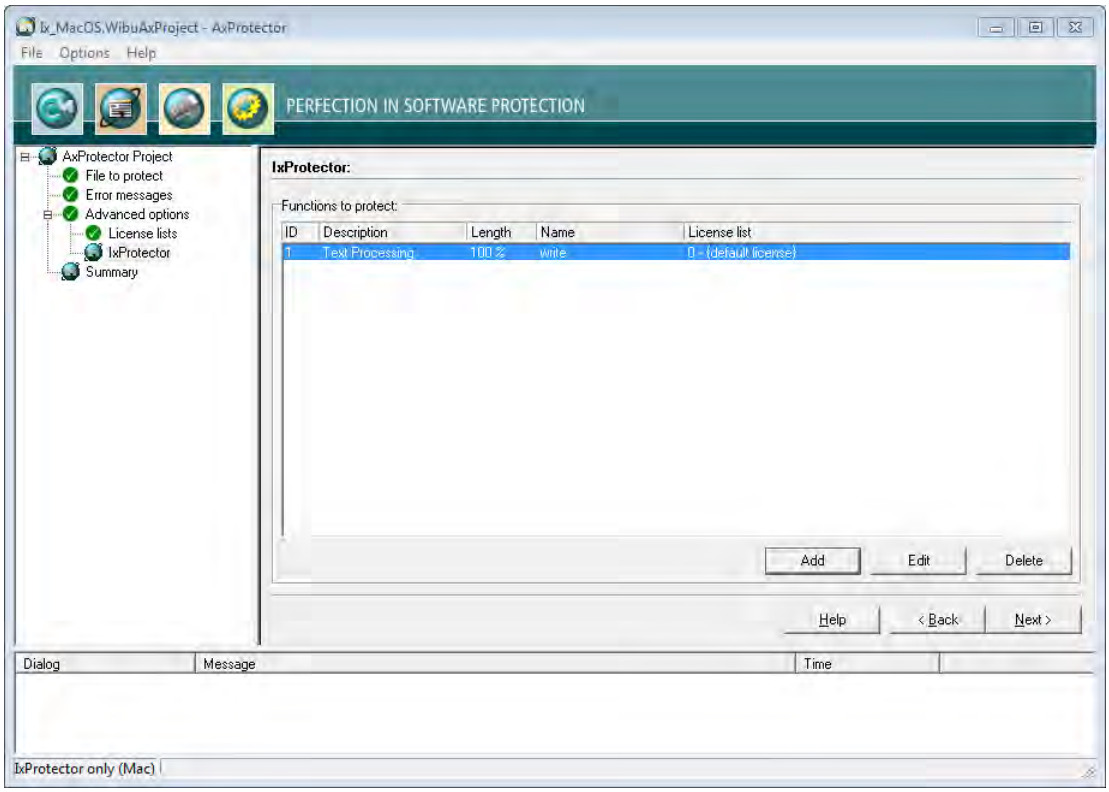


Figure 177: *IxProtector* Mac "Completed Function List"

7.5.4.4 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

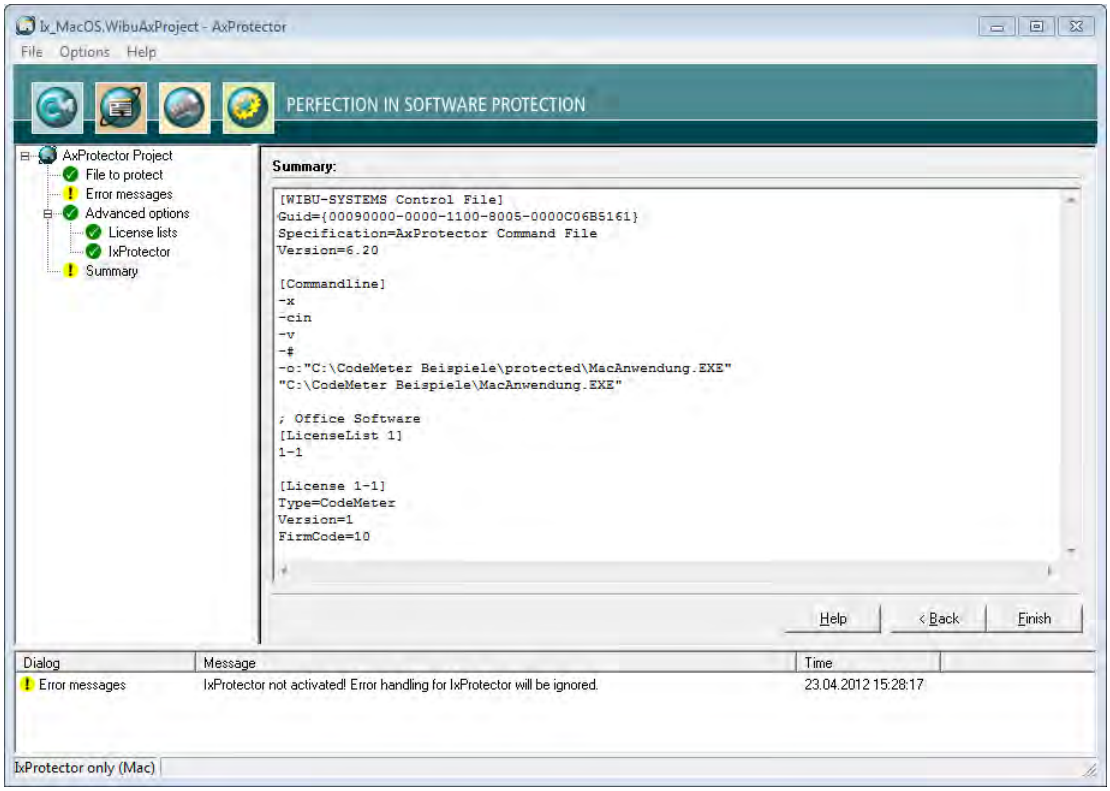


Figure 178: *AxProtector - IxProtector* only Mac "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

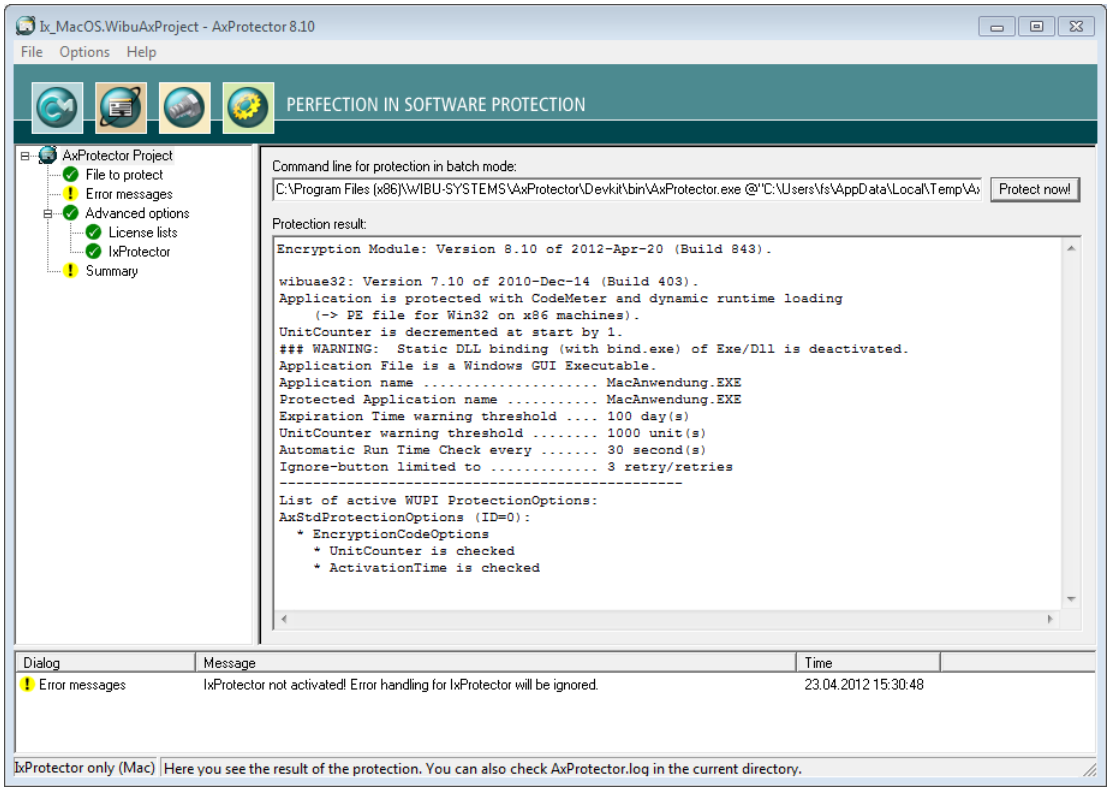


Figure 179: *AxProtector - IxProtector* only Mac "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode.

Element	Description
	You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.5.5 Linux Application or Shared Object

When you want to encrypt specified functions of an application using an index-based list, you select this project type. However, then the complete application is not additionally protected with *AxProtector*.

Wibu-Systems recommends to use *IxProtector* within *AxProtector* if no other special requirements exist.

Then *IxProtector* finds the respective code areas and encrypts them. But even when you choose the project type increased security is fact, since *IxProtector* uses static code to be integrated later when the protected application is executed.

The following table summarizes what kind of files can be encrypted using the *AxProtector* Windows GUI or the commandline.

Application to be protected	Project type	GUI Windows	Commandline
Linux Application or Shared Object	IxProtector Linux	√	Windows commandline [263] In a separate commandline for Linux, running on Linux operating systems, you are also able to insert encryption parameter [245] .

The following menu items are available in the navigation windows:

- File to protect [237]
- Error Messages [238]
- Advanced Options [239]
  - License Lists [240]
  - IxProtector [243]
- Summary

7.5.5.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

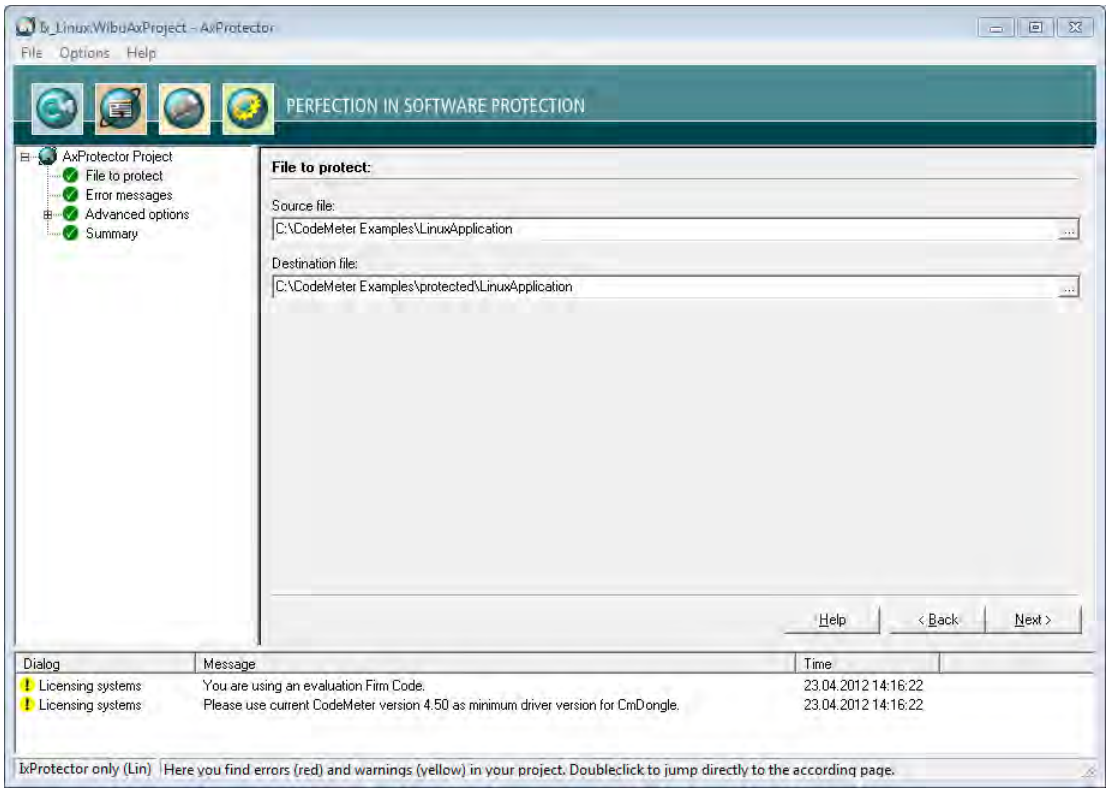


Figure 180: *IxProtector* Linux "File to Protect"

File to protect

Element	Description
Source File	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279 .

7.5.5.2 Error Messages

This input window lets you define the messages displayed if errors occur. You define whether a user message DLL with a separate error display is used, or whether you use default error message windows.

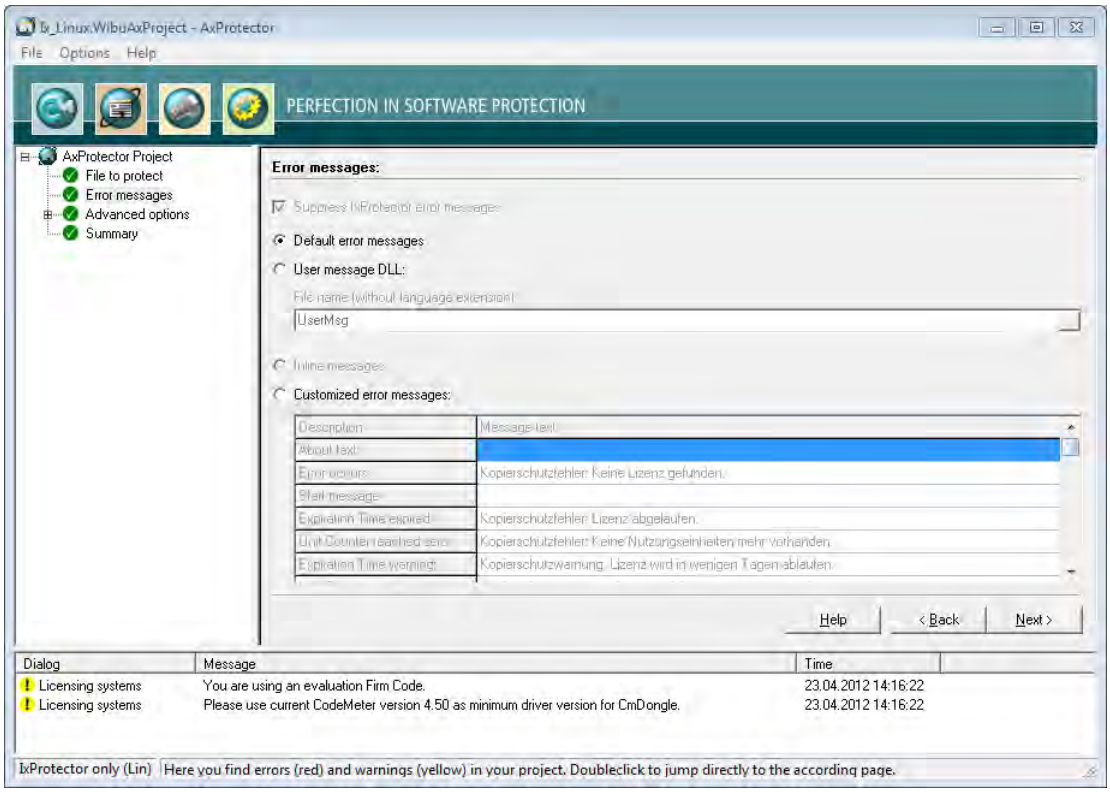
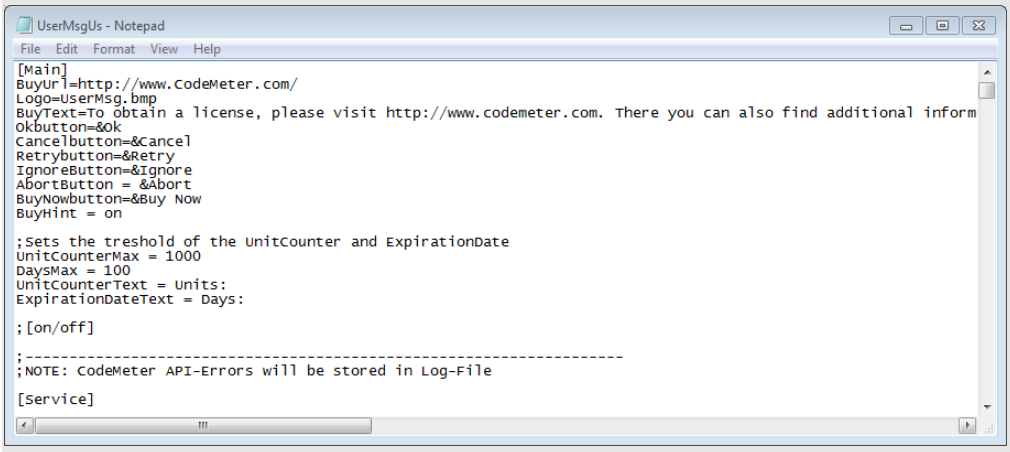


Figure 181: *IxProtector* Linux "Error Messages"

Error Messages

Element	Description
Suppress IxProtector Error Messages	The output of *IxProtector* error messages is suppressed (commandline option see here 273). If you do not activate this option, when using *IxProtector* errors, additional message windows are displayed along with the messages you program in the project.
Default Error Messages	All errors occurring at the runtime of a protected application display default error messages (commandline option see here 277).
User Message DLL	The ability to use the User Message DLL is activated. Error messages can be localized to different languages using `*.ini` files. In addition, you have the option to integrate your own designs to this file, for example, by using separate logos or text (commandline option see here 278). The `*.ini` files with the respective country suffix and the Dll program library are automatically saved to the directory where the application locates the files protected by *AxProtector*.

Element	Description
	 Figure 182: *AxProtector* – UserMsgUs.ini **File name (without Language Extension)** Enter the file name without specifying path and language file extension. The UserMsgDll is copied from the directory `%Programm Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage`. The corresponding `*.ini` files are also saved to this directory.
Inline Messages	Links for .NET projects, with an inline assembly which can also be configured by `*.ini` files (commandline option see here 278). This option is available for the encryption of .NET applications only.
Customized Error Messages	Activate this option to enter customized error messages displayed in the message boxes below.

7.5.5.3 Advanced Options

This input window lets you set further encryption options.

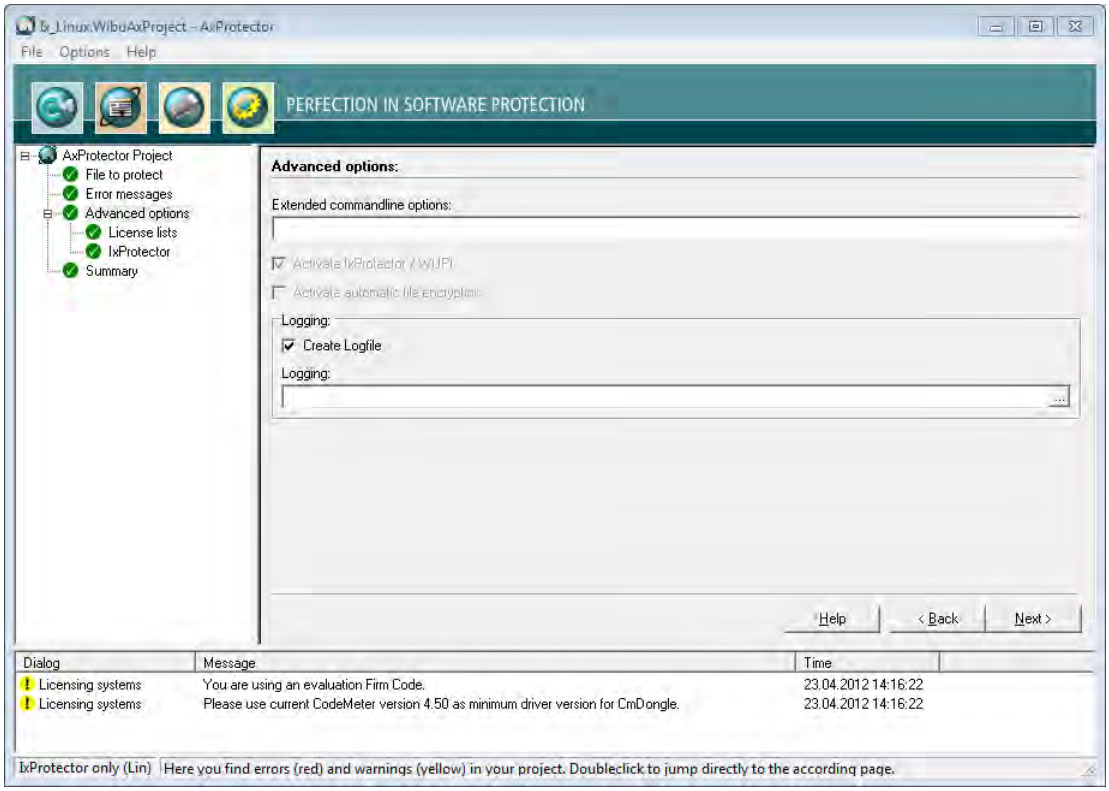


Figure 183:*IxProtector* Linux "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.

Element	Description
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.5.5.3.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* [289] .
License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* [290] .

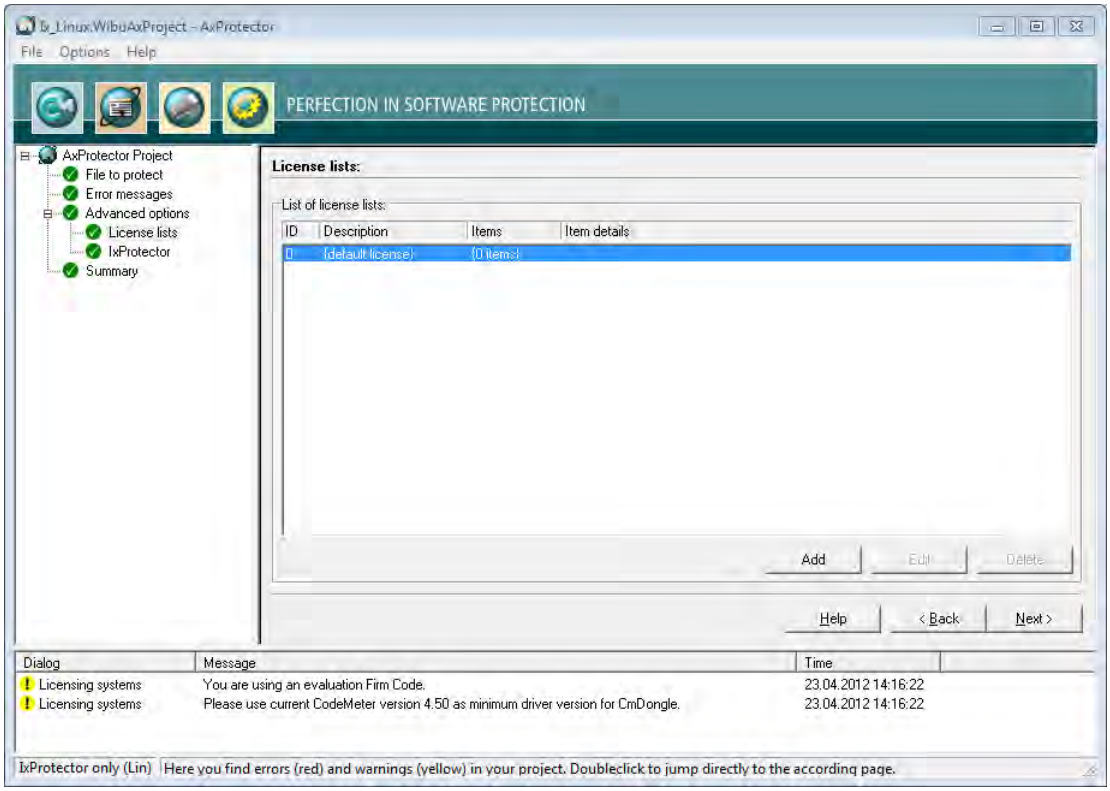
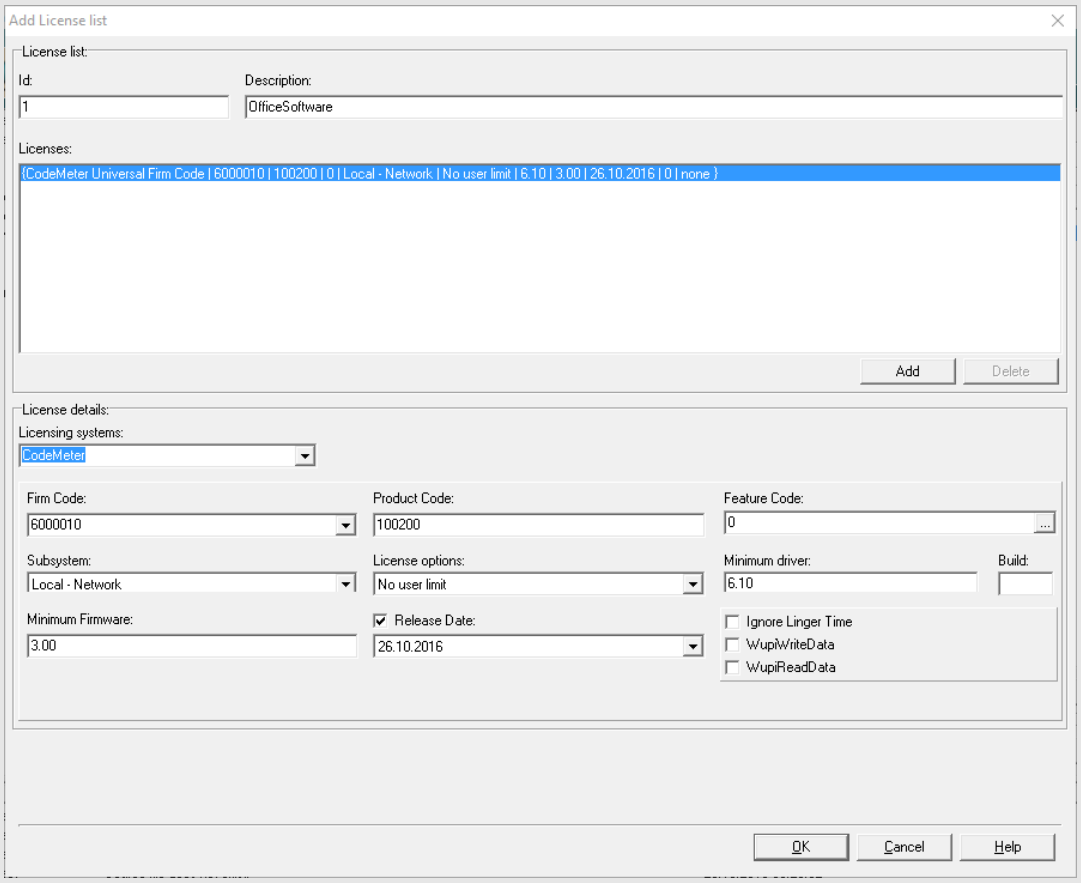


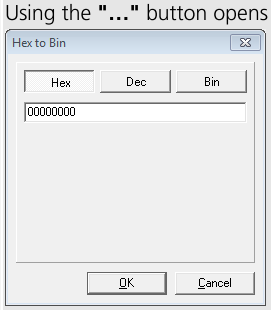
Figure 184: *IxProtector* Linux "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 185: *IxProtector* Linux "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264. Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Enter the Firm Code used for the protection of the license.
Product Code	Enter the Product Code used for the protection of the license.
Feature Code	Enter the Feature Code used, for example, to encrypt different versions of your application.

Element	Description
	Using the **"..."** button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. 
Subsystem	Select the subsystem in which the protected application is to search (local or network), and define the search order. **License Options** Select the options for license allocation: • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • *No User limit*
Minimum Driver Version	Specify the required minimum driver version for the protected application.
Release Date	Starting with Firmware version 1.18 *CodeMeter®* supports the *Product Item Option Maintenance Period*. In the PIO two date values are stored: a start and an end value. This allows you to implement license models which map the granting of support and services when using the software. Then the use of a license is limited to software versions, corrections, and extension 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. In the case the *Release Date* is not within the *Maintenance Period*, the use of the software is not covered by the license. To store the *Release Date*, please proceed as follows: **1.** Activate the **"Release Date"** checkbox to type in the *Release Date*.The current date is preset. **2.** Change if desired the date either directly in the field located below or use the calendar element which opens via the arrow button at the left margin of the field.
Minimum Firmware	Specify the minimum firmware version required. In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.
Ignore Linger Time	Activate this option to ignore a programmed *LingerTime*. This license option allows to define an allocation time of the license after a protected application has been released or the protected application has been closed (more Information in the *CodeMeter* Developer Guide).
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.

5. Click the **"OK"** button. The new license data is added to the license list.

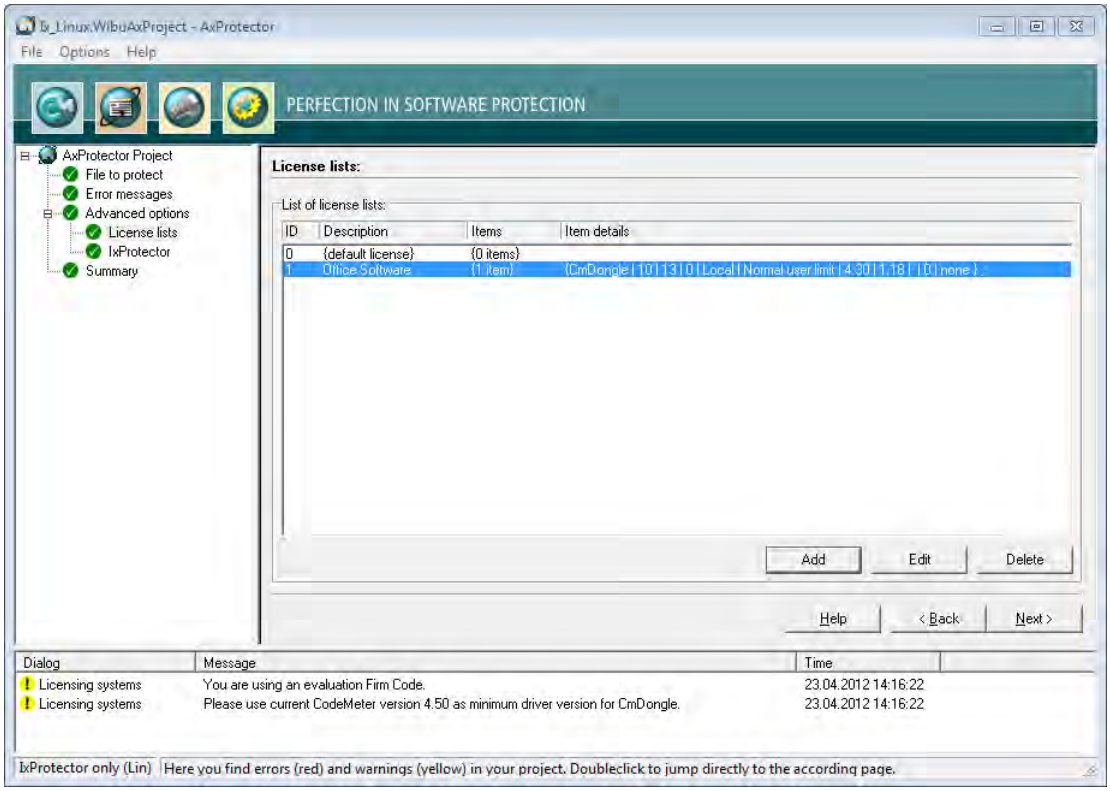


Figure 186: *IxProtector* Linux "Completed License Lists"

7.5.5.3.2 IxProtector

This menu item lets you define single modules or program functions of the protected application.

> Even when you use *IxProtector* without any further options, i.e. only the explicit encryption of functions, you nevertheless obtain more security for your application.
> In this case, *CodeMeter*® and *WibuKey* API calls, using the dynamic library (`*.dll`) are redirected to the corresponding statical libraries and appended to the application. Since the dll interface is left out, the security increases without making any changes to your application.

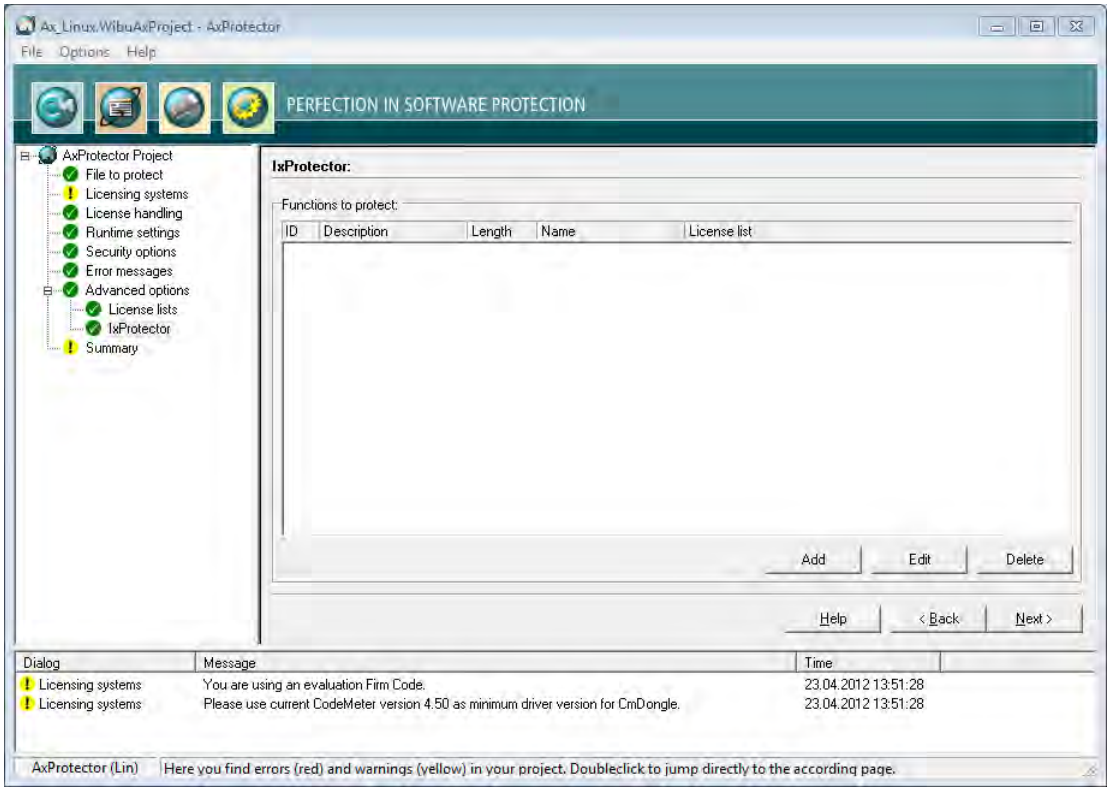


Figure 187: *IxProtector* Linux "Function List"

Element	Description
Functions to protect	Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows: **1.** Click the **"Add"** button in the group "IxProtector Options". **2.** Define the function by completing the fields in the "Function" group.

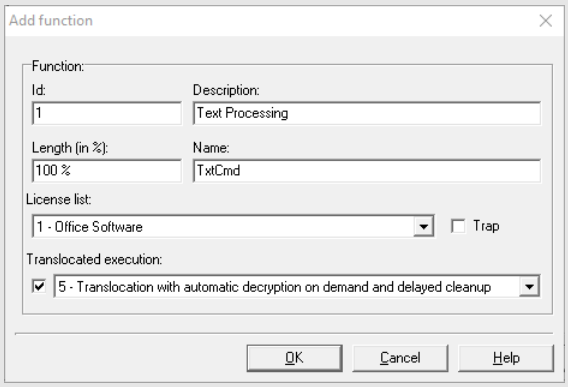


Figure 188: *AxProtector - IxProtector* only "Add Function"

Element	Description
Id	Uniquely identifies the function. This **Id** corresponds to the identification you use when calling the WUPI commands *WupiDecryptCode* 291 and *WupiEncryptCode* 291 .
Description	Enter a description of the function with text.
Length	The length of the array to be encrypted for the function is specified here. You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length. If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.
Name	Specify the name of the function to be encrypted. The function name must exactly match the name used in the export list of the linked map file. • Please note the correct spelling (case sensitive, underline, etc.). For detecting the exact function name you may use applications such as Dependency Walker.

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td>License List</td><td>Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.</td></tr><tr><td>Trap</td><td>Activates the trap function for the function.</td></tr><tr><td>Translocated execution</td><td>Uses the technique for shifting the execution of selected functions to other random locations in the process space without changing the data at the original position. There are the following selectable entries with different decryption and cleanup options.<table><tr><th>Option</th><th>Description</th></tr><tr><td>1</td><td>Translocation with automatic decrpytion on demand and cleanup.</td></tr><tr><td>2</td><td>Translocation, manual decrpytion and cleanup with WUPI-AP (Software Protection API).</td></tr><tr><td>5</td><td>Translocation with automatic decrpytion on demand and delayed cleanup. (Default)</td></tr></table>Command line option see here 286 .</td></tr></table> **3.** Click the **"OK"** button. The new functions are added to the function list.

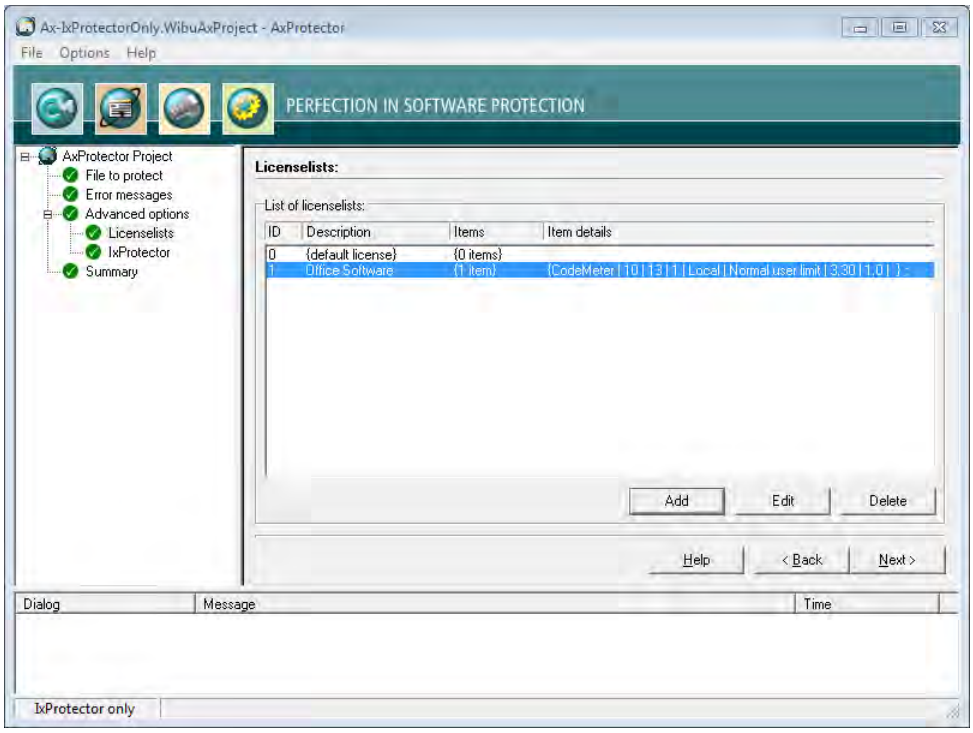


Figure 189: *AxProtector - IxProtector* only "Completed Function List"

7.5.5.4 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline 285 type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

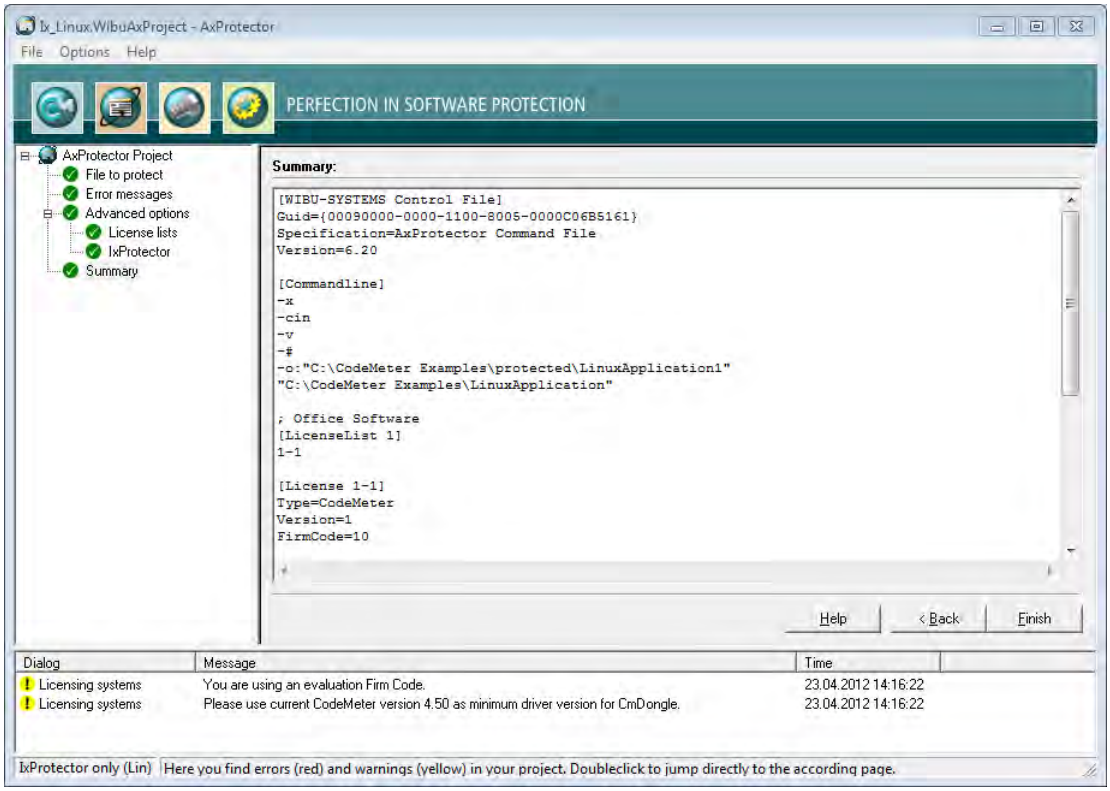


Figure 190: *AxProtector - lxProtector* only Linux "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

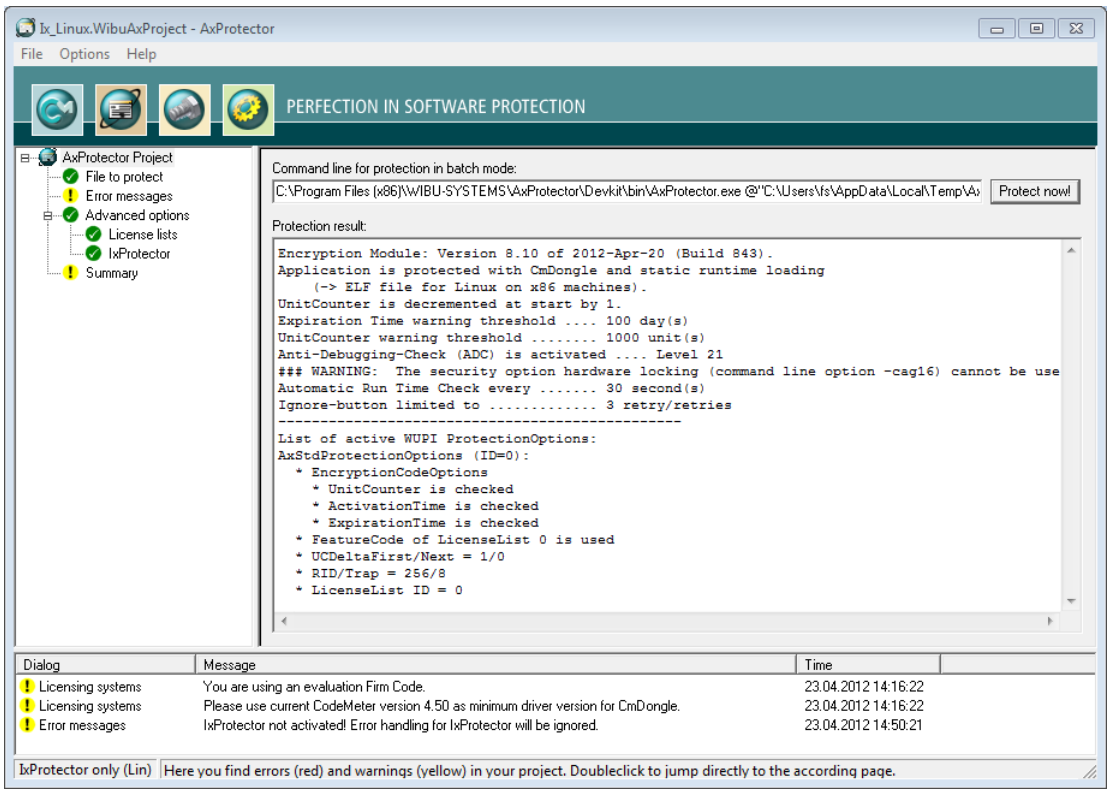


Figure 191: *AxProtector - lxProtector* only Linux "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode.

Element	Description
	You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.6 Other Tab

This tab offers you the selection of the following project type:


File encryption

7.6.1 File Encryption

AxProtector provides the automatic protection of files your protected application uses. This protection by encryption without altering the source code covers, for example:

- Flash applications consisting of a single `*.exe` or many *.swf files
- database applications, e.g. Visual Fox Pro applications consisting of a single*.exe and a single or multiple database files
- configuration data saved to separate files to be read by your software
- scripts saved to separate files to be processed by your software
- data, e.g. measuring data recorded or visualized in your application
- documents the user generates using your protected application.

The following menu items are available in the navigation windows:

- File to protect 247
- Licensing Systems 248
- Advanced Options 254
    - License Lists 254
    - File Encryption 259
- Summary 261

7.6.1.1 File to protect

To safely encrypt an executable file using *AxProtector*, first select the file you want to protect.

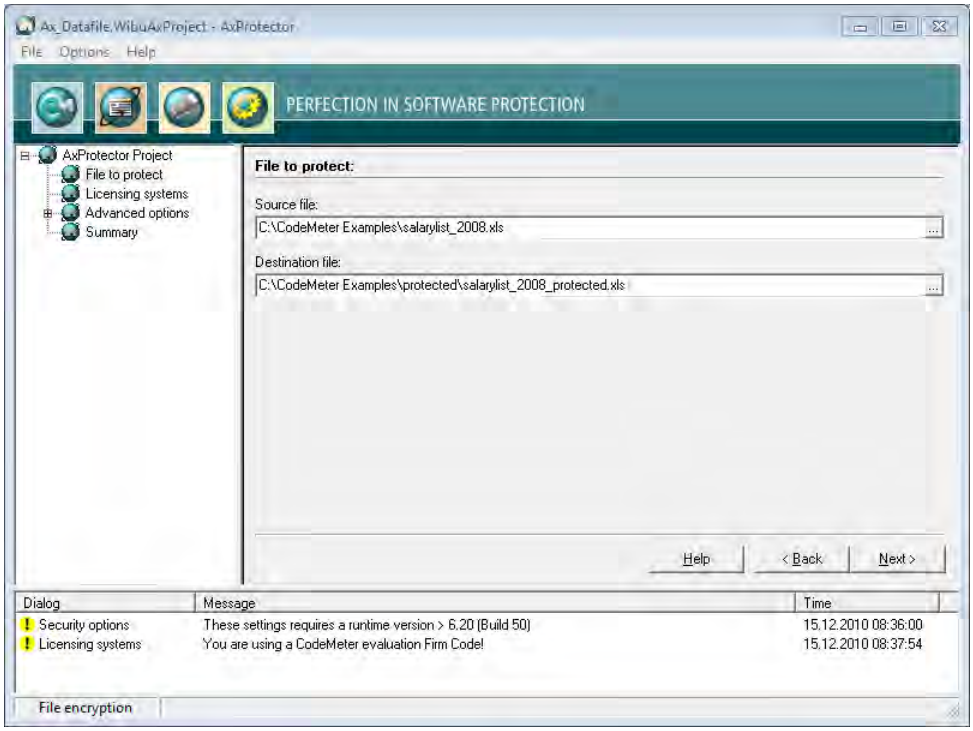


Figure 192: *AxProtector* - File Encryption "File to Protect"

File to protect

Element	Description
Source file	Click on the **"..."** button and select the file to protect using the system dialog **"Open"**. Alternatively, manually specify the path and name of the file in this field. As alternative to the **"..."** button, you may also directly drag & drop the source file from Windows Explorer into the source file field.
Destination File	After you selected the source file, *AxProtector* automatically creates a secondary folder `[..\protected\..]`. You may change this default by manually specifying the path and name of the destination file. Then the destination file corresponds to your protected application. Commandline option see here 279 .

7.6.1.2 Licensing Systems

After you select the file to be protected, the **"Licensing systems"** page displays in the input window. This is where you select and configure the license(s) to be applied. Depending on your requirements, you can select one or several licenses to be used for encrypting and later accessing your protected application.

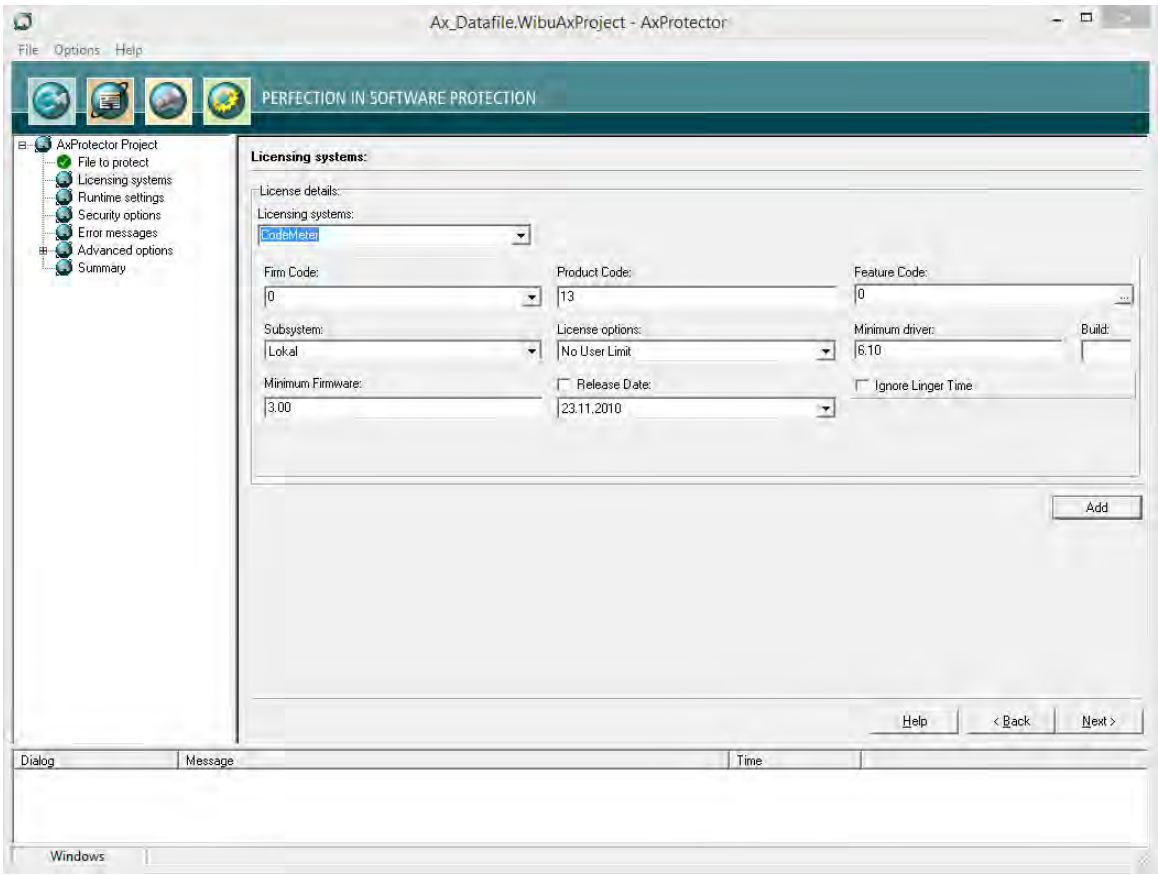
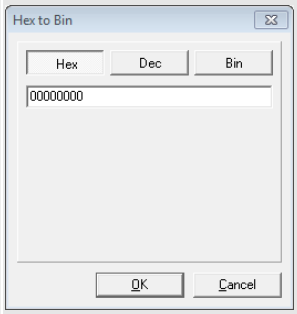


Figure 193: *AxProtector* - File Encryption "Licensing Systems"

Single License

For creating and editing the license details of a single license the following settings are available:

Element	Description
Licensing systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface.

Element	Description
	Entry / **Description** `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the ***Firm Code*** to be used for encrypting the software. As a registered licensor, you will be issued your own unique ***Firm Code***(s). The following default settings exist: ***Firm Code CodeMeter*** Software Development Kit (SDK) / Licensing system `6000010` Evaluation ***Universal Firm Code*** — *CodeMeter* `10` *CmDongle* Evaluation ***Firm Code*** — *CmDongle* `5010` *CmActLicense* Evaluation ***Firm Code*** — *CmActLicense* Commandline option see here 264 .
Product Code	Enter the ***Product Code*** which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the ***Feature Code*** which defines, for example, the encryption of different software versions. By default, a ***Feature Code*** of 0 is set. This deactivates the use of the Product Item Option ***Feature Map***.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 194: *AxProtector* - ***Feature Map*** Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description** Local — This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network — This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network — This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local — This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description** Normal user limit — Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share — Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode — Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit).

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td></td><td>This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.</td></tr><tr><td>Exclusive Mode</td><td>Here a protected application can be started only once on a PC.</td></tr><tr><td>No user limit</td><td>Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.</td></tr></table>
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>6.10 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr></table> Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>3.00 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr></table> Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the **"Add"** button to add additional license(s).

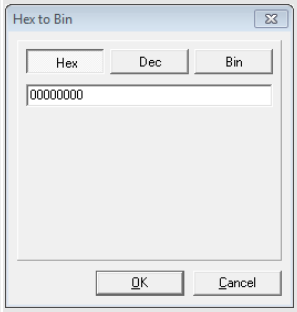
7.6.1.2.1 Licensing Systems - Add licenses

Several Licenses

If you want to use more than a single license to be used for encrypting and later accessing your protected application, you can do so. Please click the "**Add**" button to add additional license(s). The same settings as for configuring a single license are available.

Element	Description

Licensing systems	Select from the dropdown control the desired licensing system. Available are the following entries:
	`CodeMeter`
	`WibuKey`
	For setting *WibuKey* options, see the separate "WibuKey Developer Guide".
	If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems.
	In this way, you are able to ship updates and upgrades to existing customers who already have a WibuBox without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.

Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: *Firm Code CodeMeter* Software Development Kit (SDK) / Licensing system `6000010` Evaluation *Universal Firm Code* / *CodeMeter* `10` *CmDongle* Evaluation *Firm Code* / *CmDongle* `5010` *CmActLicense* Evaluation *Firm Code* / *CmActLicense* Commandline option see here 264 .
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 195: *AxProtector - Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). Element / Description Local / This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network / This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network / This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local / This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265).

Element	Description
	<table><tr><th>Element</th><th>Description</th></tr><tr><td>Normal user limit</td><td>Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network.</td></tr><tr><td>Station Share</td><td>Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.</td></tr><tr><td>WibuKey Compatibility Mode</td><td>Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'.</td></tr><tr><td>Exclusive Mode</td><td>Here a protected application can be started only once on a PC.</td></tr><tr><td>No user limit</td><td>Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.</td></tr></table>
Minimum driver	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>6.10 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.</td></tr></table> Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: <table><tr><th>*Firm Codes* (licensing system)</th><th>Version</th></tr><tr><td>`6000010; >= 6.000.000` (*Universal Firm Code*)</td><td>3.00 This supports the License Transfer feature.</td></tr><tr><td>`10, 100.000-4.999.999` (*CmDongle Firm Code*)</td><td>1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr><tr><td>`5010, 5.000.000-5.999.999` (*CmActLicense Firm Code*)</td><td>1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*.</td></tr></table> Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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).

Element	Description
	Commandline option see here 265 .

Moreover, the options WupiReadData and WupiWriteData are available.

Element	Description
	Reading and writing of data at runtime of an protected application is limited to license entries on the list which do not represent the default license.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

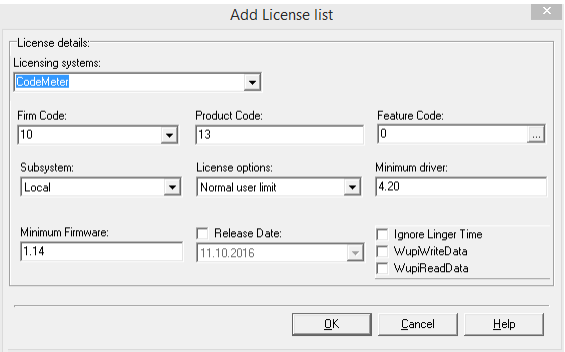


Figure 196: *AxProtector* - File Encryption "Add License list"

Click the "**OK**" button to add the new license(s) to the list. In the list display separate sort buttons at the list button allow you to sort the license entries to define a default license. In this view adding, editing or deleting licenses is supported.

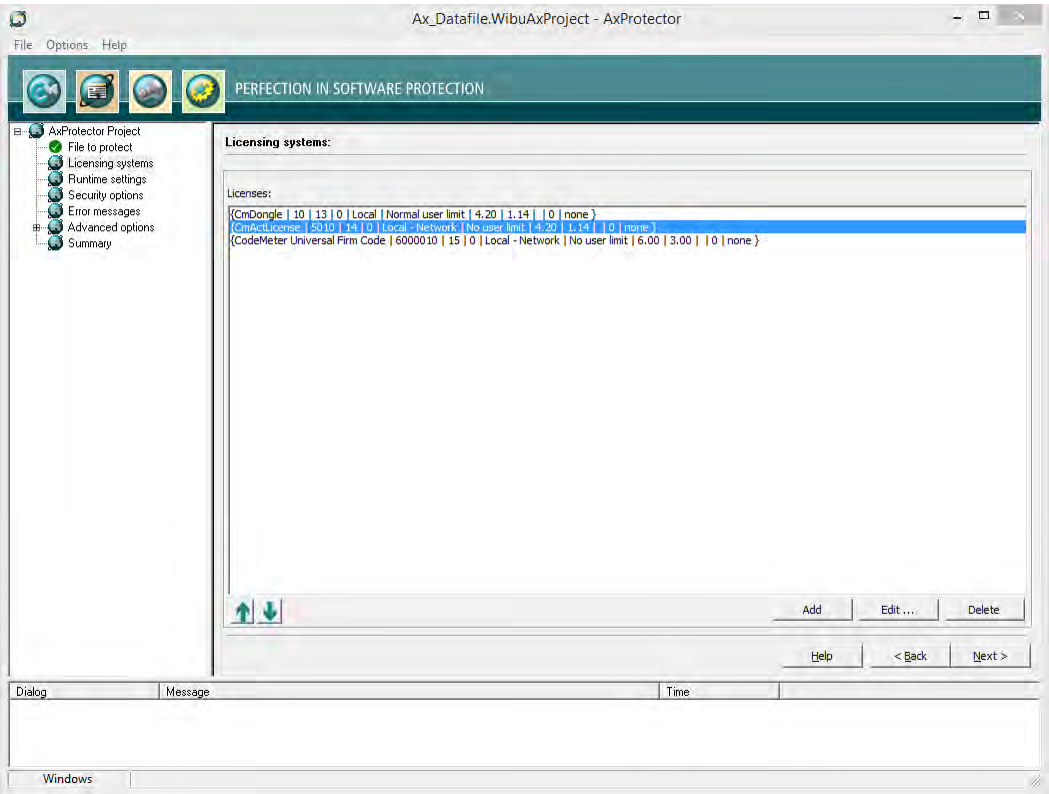


Figure 197: *AxProtector* - File Encryption "Licenses"

7.6.1.3 Advanced Options

This input window lets you set further encrypting options.

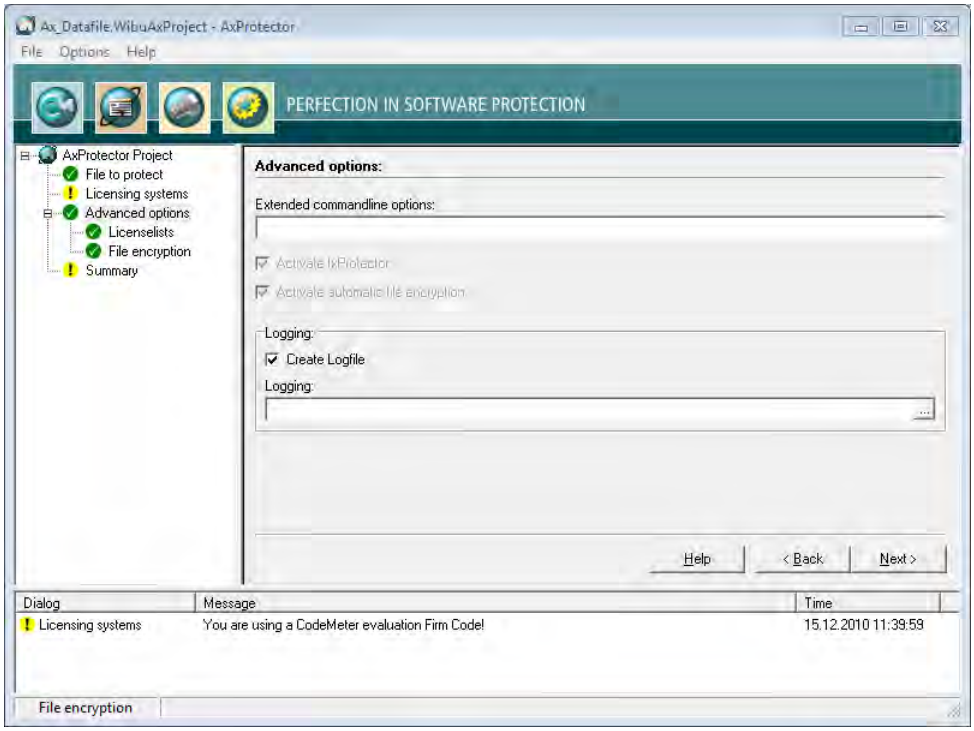


Figure 198: *AxProtector* - File Encryption "Advanced Options"

Element	Description
Extended Commandline Options	Here you are able to directly enter extended options or new feature functions using the *AxProtector* commandline. For more information please contact support at Wibu-Systems.
Create Logfile	Activate this checkbox to create file logging for the activities of *AxProtector*.
Logging	Specify the path and file name of this log file. If you specify the name of the file only, by default, this file is saved to the directory `%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin`.

7.6.1.3.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the *Software Protection-API (WUPI)* [289] .
License lists consist of a unique identifier (**ID**), a **Description**, and hold specifications on **Items** and **Item Details**.

This **ID** corresponds to the index number you require when addressing a license using most of the *WUPI commands* [290] .

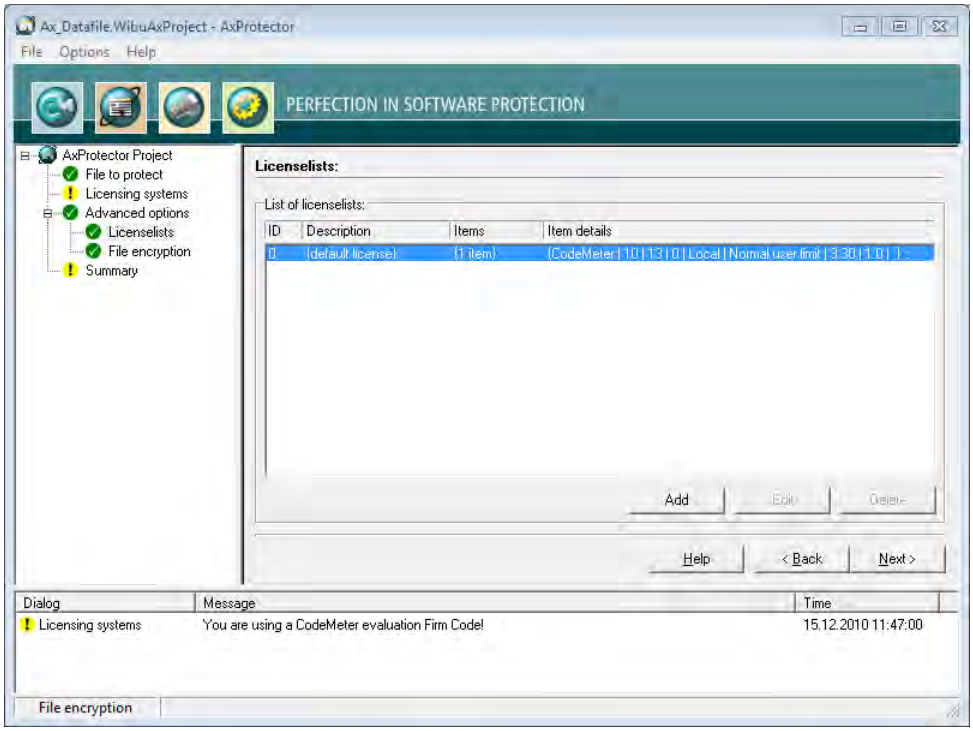
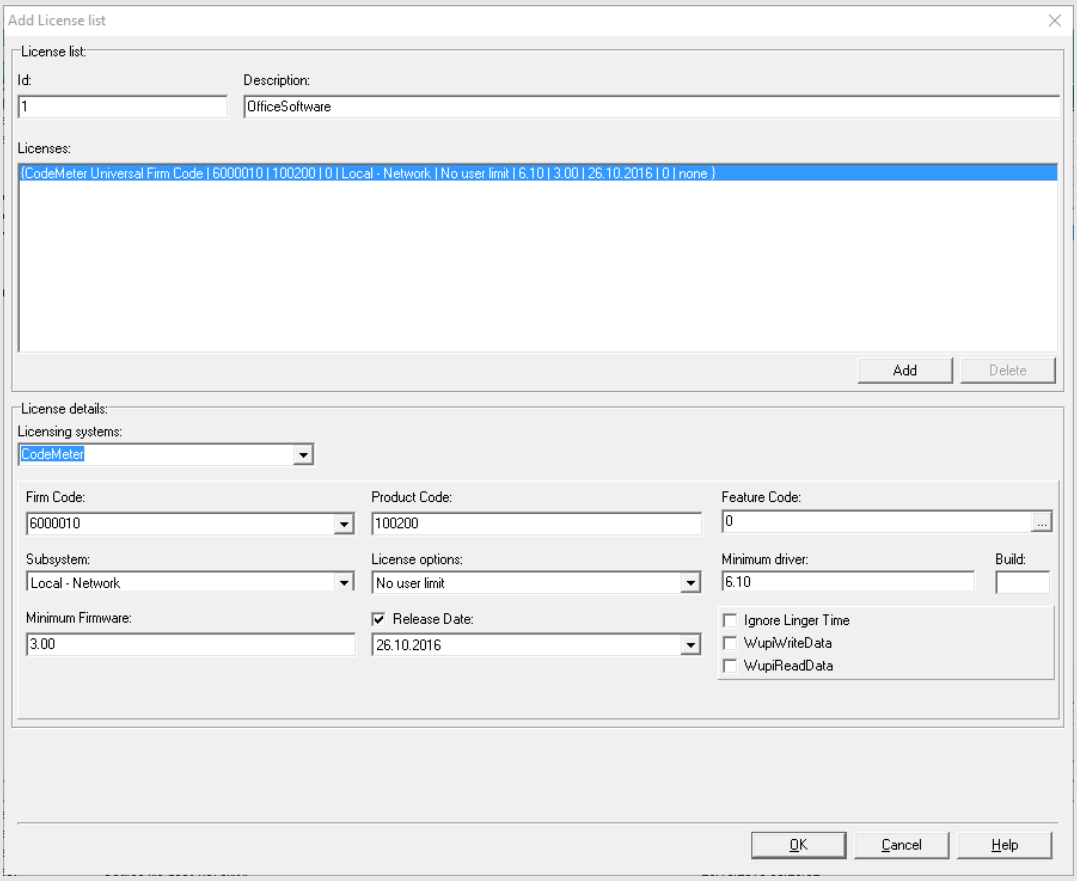


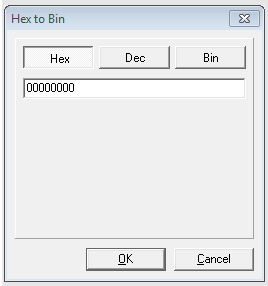
Figure 199: *AxProtector* - File Encryption "License Lists"

Using this menu items also allows you to create License Lists. Please proceed as follows:

1. Click the **"Add"** button.
2. Assign in the area **License List** an **Id** and complete the field **Description**.

Element	Description
Id	This ID uniquely identifies a license list and serves for referencing. By default, an **ID** of **0** is initially set by the selection of the licensing system. Following, you are able to add license list entries starting with **IDs** starting from **1**.

Element	Description
Description	Here you will describe a license list with text. **3.** Define the license by completing the fields in the *License item details* group.  Figure 200: *AxProtector* - File Encryption "Add License Lists"
Licensing Systems	Selecting the desired licensing system to be applied: **Entry** — **Description** `CodeMeter` — Applying the licensing system *CodeMeter*. `IP Protection` — Applying the licensing system *IP Protection*. Only the intellectual property is protected here. It is therefore not necessary to use a licensing system. However, a separate license from Wibu-Systems is required. Depending on the input file and the selected encryption options, *AxProtector* creates a key with which the application to be protected is encrypted. With unchanged parameters, this key remains constant and guarantees reproducible encryption and decryption. Commandline option see here 264 . • Please note that after a decision for exclusive protection (*IP Protection*) the selection of an additional licensing system is not supported and therefore not enabled in the user interface. `WibuKey` — Applying the licensing system *WibuKey*. For setting *WibuKey* options, see the separate "WibuKey Developer Guide". • If you are switching from *WibuKey* to *CodeMeter*, please activate both licensing systems. In this way, you are able to ship updates and upgrades to existing customers who already have a *WibuBox* without the need to replace the hardware. New end-users will be the ones to receive a *CmDongle* or a *CmActLicense* together with the protected application.
Firm Code	Specify the *Firm Code* to be used for encrypting the software. As a registered licensor, you will be issued your own unique *Firm Code*(s). The following default settings exist: **Firm Code CodeMeter Software Development Kit (SDK)** — **Licensing system** `6000010` Evaluation *Universal Firm Code* — CodeMeter Universal Firm Code `10` *CmDongle* Evaluation *Firm Code* — CmDongle `5010` *CmActLicense* Evaluation *Firm Code* — CmActLicense Commandline option see here 264 .

Element	Description
Product Code	Enter the *Product Code* which defines the encryption of a specific product. You can freely choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here 264 .
Feature Code	Enter the *Feature Code* which defines, for example, the encryption of different software versions. By default, a *Feature Code* of 0 is set. This deactivates the use of the Product Item Option *Feature Map*.Enter a 32-bit value to use the option. Using the **"..."** button you may enter the feature map value in hexadecimal, decimal or binary format.  Figure 201: *Feature Map* Input Commandline option see here 264 .
Subsystem	Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see here 265). **Element** / **Description**: Local: This setting determines if the protected application searches exclusively for licenses located on the same PC or allocated to the same VM. Network: This setting determines that the license of the protected applications is to be sought in the network, i.e. only PCs are accessed where *CodeMeter License Server* runs and is activated as network server. Local - Network: This setting determines that the license of the protected applications is to be sought first locally and subsequently on the network. Network - Local: This setting determines that the license of the protected applications is to be sought first on the network and subsequently locally.
License options	In this group you define how started instances of the protected applications perform together with the allocation of licenses (commandline options see here 265). **Element** / **Description**: Normal user limit: Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. Station Share: Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license. WibuKey Compatibility Mode: Here each started instance in the network allocates a license (normal user limit) but the local access is unlimited (no user limit). This allocation option exists only because of compatibility issues with *WibuKey*. Wibu-Systems recommends the setting 'normal user limit' and 'station share'. Exclusive Mode: Here a protected application can be started only once on a PC. No user limit: Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.
Minimum Driver Version	Enter the minimum driver version required for the installed *CodeMeter License Servers*. The following default settings exist: *Firm Codes* (licensing system) / Version: `6000010; >= 6.000.000` (*Universal Firm Code*): 6.10 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*): 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software.

Element	Description
	Firm Codes (licensing system) / Version `5010, 5.000.000-5.999.999` (*CmActLicense*) / 4.20 When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that *AxProtector* automatically handles sessions of the protected software, and each session is allocated one of the available licenses. Setting the driver version is also required when, for example, you wish to use new features for the encryption of an application. Older driver versions will not support these new features, and will trigger error messages when starting the protected software. Commandline option see here 264.
Build	Enter the Build number of the minimum driver version.
Release Date	Starting with Firmware version 1.18 *CodeMeter* supports the Product Item Option *Maintenance Period*
Minimum Firmware	Specify the minimum firmware version required. The following default settings exist: *Firm Codes* (licensing system) / Version `6000010; >= 6.000.000` (*Universal Firm Code*) / 3.00 This supports the License Transfer feature. `10, 100.000-4.999.999` (*CmDongle*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. `5010, 5.000.000-5.999.999` (*CmActLicense*) / 1.14 In order to use the Product Item Option *Maintenance Period* you require the firmware version 1.18.. After activating the checkbox you are prompted to accept that the "Mimimum Firmware" field changes to version 1.18 which is at least required to use the Product Item Option *Maintenance Period*. Commandline option see here 265.
Ignore Linger Time	Please note, that this option display only, if you checked in the menu navigation the entry "**Options | Display Advanced Licensing Options** 55". Activate this option to ignore a programmed *LingerTime*. 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). Commandline option see here 265.
WupiReadData	Activate this option to read data 292 from the *CmContainer* if this data has been previously stored at a defined location.
WupiWriteData	Activate this option to write data 293 into a *CmContainer* that has been prepared for storing additional data.

After you defined all desired settings in the area License Element Details, please proceed as follows:

4. Click on the **"Add"** button in the License List group. The summary of your specifications are displayed in the license item list.
5. Click the **"OK"** button. The new license data is added to the license list.

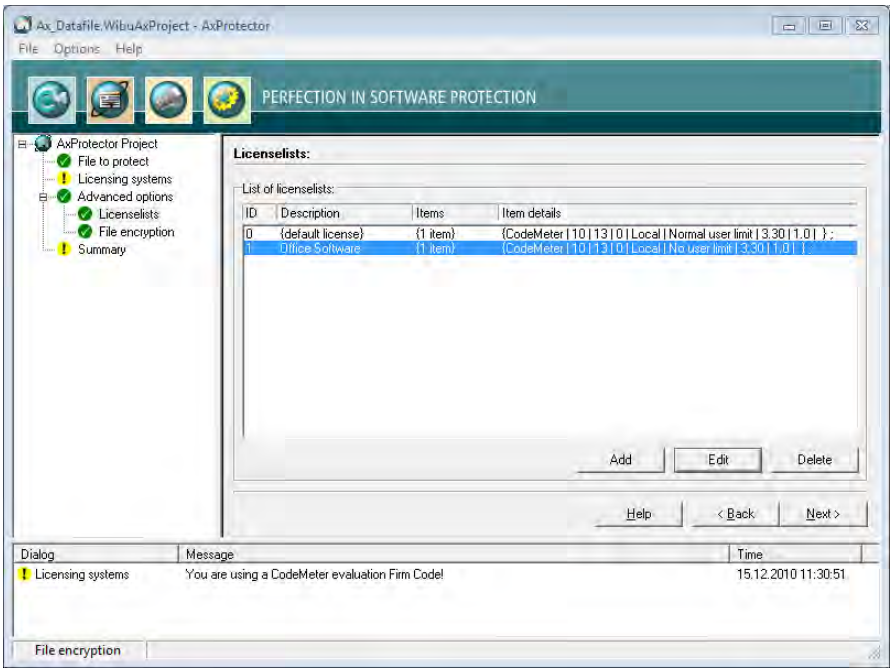


Figure 202: *AxProtector* - File Encryption "Completed License Lists"

7.6.1.3.2 File Encryption

This menu item lets you define the rules on how an application accesses the encrypted files. In addition, you have the option to define those rules in a list for different file types. You can add as many file types as possible. For a file only one file type is required.

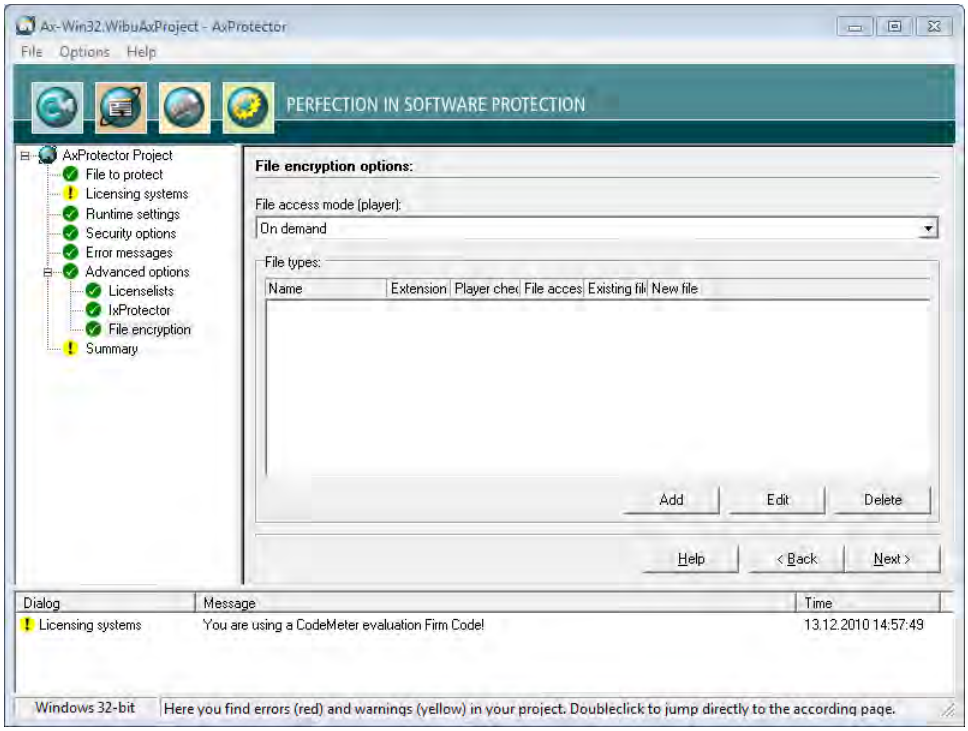


Figure 203: *AxProtector* - File Encryption "File Encryption"

Element	Description
Add File Type	**1.** Click on the **"Add"** button to add a new file type.

Figure 204: *AxProtector* - File Encryption "Add File Type"

2. Enter in the **"Name"** field a describing descriptive name for the file type. This name has no impact on the encryption.

3. Enter in the **"Extension"** field the file extension of the file type you create, e.g. txt for text files.

4. In the **"Player Check"** dropdown you define whether the license options of the accessing application (player) are checked when the decryption takes place.

License List	The player (accessing application) has to be encrypted using a license from this license list. For example, this allows you to define that a specific file type is accessed exclusively by the application you encrypted.
No player check	No check of the accessing application is performed.

5. In the **"File Access Mode"** dropdown define how the player is prepared for the access of protected files. This mode allows you to configure the memory required and the runtime behavior.

The selection of a suitable mode depends on the type of the player and the size of the file. For example, when working with video files you should select "Huge file mode (read only)". In the case of smaller files (configuration files) you may access several times, the mode "At once" is preferable.
Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.

Blockwise	The player reserves RAM space for the complete file to be read; but reads only the required part – strictly speaking all 4 Kbyte blocks are holding this part – and decrypts these blocks. For further accesses to the protected file, more required blocks are loaded (on demand) and decrypted. When the required part is located in blocks already loaded, the decrypted image in the memory is used. In this way, step-by-step the player builds up a complete memory image of the required file. This mode requires a lot of memory (the same size as the file to be loaded). However caching the decrypted data provides for good performance at runtime when accessing

Element	Description		
			already decrypted blocks. This mode is available for read and write access.
		At once	The player reserves RAM space for the complete file to be read; completely reads it, and completely decrypts it. Further accesses to the protected files, use the decrypted memory image. This mode requires a lot of memory (the same size as the file to be loaded). However, caching the decrypted data provides a good performance at runtime. Compared to the "on demand" mode, this mode requires more time for first access (the file is completely loaded and decrypted). The performance of each additional access is increased because the file resides completely in memory, in a decrypted form. This mode is available for read and write access.
		Huge file mode	The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory. This mode requires no additional memory. Multiple accesses to the same data means that the data has to be read and decrypted each time. This mode is available for read access only.
	6. In the **"Write Options"** define the settings on how changes are saved. **Existing File** In this group you define the settings on how changes to an existing file are saved.		
		Original	Changes are allowed. Where the file was encrypted, it is re-encrypted. Unencrypted files are saved with no decryption.
		No writing	Write actions are not allowed. Just read-only access is allowed.
		License list	Changes are only encrypted using the license options defined in the selected license list.
	New File In this group you will define the settings on how new files are saved.		
		Plain	New files are only saved unencrypted.
		No writing	New files cannot be saved. A new file is saved, however no data is saved to this file.
		License List	New files are only encrypted using the license options defined in the selected license list.

7.6.1.4 Summary

This input window shows you a summary of all the settings you defined for the automatic protection of your application, and allows you to start the encryption process.

> For subsequent use, the contents of this page can be copied to a `*.wbc` file (WIBU Configuration file). Copy the content into a text file, and change the file extension to `*.wbc`.
> Alternatively, you may also use this file to protect your application using the *AxProtector* commandline tool. In the commandline[285] type `AxProtector.exe @*.wbc`.
> Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding `*.wbc` file.

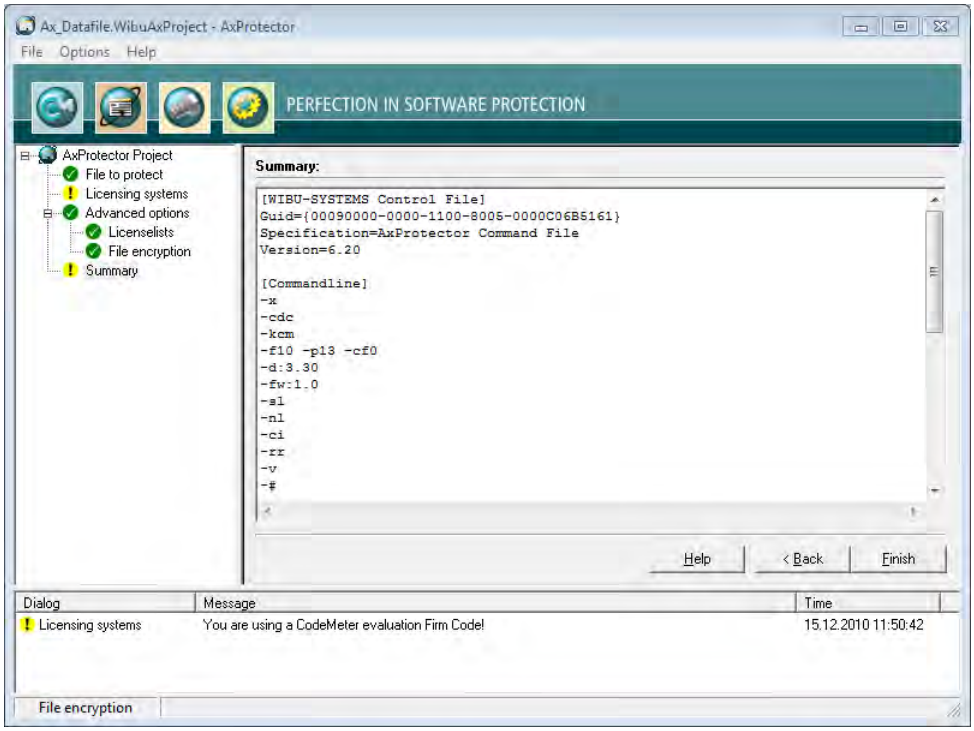


Figure 205: *AxProtector* - File Encryption "Summary"

Element	Description
Finish	Starts the encryption using *AxProtector* applying the settings you previously defined.
Back	Allows returning to change previous settings.

The result of the encryption with all relevant settings is displayed in a separate window.

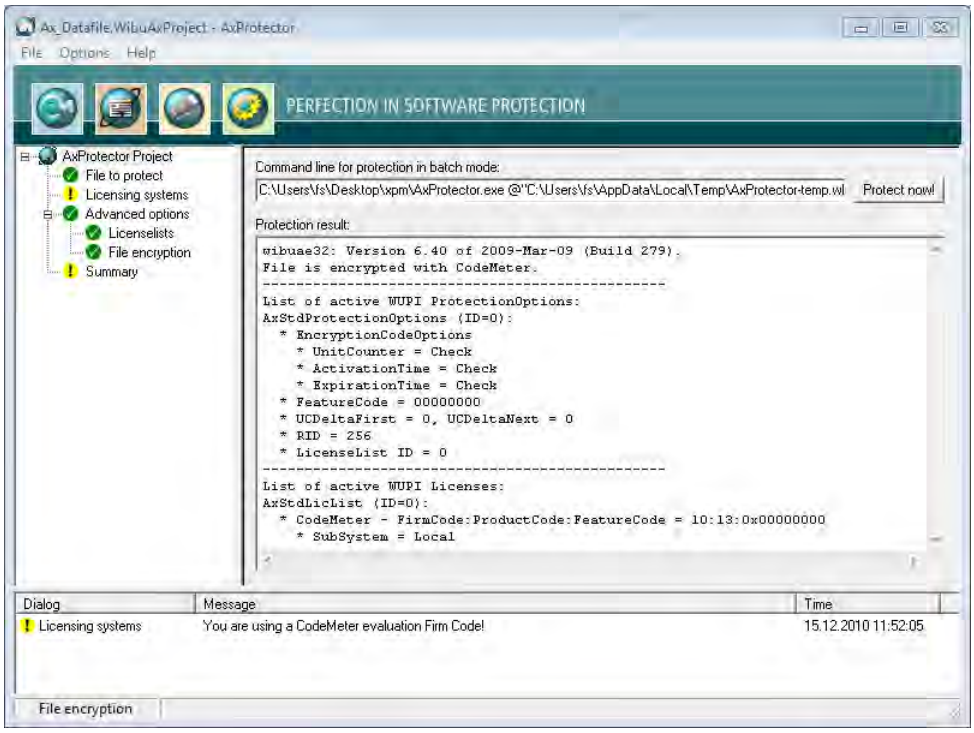


Figure 206: *AxProtector* - File Encryption "Encryption Result"

Element	Description
Protect Now	When you need to repeat the encryption operation, click the **"Protect now"** button. Then the *AxProtector* commandline is executed in batch mode. You are also able to copy the *AxProtector* commandline for the batch mode to the clipboard and insert it in the commandline input. Subsequently, you can edit it and apply any desired changes.

7.7 Commandline Options for AxProtector

As an alternative to the graphical user interface of *AxProtector*, you can also set the options for automatic encryption using the *AxProtector* commandline.

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

Version	Project types
`AxProtector.exe`	32 64
`AxProtectorNet.exe`	.NET
`AxProtectorMacX`	Mac
`AxProtector.jar`	Java
`AxProtectorLin`	in a 32-bit and 64-bit version

Which options are valid for which*AxProtector* versions is indicated by the symbols in a separate table row.

Commandline Syntax

The commandline call follows the syntax below:

```
AxProtector Version call -<Options> <Path and name of the application to be protected>
```

7.7.1 Basic Options

Option	-X
valid for	32 64 Mac

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	32 64 Mac

Specifies the encryption algorithm (*CodeMeter* only).

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

7.7.2 Options for the Licensing System

Option	-K([CA][CM][WK])
valid for	32 64 Mac .NET Java

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	-KIP[hex value]
valid for	32 64 .NET Mac
	specifies a fixed key (*IP Protection* 287) which is stored as a fixed, secret 16 byte key in the standard license list (`*.wbc` file or XML file in the case of .NET) and used for cryptographic operations. By default, *AxProtector* generates this key depending on the input file and the selected encryption options. With unchanged parameters, the key then remains constant and guarantees reproducible encryption. In the `*.WBC` file or XML file in the case of .NET the fix key can also be defined for self-selected license lists. Optionally, the key can be specified in the form of a hex value. e.g. `-kip` or `-kip0x112233445566778811223344556677 88`

Option	-Fx
valid for	32 64 Mac .NET Java
	Specifies the *Firm Code*(x) to be used. Expects the input of an unsigned integer value `<n>`.

Option	-Px
valid for	32 64 Mac .NET Java
	Specifies the *Product Code* (x) to be used. Expects the input of an unsigned integer value `<n>`.

Option	-CFx
valid for	32 64 Mac .NET Java
	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	32 64 Mac .NET Java
	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. Requires *CodeMeter* Version 4.30 and Firmware-Version 1.18.

Option	-D:v
valid for	32 64 Mac .NET Java
	Specifies the minimum driver version (`v`). Input of `v` using (`x.y`). Default setting: *CodeMeter* 4.20. Default setting: *WibuKey* 5.20.

Option	-D:v

For .NET the minimum driver version must be Version 4.0 or higher.
This version specification also holds when simultaneously using the *Softwaer Protection API* [289] (WUPI) (Oprion `-ci`) and the *CodeMeter Core API* [297] (`WibuCmNET`).

Option	-FW:v

valid for 32 64 Mac .NET Java

Defines the minimum firmware version(`v`).
Input of `v` using (`x.y`).
Default setting: *CodeMeter* 1.0
Is not used with *WibuKey*

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

valid for 32 64 Mac .NET Java

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 32 64 Mac .NET Java

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).

Since this is the default license allocation with *WibuKey*, this option ensures compatibility when both licensing systems are used at the same time.

(`A`: uses auto cancel (*WibuKey* only).

Parameter	-NL[A]

normal user limit: here each started instance allocates a license regardless whether the *CmContainer* is found locally or on the network.
(`A`: uses auto cancel (*WibuKey* only).

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.

Parameter	-NS

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' 270 for the runtime check has now to be set explicitly.

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

(`A`: uses auto cancel (*WibuKey* only).

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

valid for 32 64 Mac

Selects a private key to calculate the signature.

Parameter	Fx

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

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 32 64 Mac

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

7.7.3 Options for Encrypting and Decrypting

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

Encrypts the executable file using automatic encryption.

Parameter	-CAA <1>

valid for 32 64

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)

Parameter	`-CAA <l>`

Option	Description
	`CAA6` applies APS 2 and 3 `CAA13` applies APS 1, 4 and 8

valid for .NET

Activates the security options.
`<l>` covers the options `[0, 1]`

Option	Description
`0`	No resource encryption is performed.
`1`	Resource encryption applies.
	Allowed combinations: `CAA0` no resource encryption. `CAA1` applies resource encryption.

valid for Java

Activates the security options.
`<l>` covers exactly the options `[1, 2]`

> A combination with "or" is not supported. You have to specify each option.

Option	Description
`1`	Encryption of selected values from the constants pool
`2`	Encryption (obfuscating) of generated method calls This option must be combined with option `-ci` 274 for method encryption.

Parameter	`-CACt<,u>`

valid for 32 64 Mac

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 32 64 Mac

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 32 64 Mac

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<l>`

valid for 32 64

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

Parameter	-CAG<1>

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	-CAG<1>
valid for	Mac

Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC).
<l> covers the options [0,469]

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).
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).
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.
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	-CAG<1>

valid for .NET

Activates Anti-Debugging mechanisms (Anti-Debug Checks, ADC).
<1> covers the options [0, 257]

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

Option	Description
0	no debugger check. Default setting if -CAG is not specified.
1	Checks with a simple Debugger check (ADC1).
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.
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	-CAG<1>

valid for Java

Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC).
<1> covers the options [0, 271]

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

Option	Description
0	no Anti-Debug mechanism is applied. Default setting of -cag is not specified.
1	Checks for the detection of the JVMPI (Java Virtual Machine Profiler Interface). JVMPI can be used to manipulate the Java virtual machine sending messages to the native code. In particular, the event JVMPI_EVENT_CLASS_LOAD_HOOK may be used to intercept the original byte code of the actual class. Activating this option prevents this interception.
2	Checks for manipulation of callback functions, i.e. the access to objects of other classes is checked.
4	Checks the Java Virtual Machine for Java versions 6 up to Java 8 (1.6. - 1.8) using a signature check. Beginning with *AxProtector*-Version 10.10 the signature check is no longer supported. If this option is set in *AxProtector* Version equal to or higher than 10.10, an error is returned and a message displays that the feature works only for Java Version 1.6. - 1.8 and that the application will not run for Java Version 9 and higher. In order to transform the error into a warning message, please insert -ignore:jvmverificationerror.
8	Checks with a simple Debugger check (ADC1).
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.
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	32 64 Mac
	Limits the automatic encryption to specified areas.

Parameter	-CAM
valid for	32 64
	Adds the menu items 'Control' and 'About' to the application's system menu.

Parameter	-CAR<t>,<m>
valid for	32 64 Mac .NET Java
	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	32 64
	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	32 64 Mac .NET Java
	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). i <t> has to be equal or greater than <u>.

Parameter	-CAV<n>
valid for	32 64 Mac
	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 55 " 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
```

i 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.

Parameter	-CAV<n>
valid for	Java
	Adds a code integrity check to the automatically encrypted application. i The options can be be combined by "or". The default setting for <level> is 0.

Level n	Description
0	no integrity check <default>.

Parameter	-CAV<n>

Level n	Description
1	Checks Wibu Runtime Classes at runtime for integrity. This also includes the MessageHandler. The MessageHandler specified at encryption cannot be replaced after the encryption has been performed.
2	Checks all classes within the application, e.g. the jar file, at at encryption. This setting may be deactivated for single classes using *IxProtector* (WUPI), e.g. Integrity=false.

Parameter	-CAV<n>

valid for .NET

Adds a code integrity check to the automatically encrypted application or not.

Level n	Description
1	adds a code integrity check to the automatically encrypted application.

Parameter	-CAV2

valid for 32 64

Deactivates the code integrity check for an automatically encrypted application or for several executable files / libraries.

Parameter	-CAZ

valid for 32 64 Mac .NET Java

Saves the time when the encryption was performed within the protected application (*CmContainer* System Time. Then the application runs only when the PC time is older than this encryption time.

Requires at least *CodeMeter* version 4.10.

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 32 64

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].
<s> contains the subsystem [9]

1: x86 / Intel 32 Bit
2: ArmV4i
9: Windows CE System

Parameter	-CCB

valid for 32 64

Disables usage of proprietary (B)ase relocation representation.

Parameter	-CCE

valid for 32 64

Specifies that the PE is not enlarged.

Parameter	-CCF<number>

valid for 32 64

Specifies extended protection / special handling using a numeric value.

<number> holds numerical value

1: AutoCAD 2011 - ARX files (plug-ins)

Parameter	-CCH

valid for 32 64 Mac

Prevents all global hooking in a protected application.

Parameter	-CCI

valid for 32 64

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

Parameter	-CCI
	the option "-ccm" which is no longer required.

Parameter	-CCK
valid for	32 64
	does not explicitly unload DLL, Windows XP only.

Parameter	-CCO
valid for	32 64 Mac
	Activates special handling for ActiveX / OCX images.

Parameter	-CCQ
valid for	32 64 Mac
	Clears license use for the protected applications not when `WM_QUIT` is called but with the call of `ExitProcess()`.

Parameter	-CCR
valid for	32 64
	Deactivates the renaming of sections.

Parameter	-CCS
valid for	32 64 .NET Mac
	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	32 64
	allows specifying the AxEngine by using `<selector>` as selected architecture. The entries cover: `linux-armhf`, `embedded-micro`, `android-so`. e.g. `-ctt:linux-armhf`: specifies the AxEngine for the linux-armhf architecture (e.g. Raspberry).

Parameter	-CCX
valid for	32 64
	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	

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	32 64
	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	32 64
	Applies the file name extension from the `*.wbc` file.

Parameter	`-CDH`
valid for	32 64
	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	32 64
	Specifies the used licensing system: `CA` uses *CmActLicense* `CM` uses *CmDongle* `WK` uses *WibuKey*.

Parameter	`-CDK([CA]|[CM]|[WK])F`
valid for	32 64
	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	32 64
	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	32 64 Linux Mac

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

Simulation of encryption during development

Up to *AxProtector* version 10.40, *WUPI* functionalities could only be used and tested with *IxProtector* if the application to be protected was actually encrypted. This required the integration of the *WUPI* engine into the application by the developer and the use of a dummy library (`WupiDummy.lib`, `*.so`), which could partially provide functionalities. For example, WupiCheckLicense only returned `True`.

> Starting with *AxProtector* version 10.40 this behavior has changed by default.
> Now *WUPI* calls are also possible in the unencrypted state of the application.
> The calls are not supported: WupiCheckDebugger, WupiRead... and WriteData....

Encryption is now simulated and allows developers to test *WUPI* implementation functionality using valid license information in test environments, e.g. by calling WupiQeryInfo. For this purpose, the *WUPI* engine now imitates a normal behavior of the encrypted state of an application.

In order to tell the *WUPI* Engine which information belongs to which application, the corresponding WBC file(s) are searched. To do this, the WBC files must be located in the same directory at runtime, i.e. the execution path for an application or storage location in the case of plugins.

The WBC files are searched for the original file name of the application, i.e. not for the file name specified with the '`-o:` 279' option. If the original file name of the application matches in several WBC files, the WBC file in which this name was found for the first time is used. This also applies to a deactivated simulation (`Active` then has a value of `0`).

The search parameters function independently of upper and lower case and, in the case of several files, according to alphabetical order.

In addition, the WBC file can be told whether the simulation should be activated or deactivated. This is done via the new Topic `[Simulation]`:

```
[Simulation]
Active = 0 or 1
```

By default, simulation is enabled and `Active` has a value of `1`

> All users who want to continue working with *WUPI* must set the Topic `[Simulation]` and set `Active` to a value of `0`.

Parameter	`-CIH`
valid for	32 64 Mac Linux
	Defines that Wupi`XXX` functions functions are NOT dynamically hooked.

Parameter	`-CIN`
valid for	32 64 Mac Linux
	Defines that no error messages are displayed when an error occurs.

Parameter	`-CIP`
valid for	32 64
	Defines that no pointer based IxProtector table is searched.

Option	-CEI
valid for	32 64

activates an alternative initialization mechanism for DLLs.
The encrypted dll is loaded, but the *AxEngine* is not initialized. Only when an exported function is called for the first time does the *AxEngine* initialize and thus the first license check take place. Please note this for time-critical applications.

Option	-CI
valid for	Java .NET

> For .NET the minimum driver version must be Version 4.0 or higher.
> This version specification also holds when simultaneously using the *Softwaer Protection API* 289 (WUPI) (Oprion `-ci`) and the *CodeMeter Core API* 297 (`WibuCmNET`).

Activates the encryption of explicitly defined source code fragments (classes / methods) within the executable file to be used with *IxProtector*.
Which classes / methods are encrypted is set by using various annotations (for details see Java-specific options 280).

Option	-CK<n>
valid for	.NET

Buffers the RID key of the application for `<n>` seconds into the cache memory.
`<n>` may have values between 0 and 255.

Option	-CO(n)
valid for	.NET

Specifies which elements are obfuscated (starting with *AxProtector* Version 8.40).
`<n>` covers the options [`0 .. 15`]
The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information.

> When applying more than one obfuscation option, you can combine options by "or".
> The default setting for `<n>` is `0`.

Option	Description
0	no elements are obfuscated.
1	private elements are obfuscated.
2	internal elements are obfuscated.
4	protected elements are obfuscated.
8	public elements are obfuscated.

Also starting with *AxProtector* Version 8.40 the option exist to control obfuscation by explicitly exclude elements by setting the obfuscation attribute of the Namespace `System.Reflection`.
This attribute can be assigned to classes, methods, fields, and properties.
The following `Named Parameter` are valid:

Parameter	Values	Description
`Exclude`	`true` / `false` Default value is `true`.	True excludes the element from obfuscation
`ApplyToMembers`	`true` / `false` Default value is `true`.	The setting is valid for all Member, if the attribute is assigned to a class.
`StripAfterObfuscation`	`true` / `false` Default value is `true`.	The obfuscation attribute is removed on obfuscation.
`Feature`		is ignored

The obfuscation attributes are always interpreted.

Option	-CO(l)
valid for	Java

Activates obfuscation of class. method, field, package or local variable names.
`<l>` covers the options [`0 .. 1536`]
The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information.

> When applying more than one obfuscation option, you can combine options by "or".
> The default setting for `<l>` is `0`.

Option	-CO(l)

Option	Description
Level 0	No obfuscation (default).
Level 1	Class names.
Level 2	Method names.
Level 4	Local variable names.
Level 8	Field names.
Level 16	Package names.
Level 32	Private elements.
Level 64	Inner elements.
Level 128	Protected elements.
Level 256	Public elements.
Level 512	Ignore detection of reflection.
Level 1024	Print name mapping during encryption.

Option	-CPA

valid for .NET

Deactivates encryption of property accessors.

Option	-CMD<n>

valid for .NET

activates reencrypting of methods after discarding.
`<n>` allows specification of seconds.

Option	-CML<n>

valid for .NET

Excludes methods from encrypting which are smaller than `<n>` bytes.
`<n>` has the default value is 10. On specifying a value of `0` the feature is deactivated.

Option	-EC

valid for .NET

Encrypts class constructors in .NET (MSIL) code.

Option	-CP<l>

gilt für 32 64

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

Defines additional checks while encryption and decryption operations are performed.

Parameter	-EA

valid for 32 64 Mac .NET Java

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 32 64 Mac .NET Java

Activates an *Expiration Time* check.

C	Checks if the Product Item Option *Expiration Time* exists.

Parameter	-EE
	I — Ignores the Product Item Option *Expiration Time* (*CodeMeter* only).
	R — Requires the Product Item Option *Expiration Time*.

Parameter	-EF
valid for	32 64 Mac .NET Java
	Activates the decrement of the *Firm Access Counter* (*CodeMeter* only).

Parameter	-EM
valid for	32 64 Mac .NET Java
	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	32 64 Mac .NET Java
	Enforces an *Certified Time* after the *CmContainer* is activated.
	This option requires an activated *Expiration Time*.

Parameter	-EU
valid for	32 64 Mac .NET Java
	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).
	e.g. -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	32 64 Mac .NET Java
	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	32 64 Mac .NET Java
	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	32 64 Mac .NET Java
	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 (32 64 only).
	"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	32 64 Mac .NET Java
	Sets the minimum Firmware version in the encryption.

Option	-W[C[t]]E[t]][P][U[c]]
valid for	32 64 Mac .NET Java

Specifies the threshold of issued warnings.

Parameter	-WC[t]
valid for	32 64 Mac .NET Java

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

Parameter	-WE[t]
valid for	32 64 Mac .NET Java

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

Parameter	-WP
valid for	32 64 Mac .NET Java

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

Parameter	-WU[c]
valid for	32 64 Mac .NET Java

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

7.7.4 Runtime Options

Option	-I
valid for	32 64 .NET

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	32 64 .NET Mac

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	32 64 Mac .NET Java

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	32 64 Mac .NET Java

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.

Parameter	-MC
valid for	32 64 Mac .NET Java

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

Parameter	-ME
valid for	32 64 Mac .NET Java

Holds the text, if an error has occurred.

Parameter	-MI
valid for	32 64 Mac .NET Java

Holds the text displayed, if the required driver of the licensing systems is not installed.

Parameter	-MI
	The return value to the User Message DLL corresponds to the already existing WUPI error code `wibu::UpiErrorLicenseModuleNotLoaded`.

Parameter	-MS
valid for	32 64 Mac .NET Java
	Holds the text displayed while the protected application is started.

Parameter	-MT
valid for	32 64 Mac .NET Java
	Holds the text displayed, if the date of the *Expiration Time* has been reached

Parameter	-MU
valid for	32 64 Mac .NET Java
	Holds the text displayed, if the *Unit Counter* has reached a value of 0.

Parameter	-MWC
valid for	32 64 Mac .NET Java
	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	32 64 Mac .NET Java
	Holds the text displayed on application start, if an existing *Usage Period* has not yet been activated.

Parameter	-MWT
valid for	32 64 Mac .NET Java
	Holds the text displayed, if the end of the *Expiration Time* or the *Usage Period* has been reached.

Parameter	-MWU
valid for	32 64 Mac .NET Java
	Holds the text displayed, if the *Unit Counter* is about to reach a value of 0.

Parameter	-ML
valid for	.NET
	Holds the text displayed, if the license have not been found.

Parameter	-MR
valid for	.NET
	Holds the text displayed, if the runtime check failed.

Parameter	-MV
valid for	.NET
	Holds the text displayed, if the runtime version is too old.

Option	-U[:FileName]
valid for	32 64 Mac
	Calls the file `UserMsgXX.dll` where XX stands for a country placeholder, e.g. `De`, `Us`, etc.. `[:FileName]` When specifying the `FileName`, the user-defined Message DLL holds the name `FileNameXX.dll` where `XX` stands for an optional country placeholder `Us`, `Sa`, `Cn`, `Dk`, `Nl`, `Fr`, `De`, `Gr`, `It`, `Hu`, `Jp`, `Ko`, `Br`, `Es`, `Se`, `Tw` (Project type 32 64 only.
valid for	Java
	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	-UI
valid for	.NET

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

Option	-UM[:FileName]
valid for	.NET
	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	-UN
valid for	32 64 Mac
	switches off standard error messages when standard UserMessage is used.. This option has no effect on self-used UserMessage libraries.

Option	-UDT[C|A|:"<text>"]
valid for	32 64 .NET Mac Java
	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	-ANF
valid for	.NET
	Specifies the text displayed when an assembly is not found. Default setting: `The assembly "#requiredassembly#" could not be found.`

Option	-PROBING:<Name> (oboslete with Version 9.20)
valid for	.NET
	Specifies the path information where assemblies can be found. The input format is either separated by `';'` or specification of the name of an `app.config` file.

Option	-SNK[F,N]:<Name>
valid for	.NET
	Specifies the Strong Name key for the assembly, and use it for signing the assembly. `f` Signs the assembly with the key pair defined in the file `<Name>`. `n` Signs the assembly with the key pair defined in the key container `<Name>`. In the case of .NET on specifying a file name, please note that the relative path information always bases on the directory of the input assembly.

Option	-TRAP[1[:n]]
valid for	.NET Java
	Inserts hacker traps into the protected assembly. Adds approx. `n%` methods to the encrypted Assembly which will lock the *CmContainer* on the next decryption process. The default setting for `n` has a value of 10. For Java automatic traps are generated only for methods encrypted using IxProtector License List. Only those methods will be transformed to new classes.

Option	-PRIO[0..31]<S>
valid for	32 64
	sets the process priority during startup of the image. `0` 0 sets no priority change, 8 is normal priority `S` specifies that the priority is not restored after running startup.

Option	-O[:FileName]
valid for	32 64 Mac Java .NET

Option	-O[:FileName]
	Specifies the path and the name of the encrypted destination file.

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

7.7.5 Java-specific Settings

Method Encryption (*IxProtector* Java starting with *AxProtector* Version 9.0)

Starting with Version 9.0 the option to encrypt single methods using *IxProtector* is introduced. Therefore the commandline option -ci 274 is featured.

Please note that once you set the option `-ci`:

- the options `-jn:t` 284 and `-jh:n` 283 are activated by default and must not be specified.
- Only JVMTI (Java Virtual Machine Tool Interface) is supported but no longer JVMPI (Java Virtual Machine Profiler Interface).
- encryption of single classes instead of `Jar` files is not possible
- classloader (except `SystemClassLoader` / `ToolsSysCl`) are not supported.

Combined with the method encryption feature, several new options have been introduced:

- `-caa1` 267, `-caa2` 267 for encrypting of values from the constants pool and of method calls.
- `-jcp` 282 for defining libraries required for encryption.
- `-jff:[c|w]` 283 for defining the encryption result formats of methods (class files).
- `-jgs-` 283 for deactivating of the default generation of getter / setter and wrapper methods.

The option `-ci` for method encryption is set as default for the encryption of all Java projects.

For new projects Wibu-Systems recommends setting the options `-ci` 274 and `-jff:c` 283: This activates method encryption and the result format is still the valid class format - very important for application server, such as, JBoss, Eclipse, Tomcat etc. Please proceed as follows:

1. After setting up and encrypting the Java *AxProtector* project, navigate in *AxProtector* GUI to page **Summary**.
2. Insert the recommended options in the field "**Commandline for encryption for batch mode** 174" before the output parameter "`o:`".
3. Click the button "**Protect now**".

These settings are valid for the Java archive formats (`*.jar`, `*.war`). Moreover, it does not matter whether the protected application is a desktop or a web application.

Modular *IxProtector* protection using annotations and WUPI

If you activated method encryption, you also have the option to use extended individual software protection. Alternatively to using the AxProtector / IxProtector GUI, either set annotations annotations directly in the source code 280 or use a WUPI license list 281 generated in a XML file.

Annotations

By using annotations in the source code you may set additional definitions which classes / methods are encrypted. The following definitions can be set:

Annotation	Class	Method
no (default)	Class is encrypted/protected	Method is not protected

`@Protected`	Class is protected (corresponds to `@Protected(licenseList=0)`)
	Optional parameters:
	`licenseList`
	• `@Protected(licenseList=1)` encrypts the class using license list 1 (or another specified index entry, e.g. 2, 3 etc.).
	`scope`
	• `@Protected(scope = {Class})` specifies that only this class is encrypted.
	• `@Protected(scope = {Method})` specifies that encryption is performed for the methods only.
		Method is protected (corresponds to `@Protected(licenseList=0)`)
		Optional Parameter:
		`licenseList`
		• `@Protected(licenseList=1)` encrypts the class using license list 1 (or another specified index entry, e.g. 2, 3 etc.).

Annotation	Class	Method
	This results in encrypting all methods using a single annotation. • `@Protected(scope = {Class, Method}` specifies that encryption is performed for the class and all methods except constructors. The `scope` and `licenseList` options may be combined.	
`@Unprotected`	Class is not protected	Method is not protected (default settings which can be overwritten by using the `scope` option for all methods.)
`@EntryPoint`	Entry point for all methods	Entry point for this method. A class may not be encrypted.

WUPI License List

If you want to use a WUPI license list for modular protection, you must first create this list in a XML file. In addition, this XML file must also contain:

- *AxProtector* encryption parameter.

  Here you must transfer the parameter of your project WBC file (licensing system, license option, license handling, etc.) in the XML file.
- Java objects including package, class, and method specification.

  Here you are able not only to define method entry points but also assign objects to existing license list entries. Using the commandline option `-extract` 285 you are able to output all Java methods.

Finally, integrate the XML file in the `commandline` 285 using parameter `'@'`. Your application then is encrypted using the defined parameter.

Below see a highly simplified example for such a XML file you must create.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AxProtectorJava xmlns:wibu="http://wibu.com/2013/AxpJavaControlFile/1.0">
  <CommandLine>
    <Command>-kcm</Command>
    <Command>-f6000010</Command>
    <Command>-p13</Command>
    <Command>-cf0</Command>
    <Command>-d:6.20</Command>
    <Command>-fw:3.00</Command>
    <Command>-sl</Command>
    <Command>-nl</Command>
    <Command>-ci</Command>
    <Command>-wu1000</Command>
    <Command>-we100</Command>
    <Command>-eac</Command>
    <Command>-eec</Command>
    <Command>-eusc1</Command>
    <Command>-emc</Command>
    <Command>-car30,3</Command>
    <Command>-cml10</Command>
    <Command>-trap</Command>
    <Command>-v</Command>
    <Command>-cag15</Command>
    <Command>-jff:c</Command>
    <Command>-#</Command>
    <Command>-o:/protected/sample.jar</Command>
    <Command>/sample.jar</Command>
  </CommandLine>
  <Wupi>
    <!-- Lizenz 1 -->
    <LicenseList Index="1" Name="1">
      <License>1-1</License>
      <License>1-2</License>
    </LicenseList>
    <License Name="1-1">
      <Type>CodeMeter</Type>
      <FirmCode>6000010</FirmCode>
      <ProductCode>13</ProductCode>
      <FeatureCode>0</FeatureCode>
      <SubSystem>Local</SubSystem>
      <Access>UserLimit</Access>
      <MinimumDriverVersion>6.20</MinimumDriverVersion>
```

```
      <MinimumFirmwareVersion>3.00</MinimumFirmwareVersion>
      <UserData>None</UserData>
    </License>
    <License Name="1-2">
      <Type>CodeMeter</Type>
      <FirmCode>6000010</FirmCode>
      <ProductCode>13</ProductCode>
      <FeatureCode>0</FeatureCode>
      <SubSystem>Local</SubSystem>
      <Access>UserLimit</Access>
      <MinimumDriverVersion>6.20</MinimumDriverVersion>
      <MinimumFirmwareVersion>3.00</MinimumFirmwareVersion>
      <UserData>None</UserData>
    </License>
  <Jar MethodProtectionLicenseList="None" ClassProtectionLicenseList="0" EntryPoint="false" >
    <Package Name="com.wibu.guitest.actionclasses" >
      <Class Name="ButtonA" ClassProtectionLicenseList="None" MethodProtectionLicenseList="0" >
        <Method Name="method1" Desc="()V" MethodProtectionLicenseList="None"  />
        <Method Name="method3" Desc="()V" MethodProtectionLicenseList="None"  />
      </Class>
    </Package>
    <Package Name="com.wibu.guitest" ClassProtectionLicenseList="0" >
      <Class Name="GuiTest$1" ClassProtectionLicenseList="None" MethodProtectionLicenseList="0" >
        <Method Name="windowClosing" Desc="(Ljava/awt/event/WindowEvent;)V" MethodProtectionLicenseList="1"  />
      </Class>
      <Class Name="GuiTest" ClassProtectionLicenseList="None" MethodProtectionLicenseList="0" >
        <Method Name="main" Desc="([Ljava/lang/String;)V" MethodProtectionLicenseList="None"  />
      </Class>
  </Jar>
  </Wupi>
</AxProtectorJava>
```

Commandline options

Option	-cml<n>
valid for	Java

Outputs encrypted methods.
`<n>` represents the minimum number of method instructions in order to encrypt a method. Default is `<n>=5`.
Deactivating this feature by using `'-CML0'`.

Option	-ja:"params"
valid for	Java

Specifies arguments transferred to the Main Class at runtime.

Option	-jb:<number>
valid for	Java

Activates or de-activates a special error exception handling.

> Contact Wibu-Systems support for more details.

Option	-jcl:<ClassLoader>
valid for	Java

Specifies an alternative WIBU ClassLoader.
Currently, the following ClassLoader are available:

`ClassLoader`	`CassLoader` derived from `java.net.URLClassLoader`
`DelegateClassLoader`	`ClassLoader` derived from `java.lang.ClassLoader`

Option	-jcp:<additional Jar files>
valid for	Java

Announces *AxProtector* that external libraries exist additionally to the ones stated in the classpath.

Specifying libraries required for encryption

With immediate effect, all externally referenced Jar files as part of the protected Jar file are required on encryption, i.e. all classes specified as classpath in the manifest have to be provided. This option allows the announcement of further external libraries.

e.g. `-jcp:javaee-api-7.0.jar;someapi.jar`

Option	-jd:vmin[-vmax]
valid for	Java

Option	-jd:vmin[-vmax]

Specifies the used minimum (and maximum) Java version.
The version must match the format as specified in system property `java.version`. For versions up to Java 1.8 the final number can be omitted.
Starting with Java 9 version names change. Now the first number refers to a major release, the second to a smaller update.

e.g. `-jd:1.4-1.5.0` allows the runtime versions from 1.4 to Java 5 Update 0.
`-jd:1.4.0-9.1` allows the runtime versions from Java 4 Update 0 to Java 9 Update 1.

Option	-jff:[c|w]

valid for Java

Defines the encryption result of methods.

Creating machine-readable Class files

If you set the parameter -ci for method encryption (IxProtector), for external application which analyze annotations, this option allows to output the encryption results either as encrypted (not readable) class files or as again valid class files.
Valid class files then compose of the method bodies and fields with annotations of the original classes. The encrypted bytecode in embedded in the constants sections.

Setting the option `-ci` for method encryption is required.

`w`	creates encrypted (not readable class files (default).
`c`	creates as result valid class files.

Option	-jfx

valid for Java

Initializes JavaFX. This is required for encrypting some JavaFX applications. If JavaFX is not initialized, the error message 'Toolkit not initialized' displays.

Option	-jgs-

valid for Java

Deactivates the generation of getter / setter methods and the generation of wrapper methods as set by default on using option `-ci` 274.

Option	-jh:[a|e|n]

valid for Java

Hides or renames encrypted classes.

`a`	Renames all classes according to the pattern `'<MyClass>.class.wibu'`.
`e`	Renames only encrypted class names according to the pattern `'<MyClass>.class.wibu'`. This corresponds to the default setting.
`n`	Does not rename encrypted classes.

Option	jip:<OS-arch>[;<OS-arch>]

valid for Java

Specifies the operating system-specific native runtime component (DLL) to be added to the JAR archive on activating the feature IP Protection 287.
Using the option `OS-arch`, one or more components of specific operating system architecture(s) can be selected.

Option	Operating system architecture
`win-32`	Windows 32 bit
`win-64`	Windows 64 bit
`win`	all Windows systems
`mac-64`	macOS 64 bit
`mac`	all macOS systems
`lin-32`	Linux 32 bit
`lin-64`	Linux 64 bit
`lin`	all Linux systems
`lin-armhf-32`	Linux-arm/hf 32 bit
`lin-armhf_64`	Linux-arm/hf 64 bit
`lin-armhf`	all Linux-arm/hf systems
`std`	includes the options `win`, `mac`, and `lin`, i.e. `win-32`, `win-64`, `mac-64`, `lin-32`, and `lin-64`.
`all`	includes all DLLs currently known to *AxProtector*.
`none`	no DLL will be included.

If `LicenseList 0` contains an "IP Protection" license and the -jip option is not explicitly set, all default DLLs are added (equivalent to the -jip:std option).

Option	-jm:<Main-class>
valid for	Java

Specifies the starting Main Class.

Option	-jn:[p|s|t]
valid for	Java

Activates the native class load process.

This process requires an intervention into the source code of the application.

p	Uses JVMPI (Java Virtual Machine Profiler Interface).
s	Uses the Java 6 module (not yet supported).
t	Uses JVMTI (Java Virtual Machine Tool Interface).

Option	-jl[w|b]:<list>
valid for	Java

Specifies which classes are encrypted.

w	Whitelist: all classes matching this list are encrypted.
b	Blacklist: all classes matching this list are not encrypted.
<list>	Holds the complete class or package name, parts of the name (fragments). The list items are separated by ':'. e.g. '-jlw:com.wibu.:de.wibu.MainClass'

Option	-jo[[a:<jars>],[losf],[e:[e]]:<list>
valid for	Java

Specifies output options

a:	<jars> Adds the specified *.jar file to the output. e.g. -joa:CodeMeter.jar,WibuKey.jar adds the contents of those jar files to the output specified by -o.
l:	Lists the license information of an encrypted *.jar file.
o:	Separates the output in two different *.jar files. The WIBU runtime classes are created in form of a OSGI bundle named WibuXpm4JRuntimePlugin.jar. The dependencies of the encrypted source *.jar from this OSGI bundle are automatically created at encryption. Please note that the own MessageHandler must locate in WibuXpm4JRuntimePlugin.jar or must be copied there. Make sure, that the package name of the MessageHandlers is explicitely specified in the MANIFEST.MF of the Plugins as "Export Package", if the MessageHandler package has not not been already exported. The MessageHandler contained in *AxProtector* are already exported. With the option -joo now also optionally specifying the name and the versions is supported, e.g. -joo:com.wibu.runtime;1.2.3 This covers the following: • the generated OSGI Plugin by convention has the name com.wibu.runtime_1.2.3.jar • The version number is entered in the Runtime Plugin • The dependencies in the encrypted jar is entered with a version dependency Please note that the data, i.e. name and version, as part of the bundles.info must be updated each time.
m:	Encrypts modular jar files created using Java 9. Then on encryption a modular Wibu Runtime jar file with the name com.wibu.xpm.jar is created. Dependencies of the protectee to this Wibu Runtime jar file are automatically added to the module-info.class of the protectee.
s:	Separates the output in two different *.jar files. The WIBU runtime classes and the source *.jar files are not merged. This option is recommended for servlets, or when you encrypt several *.jar files in a project to save space. The created WibuXpm4JRuntime .jar file must be manually added to the class path.
f:	Separates the output in three different .jar files. This is an extension of the option -jos and creates a file with the name WibuXpm4JO<outputfile>.jar holding a few options only.
e: [e]	Specifies which files are excluded from the output. [e]: specifies the file to be excluded, e.g. com/wibu/xpm/encrypted.

Option	-jx
valid for	Java

Option	-jx
	Exits the application using the `System.exit()` call after the 'Main-Class' main method has returned a value.

By default, executing the encrypted Java application covers the output of diverse logging information.

By customizing the logging configuration of Java you may manage log levels. Using the file `logging.properties` and coupled log levels you have the option to handle which important and urgent information of which class is to be logged and displayed.

You may do this globally for the Java Virtual Machine (JVM) or set parameters directly for the encrypted application.

The original file locates in the Java installation directory below the `lib` directory.

If you do not want the original `logging.properties` file, you are able to specify the alternative file using the system property `java.util.logging.config.file`:
For example:

```
java -Djava.util.logging.config.file=D:\Tests\JavaTest\logging.properties -jar
JavaTest0.jar
```

If in the `logging.properties` file in the area `Facility specific properties` the log level for the *AxProtector* Runtime (`com.wibu.xpm.Runtime`) is set to the value `WARNING`, the logging information does display in the console output.

For example:

```
############################################################
# Facility specific properties.
# Provides extra control for each logger.
############################################################

# For example, set the com.xyz.foo logger to only log SEVERE
# messages:
com.xyz.foo.level = SEVERE
com.wibu.xpm.Runtime.level=WARNING
```

7.7.6 Operational Options

Option	-!
valid for	32 64 Mac Java
	Creates a command file (*.wbc).

Option	-V
valid for	32 64 Mac Java
	Activates the verbose mode.
	In the case of .NET use `-vn`.

Option	-#[File]
valid for	32 64 .NET Mac Java
	Prints the logging to the specified `[File]`. This option exists next to automatic output to the `AxProtector*.log` [directory: `\Users\<username>`, i.e. `%USERPROFILE%`].

Option	-EXTRACT
valid for	.NET
	Prints content of assembly application file (enter `-EXTRACT?` for details).

Option	-? or -h
valid for	32 64 Mac Java
	Shows the options in commandline mode.

Option	@cmds.wbc
valid for	32 64 Mac Java
	Adds a specified `*.wbc` file holding parameters for the automatic encryption of an executable file.

7.8 Advanced AxProtector Options

In addition to working with the graphical user interface (GUI) or the command line, you also have the option of combining both editing options. This is required for some advanced *AxProtector* options.

Editing of `*.wbc` and/or `*.xml` files

The GUI window "Summary" window displays a summary of all the settings you have made to automatically protect your application.

The contents of this page can be copied to a `*.wbc` (WIBU Configuration file) or `*.xml` file (for *AxProtector* .NET) for later reuse.

Please proceed as follows:

1. Click the **"File – Export"** menu item to copy the window content from the GUI into a `*.wbc` or `*.xml` file.
2. Open the saved file and modify as desired.
3. Save the file.
4. Protect your application using these modified settings via commandline input 285 `AxProtector.exe @*.wbc` or `AxProtector.exe @*.xml`.

7.8.1 Translocated Execution

AxProtector supports the technique to translocate, i.e. shift the execution of selected functions to other arbitrary processing areas. At the same time, data at the original location is not changed.

Translocated Execution is implemented by setting a parameter in the WBC (WIBU-SYSTEMS Control File) file.

Parameter	OOPE=[1,2,5]
valid for	
	translocates the execution of selected functions to other arbitrary processing areas. If `OOPE` (out of place execution) is set, in the WBC file for the respective function the `Length` attribute is ignored. A function which is translocated has to be processed as a single function. Functions which themselves hold try / catch constructs cannot be translocated. *AxProtector* detects this and issues an error warning.

1	Translocation without using *CodeMeter Software Protection*-API WUPI.
	On calling the function translocates itself and removes itself after being left.
	In the WBC file in the function descriptive part the parameter is set to a value of `1`.
	`[Function 1]`
	`FunctionName = examplefuntion`
	`ProtectionOptions = 1`
	`Length = 100%`
	`OOPE = 1`
	The error case is part of the source code and throws an exception. For this the function must be framed by a programmed try - catch block receiving a WupiException. Using `GetError()` allows to read the respective error code.
	The definition of the exception call locates in the `wibuixap.h` header file, e.g.
	`#include <wibuixap.h>`
	`void DoSomething()`
	`{`
	`try {`
	`ProtectedFunction();`
	`} catch (const WupiException& e) {`
	`printf("Errorcode: %d\n", e.GetError());`
	`}`
	`}`
	The working of this exception handling requires that exactly a single CPP source code file must hold the macro `WUPI_INITIALISE`. For example:
	`// Impl.cpp`
	`#include <wibuixap.h>`
	`WUPI_INITIALISE`
	If `WUPI_INITIALISE` has not been set, *AxProtector* returns a respective error message and aborts the encryption.

2	Translocation with using *CodeMeter Software Protection*-API WUPI.
	In the WBC file in the function descriptive part the parameter is set to a value of `2`.
	`[Function 1]`
	`FunctionName = examplefuntion`
	`ProtectionOptions = 1`

Parameter	OOPE=[1,2,5]

	`Length = 100%`
	`OOPE = 2`
	Before execution the functions must be initialized using *WupiDecryptCodeId*. Then the actual function translocation is performed. In the case of an error, the Wupi method returns FALSE and using *WupiGetLastError* displays the error code.
	Using *WupiEncryptCodeId* re-encrypts the function. For example:
	`#include <wibuixap.h>`
	`void DoSomething()`
	`{`
	`if (!WupiDecryptCodeId(123)) {`
	`printf("Decryption failure: %d\n", WupiGetLastError());`
	`return;`
	`}`
	
	`ProtectedFunction();`
	
	`if (!WupiEncryptCodeId(123)) {`
	`printf("Encryption failure: %d\n", WupiGetLastError());`
	`}`
	`}`

5	Moves the function to be translocated for several second into the cache memory. This means the function must not be decrypted and encrypted each time if called. This results in an increase of performance.
	This parameter is valid only if combined with translocation without using *CodeMeter Software Protection*-API WUPI (Parameter `1`).
	`[Function 1]`
	`FunctionName = examplefuntion`
	`ProtectionOptions = 1`
	`Length = 100%`
	`OOPE = 5`

7.8.2 License allocation in license lists

The parameters `CheckAutoReconnect` and `SwitchLicense` are available for handling license assignments in license lists. They are used to regulate the reassignment of licenses in license lists if accesses have become invalid, e.g. at the end of the maximum document duration of a handle.

Parameter	Description
valid for	.NET Java
`CheckAutoReconnect`	specifies, if an allocated license that is invalid, is automatically reassigned when accessing a license list (WupiEncryptCode/WupiDecryptCode). In this case, no error message "Invalid handle" is issued, but the license is released and immediately reassigned. The default value is FALSE.
`SwitchLicenses`	specifies whether any other license from the license list can be used with an AutoReconnect. If the setting is set to FALSE, only the already allocated license entry is searched for. With a manual release and reassignment, the complete license list is always searched again from top to bottom. The default value is FALSE.

The parameters are set 286 in the license list description part of the `*.xml` file.

e.g.

```
<LicenseList Index="1">
  <License>CM10-1001</License>
  <License>CM10-1002</License>
  <CheckAutoReconnect/>
  <SwitchLicenses/>
</LicenseList>
```

7.8.3 IP Protection - protecting know how

Available for 32 64 .NET Mac Java.

For Java see -jip 283.

The software manufacturer (ISV, Independent Software Vendor) can guarantee the protection of the intellectual property of applications / libraries without being bound to a licensing system by using a (self-selected) fix key: an *AxProtector* for protection without licensing.

This (self-selected) secret key is permanently stored in the license list and used for cryptographic operations.

IP Protection and *Software Protection API*

In addition to *AxProtector*, IP Protection also supports *IxProtector* protection for function encryption via license lists (*Software Protection* API, *WUPI*, *Wibu Universal Protection Interface*). The following WUPI functions are supported:

- *WupiCheckLicense()*
- *WupiCheckDebugger()*
- *WupiDecryptCode()*
- *WupiEncryptCode()*

> Please note that *IP Protection* does not support all other WUPI functions. If such a function is called a separate error message displays (see the separate help document on *Software Protection API (WUPI).*

IP Protection and mixing licensing systems

IP Protection also supports mixing of licensing systems. It is possible to use IP Protection for the standard license list and to define separate license lists with *WibuKey* or *CodeMeter*.

An example scenario here would be that the ISV uses IP Protection to protect a basic version of its application against reverse engineering. For additional features of a Pro version, the ISV uses *IxProtector* and a license list that makes these features usable via *WibuKey* / *CodeMeter* licenses.

Use

By default, *AxProtector* generates this 16 byte key depending on the input file and the selected encryption options. With unchanged parameters, then the key remains constant and ensures reproducible encryption.
On the command line, the key is determined using the -kip 264 option.

Optionally, the key can also be specified as a hex value in the command line.

The parameters are set 286 in the license list description part of the `*.wbc` file or XML file in the case of .NET.

e.g. Default: *AxProtector* generates the key

```
[LicenseList 1]
FK123

[License FK123]
Type=IPProtection
```

Optional: self-selected key

```
[LicenseList 1]
FK123

[License FK123]
Type=IPProtection
Key = 0x11223344556677881122334455667788
```

for .NET Java

```
<License Name="FK123">
  <Type>IPProtection</Type>
  <Key>0x12345678123456781234567812345678</Key>
</License>
```

or

```
<License Name="FK234">
  <Type>IPProtection</Type>
</License>
```

8 Individual Software Protection

Next to using *AxProtector* for automatic software protection where the source code of your application remains unaltered, *CodeMeter* also provides several options to individually integrate software protection into your application and to increase security.

> Please note that using *IxProtector* modifies the application to be protected.
> This may affect other already applied security measures, e.g. the use of signatures.
> In such cases, you must perform other already applied security measures, e.g. the use of signatures after *IxProtector* has made the modifications.

***IxProtector* (Tool of *CodeMeter Protection Suite*)**

With *IxProtector* you have a protection technology at hand which allows you to define and protect individual parts (segments) in the source code while developing software. Then during runtime, these segments are linked to different license entries.

Software Protection API WUPI

WUPI (*WIBU Universal Protection Interface*) represents the tool used to decrypt segments protected by *IxProtector* at runtime. This *Software Protection API* is lean, comprises only a few but essential functions, and is standardized and applicable for a variety of programming languages.

Core API

When additional requirements have to be met, for example, the encryption or decryption of any kind of data, more extensive data read-out, personalization, etc., the *CodeMeter Core API* 297, as the interface on which all other APIs and protection mechanisms are based, provides you with many functions. By using the interactive *CodeMeter API Guide* 300 you can quickly generate the matching source code to be integrated into your software.

> Wibu-Systems recommends the combined use of automatic and individual software protection to increase security. Moreover, the security mechanisms of *AxProtector* and *IxProtector* are constantly being developed and improved. The recompilation of your software is not required; simply re-encrypt it by using *AxProtector* or *IxProtector*.

Easy combination: automatic and individual software protection

The combination of automatic and individual software protection is quite easy. First, *IxProtector* is integrated in *AxProtector*, i.e. you use both protection technologies at the same time. And second, transitions between the single protection levels are smooth, because the identification of handles 289 provides access to the same license entries. For example, WUPI allocates the license entry also used by *AxProtector* and by calling ***WupiGetHandle*** 291 you can read out the entry to be further processed using *CodeMeter Core API*.

8.1 Handles

Handles implement access and identify license entries. Eventually handles can become invalid which happens in the following cases:

1. You use a protected software or a software applying *CodeMeter Core* API / *Wupi* API and while doing so restart the *CodeMeter* service. This renders all handles invalid and reusing handles in your software result in error 106.
2. *CodeMeter* itself automatically release handles, if:
   - **a)** the *CmDongle* was disconnected to which the handles previously referred to,
   - **a)** the process which opened the handles no longer exists. The check for the processes is performed for each new license access, e.g. using *CmAccess2()*, however at latest after 1 minute.
   - **a)** the handles were not used longer than the specified *CleanUpTimeOut* value (default 120 minutes).

   In order to avoid the automatic release of handles as described in **2.c)**, you should:
   - manually release all handles no longer required after the access using *CmRelease()* / *WupiFreeLicense()*.
   - regularily use handles responsible for counting licenses or license management as in *AxProtector* runtime check on encryption or on calling *CmCrypt()* / *WupiCheckLicense()* in the code.
     Using *CmCrypt()* you may encrypt/decrypt data increasing security.
     Morever, at *CmCrypt()* / *WupiCheckLicense()* also the license itself is checked, i.e. whether the license is still valid and meanwhile the unit counter has not a value of 0 or the expiration time has reached.

   In addition, you may also set an own *CleanUpTime* at *CmAccess2()* using the CMCREDENTIAL structure and the member 'mulCleanupTime'. This own time will overwrite the registry default value in registry entry HKEY_LOCAL_MACHINE\SOFTWARE\WIBU-SYSTEMS\CodeMeter\Server\CurrentVersion\CleanUpTimeOut.

   This may look like as follows:

```
CmCredential cmCred = new CmCredential();
cmCred.CleanupTime = 240;

CmAccess2 cmAcc = new CmAccess2();
cmAcc.Credential = cmCred;
cmAcc.Ctrl |= CmAccess.Option.UserLimit;
cmAcc.FirmCode = 10;
cmAcc.ProductCode = 13;

HCMSysEntry hcmse = cmApi.CmAccess2(CmAccessOption.Local, cmAcc);
```

8.2 IxProtector (Tool of CodeMeter Protection Suite) and Software Protection API (WUPI)

The *IxProtector* protection technology allows you to define 'real' single segments (modules, functions) in the source code when developing an application, encrypt them, and then link them to license entries at runtime by using index-based placeholders. The *CodeMeter Software Protection API WUPI* (*WIBU Universal Protection Interface)* assists you in this process.

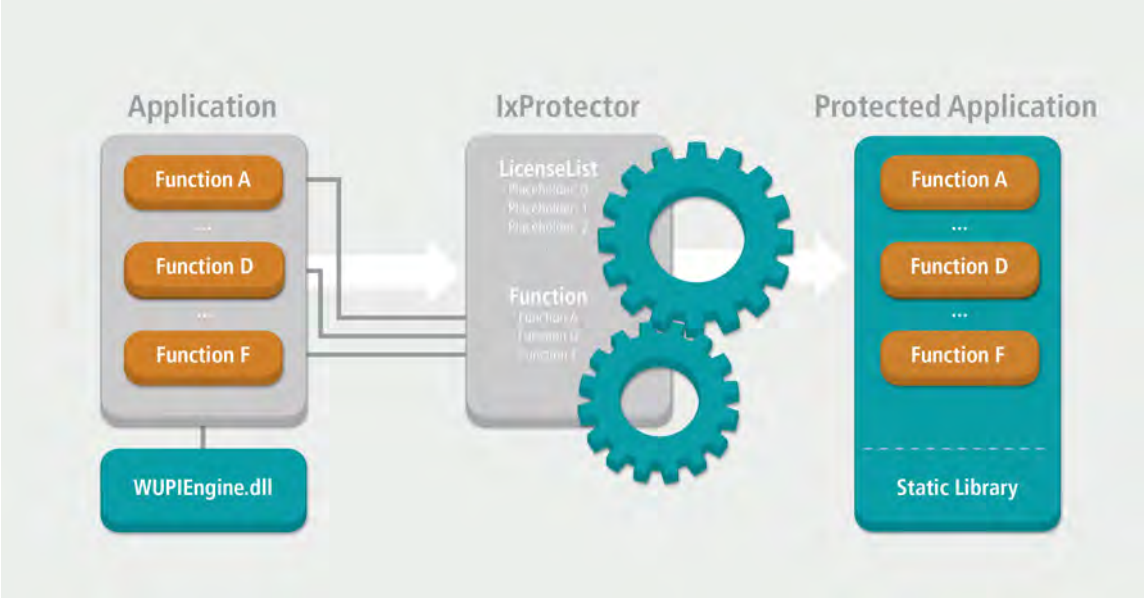


Figure 207: *IxProtector - Software Protection API* - WUPI

The interaction of *IxProtector* and WUPI is suited for the following application areas:

- Protecting and activating single modules of an executable file, i.e. modular software protection , using specified function and license lists.
- Integrating individual license queries. You freely define where and when,
- Encrypting 'real' code fragments to increase the level of security.
- Implementing pay-per-use functionalities, i.e. decrementing counters for specified software actions.
- Specifying which kind of anti-debugging measures *AxProtector* applies at which point in time.
- Simultaneously implementing for all licensing systems (*WibuKey*, *CmDongle*, and *CmActLicense*) while still able to change for future encryptions.
- Accessing a license allocated by *AxProtector* for further use in *CodeMeter Core API*.

Using WUPI you implement:

- Easy-to-accomplish protection: available for many programming languages with a one-time implementation in the same executable file without recompiling your source code,
- constantly updated protection: continuing the security-related revisions and improvements of *AxProtector* functionalities without requiring changes to your source code.

8.3 WUPI Functions

The lean and effective*CodeMeter Software Protection API* provides the following functions.

> Except for the functions ***WupiEncryptCode()*** and ***WupiDecryptCode()*** referring to in *IxProtector*, all other functions relate to license lists.

Simulation of encryption during development for the project types 32 64 Mac

Up to *AxProtector* version 10.40, *WUPI* functionalities could only be used and tested with *IxProtector* if the application to be protected was actually encrypted. This required the integration of the *WUPI* engine into the application by the developer and the use of a dummy library (`WupiDummy.lib`, `*.so`), which could partially provide functionalities. For example, WupiCheckLicense only returned `True`.

> Starting with *AxProtector* version 10.40 this behavior has changed by default.
> Now *WUPI* calls are also possible in the unencrypted state of the application.
> The calls are not supported: WupiCheckDebugger, WupiRead... and WriteData....

Encryption is now simulated and allows developers to test *WUPI* implementation functionality using valid license information in test environments, e.g. by calling WupiQeryInfo. For this purpose, the *WUPI* engine now imitates a normal behavior of the encrypted state of an application.

In order to tell the *WUPI* Engine which information belongs to which application, the corresponding WBC file(s) are searched. To do this, the WBC files must be located in the same directory at runtime, i.e. the execution path for an application or storage location in the case of plugins.
The WBC files are searched for the original file name of the application, i.e. not for the file name specified with the '-o: [279]' option. If the original file name of the application matches in several WBC files, the WBC file in which this name was found for the first time is used. This also applies to a deactivated simulation (`Active` then has a value of `0`).
The search parameters function independently of upper and lower case and, in the case of several files, according to alphabetical order.
In addition, the WBC file can be told whether the simulation should be activated or deactivated. This is done via the new Topic `[Simulation]`:

```
[Simulation]
Active = 0 or 1
```

By default, simulation is enabled and `Active` has a value of `1`

> All users who want to continue working with *WUPI* must set the Topic `[Simulation]` and set `Active` to a value of `0`.

Access API: Allocating and Releasing Licenses

WupiAllocateLicense()	This function allocates a license (LicenseList) for the selected licensing system.
	Return Value
	`TRUE (1)` if the function was successfully executed, otherwise `FALSE (0)` if an error has occurred.
WupiFreeLicense()	This function releases a license (LicenseList) for the selected licensing system.
	Return Value
	`TRUE (1)` if the function was successfully executed, otherwise `FALSE (0)` if an error has occurred.
WupiGetHandle()	This function returns the actual native handle of the license (LicenseList).
	Return Value
	The actual native handle of the license is returned. Otherwise 0 is returned if an error has occurred.

Encryption and Decryption API

WupiEncryptCode()	This function encrypts a function (Function).
	Return Value
	`TRUE (1)` if the function was successfully executed, otherwise `FALSE (0)` if an error has occurred.
WupiDecryptCode()	This function decrypts a function (Function).
	Return Value
	`TRUE (1)` if the function was successfully executed, otherwise `FALSE (0)` if an error has occurred.

Security API

WupiCheckDebugger()	This function (LicenseList, Level) checks whether the protected application runs within a debugger, a debugger runs on the system, or a Kernel debugger is installed.
	Return Value
	`TRUE (1)` if the function has detected a debugger attack, otherwise `FALSE (0)`.
	Please note that this security function can be used for *AxProtector* protected applications only. This function cannot be used for *IxProtector* protected applications.
WupiCheckLicense()	This function checks a license (LicenseList) for the selected licensing system.
	Return Value
	`TRUE (1)` if the function was successfully executed, otherwise `FALSE (0)` if an error has occurred.
WupiDecreaseUnitCounter()	This function (LicenseList, Units) decrements an *Unit Counter* in the specified license by the defined number of units.
	Return Value
	`TRUE (1)` if the *Unit Counter* was successfully decremented, otherwise `FALSE (0)`.

Information Query

WupiQueryInfo()	This function returns information on an entry (LicenseList) or on a *CmContainer*.
	Return Value
	If the queried value exists it is returned. If an error occurred or the queried information does not exist `-1` is returned including a related error code.

Reading and writing of data

When using *CodeMeter Software Protection API* WUPI at runtime of the protected application you have the option to read data you previously saved to the *CmContainer*, for example, to use the saved data for the program functionality. Reading previously saved data is provided by the WUPI functions ***WupiReadData*** or ***WupiReadDataInteger***.

In *CodeMeter* the actual data is stored in the *Hidden Data* field and the data, for example, programmed using *CmBoxPgm*.

The data is saved using indexed entries (type). The licensor (software developer) is able to use 128 *Hidden Data* types (0-127).

> 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.

In the case of 128 *Hidden Data* entries and the default length, 30,976 Bytes are readable. This increase to 32,768 bytes using the maximum length.

Reading data from a *Hidden Data* field in a *CmContainer* requires the specification of an access code, i.e. the *Hidden Data Access Codes* (HDAC). This HDAC may correspond to an automatically calculated derived value. This calculated derived value consists of several parameters, such as, for example, *Firm Code*, *Product Code*, etc.

> Wibu-Systems recommends using this derived value.

When you have used the *Programming API (HIP)* to write the data into the *CmContainer* you cannot use the automatically derived value as HDAC. Then you are required to manually specify the necessary *AxProtector* settings using the `*.wbc` file.

> In the case you do not use the *Programming API (HIP)* to write the data into the *CmContainer* you cannot use the automatically derived value as HDAC. Then you are required to manually specify the necessary *AxProtector* settings using the `*.wbc` file.

The license definition area of the `*.wbc` files then looks as follows:

```
[License CM1]
Type=CodeMeter

UserData=read ;          <- required, activates data reading mode
FirstHiddenData=13 ;     <- optional, default value equals 0
HiddenDataAcessCode=42 ; <- optional, default value is the derived value as HDAC
DataBlockSize=240 ;      <- optional, default value equals 242
```

WupiReadData (int iLicenseList, int iOffset, void* pvData, unsigned int cbData);

This function reads raw data which has been previously stored at a specified location from the CmStick.
This function can be used for all programming languages working with pointers, i.e. special variable holding memory addressed.
For the other programming languages the function ***WupiReadDataInteger*** 292 is provided.

Parameter	Description
`iLicenseList`	specifies the number of the license list index.
`iOffset`	holds in number of bytes the offset from the start of the data block.
`pvData`	holds the data array to be filled.
`cbData`	holds the number of bytes for `cbData`.

Return Value

Number of bytes stored in `pvData`.
If the return value has a value of **0** call ***WupiGetLastError*** 293 to obtain more detailed information.

WupiReadDataInteger(int iLicenseList, int iOffset);

This function reads raw data which has been previously stored at a specified location from the *CmContainer*.
The data is read 2 bytes at a time.
This function can be used for all programming languages.
For programming languages working with pointers, i.e. special variable holding memory addressed, Wibu-Systems recommends the function ***WupiReadData*** 292.

Parameter	Description
`iLicenseList`	specifies the number of the license list index.
`iOffset`	holds in number of bytes the offset from the start of the data block.

Return Value

The return value has a size of 4 bytes. It is separated in 2 upper bytes holding status flags for error and message handling and 2 lower bytes holding the data.
The upper 2 bytes may, for example, hold the following values:

```
#define WupiRDError (0x80000000)
#define WupiRDMoreDataAvail (0x40000000)
```

WupiWriteData (int iLicenseList, int iOffset, void* pvData, unsigned int cbData);

This function writes raw data into a *CmContainer* that was previously prepared for writing.
This function can be used for all programming languages working with pointers, i.e. special variable holding memory addressed.
For the other programming languages the function *WupiWriteDataInteger* 293 is provided.

Parameter	Description
`iLicenseList`	refers to the license list index.
`iOffset`	contains the offset of bytes from the start of the data area.
`pvData`	contains the data array to be written.
`cbData`	contains the number of bytes of `cbData`.

Return Value

This function returns FALSE (0) if an error occurs, otherwise TRUE (1).
If the return value has a value of 0 call *WupiGetLastError* 293 to obtain more detailed information.

WupiWriteDataInteger(int iLicenseList, int iOffset, int iData);

This function writes raw data into a *CmContainer* that was previously prepared for writing.
The data is written 2 bytes at a time.
This function can be used for all programming languages.
For programming languages working with pointers, i.e. special variable holding memory addressed, Wibu-Systems recommends the function *WupiWriteData* 293.

Parameter	Description
`iLicenseList`	refers to the license list index
`iOffset`	contains the offset of bytes from the start of the data area.
`int iData`	contains the data to be written.

Return Value

This function returns FALSE (0) if an error occurs, otherwise TRUE (1).
If the return value has a value of 0 call *WupiGetLastError* 293 to obtain more detailed information.

Error API

WupiGetLastError()

This function returns the actual defined error code of the actual defined license type (LicenseList).

Return Value

`wibu::UpiErrorNoError (0)` -->no error occurred.
`wibu::UpiErrorNoDefaultLicense (-1)`
--> no default license is set, i.e. the application is not additionally automatically encrypted.
`wibu::UpiErrorLicenseNotFound (-2)`
--> the specified index for a license could not be found.
`wibu::UpiErrorFunctionNotFound (-3)`
--> the specified index for function could not be found.
`wibu::UpiErrorRuntimeTooOld (-4)`
--> the drivers of the licensing system in use are outdated.
`wibu::UpiErrorDebuggerDetected (-5)`
--> a debugger attack had been detected.

8.3.1 WUPI: example of index-based placeholders

Index-based Placeholders

In the programming sequences of your application, the *CodeMeter Software Protection API* WUPI links software protection mechanisms and license queries with parts of the source code using index-based placeholders. In the following, excerpts from the sample application "Second Sample" show you how to create modular software protection via WUPI.

> You will find the full example after installing the *CodeMeter* SDK for respective programming languages in the directory "`%\Users%\Public\Documents\WIBU-SYSTEMS\Software Protection`".
> Alternatively, find the samples using the navigation item **"Start | All Programs | CodeMeter | Samples"** or via *CodeMeter Start Center* 49.

The basic task in this example is to create, for end-users, a copy-protected application. The use of the application requires matching entries in the *CmContainer*. In order to use the different modules, the end-user will need additional matching entries.

The creation comprises five single steps:

1. defining modules 294,
2. creating index-based license and function lists 294,
3. programming license entries 296,
4. integration into the source code 297,
5. encryption of the application 297.

8.3.1.1 Definition of Modules

The functional scope of the simple text editor of the "Second Sample" is modular by design. Next to the "Save" function as part of the general license the function "Change Font" exists which requires a separate license.

8.3.1.2 Placeholders in IxProtector License and Functions Lists

The information of the table below is sufficient for the subsequent connection between *IxProtector* and the WUPI function calls made from within the source code for creating index-based placeholders.

The following overview summarizes the required information you need for the later completion of the license and function lists.

Modulle	Firm Code, Product Code, Feature Code	Function Name
Basic License	`6000010:201000:1`	Save
Change Font	`6000010:201001:1`	ChangeFont

Table 6: Second Sample – Overview

To create the placeholders, please proceed as follows:

> Please find the *AxProtector* project files for the respective programming languages also in the directory "`%\Users\Public\Documents\WIBU-SYSTEMS\Software Protection`"

1. Activate *IxProtector* in *AxProtector* in the **"Advanced Options"** input window.

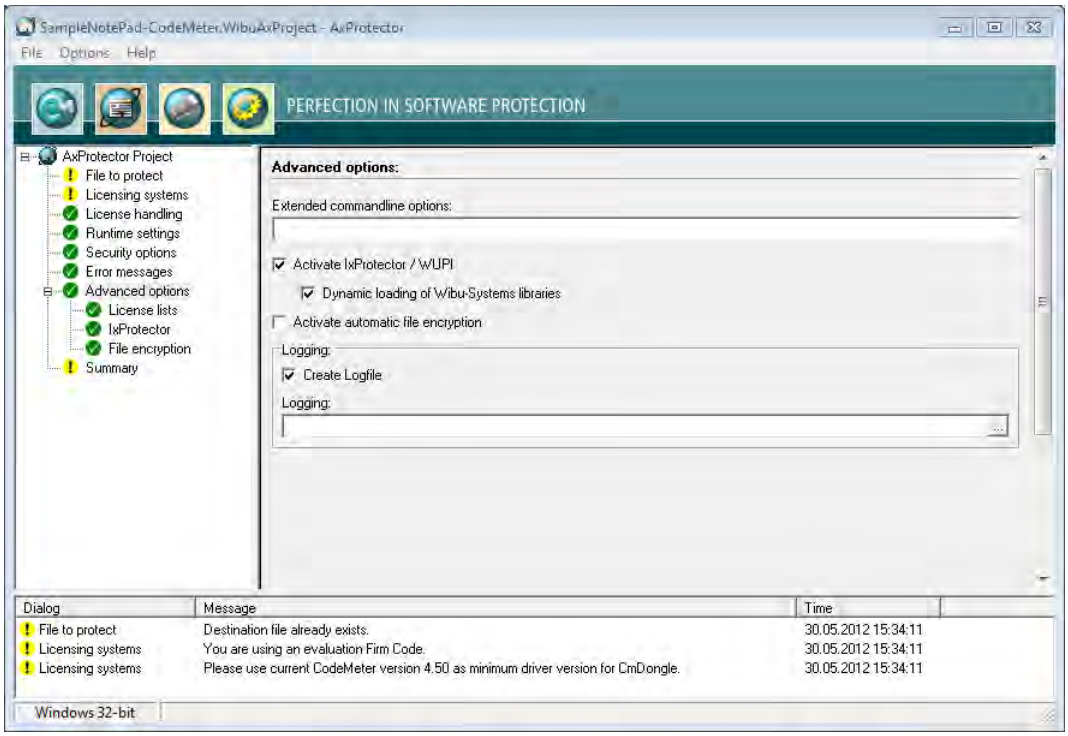


Figure 208: Activate *IxProtector* in *AxProtector*

> With this option *IxProtector* finds the related source code segments, and encrypts them before *AxProtector* wraps a protection envelope around the compiled application.
> If you want to use *IxProtector* without *AxProtector*, select the project type "IxProtector Only[196] ".
> Unless you have a special reason, Wibu-Systems recommends using *IxProtector* within *AxProtector*.

2. Navigate to the **"License Lists"** input window.

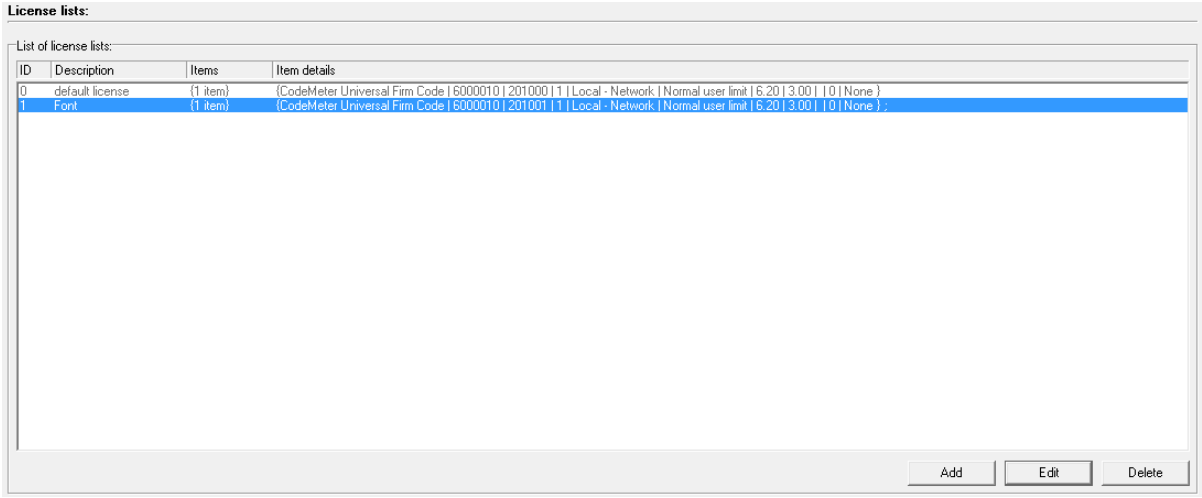


Figure 209: *IxProtector* - License List

License lists allow to summarize licenses with different license elements (licensing system, ***Firm Code***, ***Product Code*** etc.) into single entries. A single entry may hold several license elements.

> Entries in license list can hold all Wibu-Systems licensing systems (*WibuKey*, *CmDongle*, and *CmActLicense*). You can later change an assignment to single licensing systems without altering the source code. However, then the changed license information has to become part of the encryption.
> The license list entry of `0` describes the license to which *AxProtector* refers.

3. Select the license entry item with ID 1 and click the **"Edit"** button.

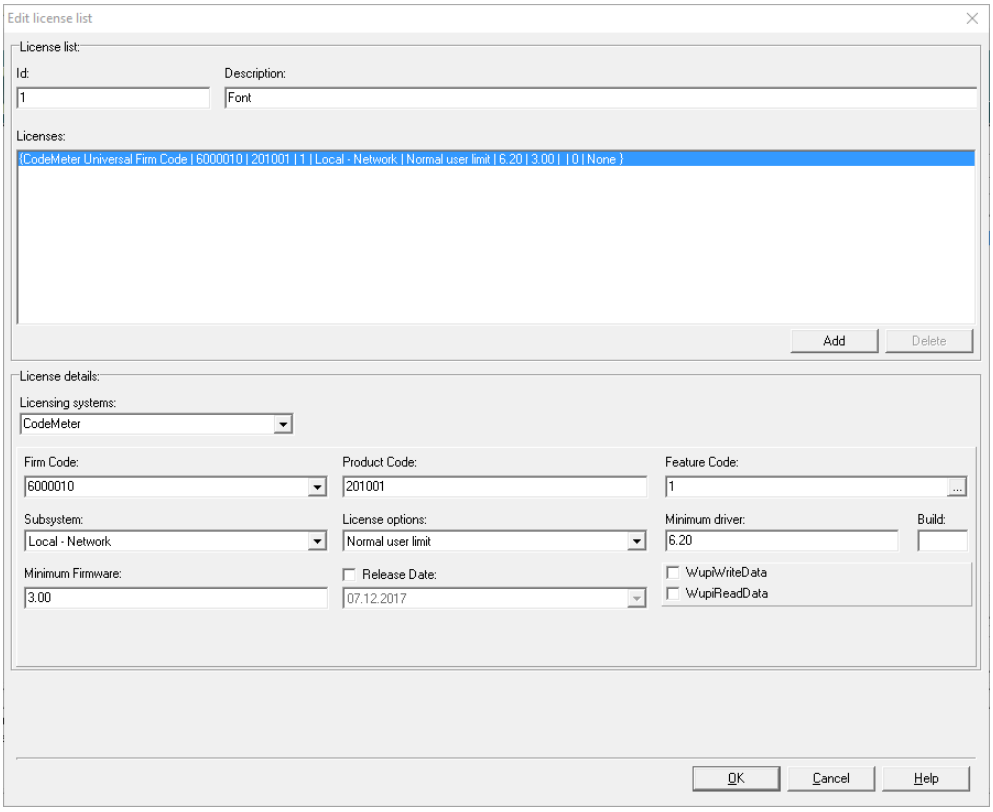


Figure 210: *IxProtector - License List Entry*

The "Second Sample" example prompts you with an index-based placeholder of ID=`1` for the Change Font License. Transfer the necessary data of the overview table 294, i.e. ***Firm Code*** `6000010` and ***Product Code*** `201001` with a ***Feature Code*** of `0` in the Feature Map.

The "ID" column now holds the index-based placeholders which will be addressed by the WUPI license calls.

4. Navigate to the **"IxProtector"** input window to display the function list.

The *IxProtector* options define the functions to be protected and allow for the assignment of functions to the license list entries you defined above.

5. Click the **"Edit"** button.

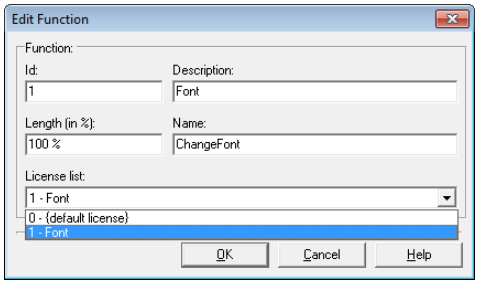


Figure 211: *IxProtector* - Function List Entry

In the "Secind Sample" example, *IxProtector* prompts you with an index-based placeholder of ID=`1` for the Change Font function. Transfer the necessary function name from the overview table[294] above.

> The description of the function in field **"Name"** must exactly match the name which is later addressed in the source code by the index-based placeholder. Overloaded functions are not supported.

Specify the length of the array to be encrypted for the function.

You enter the length, in percent, anywhere from 0 to 100%. If you want this number to represent percentage, you must enter the percent character (%). Alternatively, you are able to specify the length by number of bytes. Then *AxProtector* automatically calculates the length.

Then select the license list to which the function is to be assigned.

Now all required data has been completed in *IxProtector*, and all index-based placeholders are defined.

8.3.1.3 Programming the CmContainer

After protecting the "Second Sample" application using *IxProtector*, you now have to transfer the license entries into the *CmContainer*. For this either use *CodeMeter License Editor*[306], *CmBoxPgm*[315] or *CodeMeter License Central*[337].

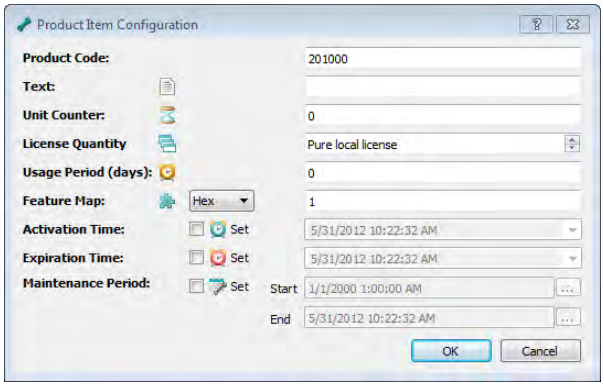
The programming covers:

- a *Product Item* with a *Product Code* `201000` for the license container with the Test *Firm Code* `6000010` and a *Feature Code* value of `1` for the *Feature Map.*
- a *Product Item* with a *Product Code* `201001` for the license container with the Test *Firm Code* `6000010` and a *Feature Code* value of `1` for the *Feature Map.*

CodeMeter License Editor

In *CodeMeter License Editor*, please proceed with the following steps:

1. Select the license container level of the Test *Firm Codes* `6000010` and create the *Product Item* with a *Product Code* `201000` or `201001` respectively using the **"Add"** control either by the respective button, or the context menu.



2. Complete the **Product Code**, **Text**, **Unit Counter**, and **Feature Map** fields according to the specifications.
3. Click the **"Execute"** button to program this license entry into the connected *CmDongle*.

CmBoxPgm

In *CmBoxPgm* proceed with the following steps:

1. Create a *Product Item* with *Product Code* `201000` in license container with the Test *Firm Code* `6000010` and a *Feature Code* value of `1` for the *Feature Map*.

```
CmBoxPgm.exe /f6000010 /p201000 /pfm1 /ca
```

2. Create a *Product Item* with *Product Code* `201001` in license container with the Test *Firm Code* `6000010` and a *Feature Code* value of `1` for the *Feature Map*.

```
CmBoxPgm.exe /f6000010 /p201001 /pfm1 /ca
```

8.3.1.4 Integration into the Source Code

Subsequently, you insert the WUPI functions into the source code where the software protection mechanism or license queries have to be applied.

The WUPI functions now will refer to the index-based placeholders you created before in *IxProtector* by completing the license and function lists

> When developing, you first have to integrate a Dummy-DLL which holds the WUPI function calls.
> Depending on the operating system (Windows 32- or 64-bit), use the `WupiEngine32.dll` or `WupiEngine64.dll`. When protecting .Net applications, use the `WupiEngineNet.dll`.
> These files are located in the directory "`%Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\lib`".

In the following, some source code samples show how some of the WUPI functions have been implemented for the "Second Sample" example.

> The source code file for "Second Sample" in the programming language C++ (and also for the other languages) you can find in the directory "`%\Users\Public\Documents\WIBU-SYSTEMS\Software Protection`". All following samples are taken from this implementation file.

WupiQueryInfo

In the file `SampleNotePad.cpp` of the Second Sample (C++), for example, ***WupiQueryInfo*** is called at start of the application.

```
CTextEditApp::CTextEditApp()
{

        // Construction code, initialization in InitInstance
  // Checks if the software is encrypted
  if (WupiQueryInfo(0, WupiQIFirmCode) == 0)
  {
    MessageBox(NULL, TEXT("Software is not encrypted correctly! \nDon't ship this version."),
               TEXT("SampleNotePad - INTERNAL version"), MB_ICONERROR);
  }//if

}//CTextEditApp()
```

WupiEncryptCode and ***WupiDecryptCode***

In the file `CMainFrm.cpp` of the Second Sample (C++) ***WupiDecryptCode*** and ***WupiEncryptCode*** are called on calling `OnViewFont()`.

```
/// <summary>
/// Checks the license for the Font module and calls the Font Dialog.
/// </summary>
void CMainFrame::OnViewFont() //Menu option "Font" from "View"
{
        int iWupiResult;
    if (WupiDecryptCode(1) == 1)
        {
      ChangeFont();
      iWupiResult = WupiEncryptCode(1);
    }
    else
    {
      MessageBox("This module is not activated!", "License Error", MB_ICONERROR);
    }//elseif

}//OnViewFont()
```

8.3.1.5 Encryption using AxProtector

After compiling "Second Sample" you encrypt using *AxProtector* with *IxProtector* simultaneously activated. *IxProtector* now replaces the placeholders by entries of the license or function list.

> *IxProtector* is integral part of *AxProtector.* You can alternatively use it alone by choosing an "IxProtector" project type or additionally in combination with *AxProtector*. When integrating *IxProtector* in *AxProtector, IxProtector* searches the respective source code parts and encrypts them before *AxProtector* encrypts the completed application.
> But remember when using a "IxProtector" project type, a higher security level is provided, since the Dummy-DLL is replaced by static code. This DLL is not used later when the application is executed.

8.4 The CodeMeter Core API

With the *CodeMeter Core API* Wibu-Systems presents a powerful interface to communicate with *CmContainer* at the runtime of *CodeMeter License Server*. After all, all other APIs and protection mechanisms (*AxProtector*, *IxProtector*, *Software Protection API* WUPI) base on *Core API* Functions. This accounts for the supplemental use of this interface complementing the other protection options provided by *AxProtector* and *IxProtector*. The transitions between the protection levels are smooth and seamingless.

One Entry - several Protection Layers

The license entry *Software Protection API* WUPI is using at runtime is allocated by *AxProtector*. With the WUPI function ***WupiGetHandle*** [291] within *IxProtector* you are able to read out this entry and further use it in *CodeMeter Core API*.

Application Scenarios

Additional application areas comprise:

- Reading out further data from the *CmContainer*, z.B. e.g. display and transfer of user-specific license information (COLI) when a support request is triggered (***CmGetInfo*** via ***WupiGetHandle***).
- Encryption and decryption of data of any kind within applications, e.g. encryption using ***CmCrypt*** or ***CmCrypt2*** including different security features (*Encryption Code Options*) for variable data within an application. Then sensible data for separate customers are differently encrypted.
- Using a *CmDongle* for authentication, e.g. for signing of data and the verification that data transferred by separate users has been actually send by these users.
- Updating of license information by the creation of context files (***CmSetRemoteContext***) and their update (***CmSetRemoteUpdate***). This allows for obtaining pay-per-use information.

These are only some of the additional options *CodeMeter Core API* provides. For further questions and inquiries contact Wibu-Systems customer support.

8.4.1 Functional Areas

The functions of *CodeMeter Core API* combine several areas. The predominant part of the functions you will also find in *CodeMeter API Guide* [300]. The functions here are outlined only briefly. For a detailed description of functions, used syntax and parameters see *CodeMeter Core API Help* (accessible as context online help (F1) in *CodeMeter API Guide* or by the **"Start | All Programs | CodeMeter | Documentation"** start menu item (Press "Windows" key to open Start screen | Type "CodeMeter API Guide" | Press "Enter" key)).

8.4.1.1 Access API

This API covers all functions to access a *CmContainer*.

Command	Description
CmAccess	accesses a subsystem, a *CmContainer*, a *Firm Item*, or product entry (*Product Item*) in a given subsystem.
CmAccess2	executes an access as ***CmAccess*** but provides extended functions (available since *CodeMeter* Version 3.30). Use ***CmAccess2*** to profit from the extended functional scope.
CmRelease	closes a handle opened by ***CmAccess*** or ***CmAccess2*** including all related subsystem accesses. Never use ***CmRelease*** on entries you addressed with ***WupiGetHandle*** within WUPI.

8.4.1.2 Authentication API

This API covers all functions to execute authentication operations.

Command	Description
CmCalculateDigest	calculates a 32 bytes hash value of an entered input sequence for the use in an authentication operation. The algorithm SHA-256 is applied.
CmCalculate Signature	calculates an ECDSA (Elliptic Curve Digital Signature Algorithm) signature with the specified hash value in the *CmContainer*.
CmGetPublicKey	reads the public key from a *CmContainer*.
CmValidateSignature	validates a ECDSA (Elliptic Curve Digital Signature Algorithm) signature with the specified public key.

8.4.1.3 Encryption API

This API covers all functions required for encryption and decryption operations of data.

Command	Description
CmCrypt, CmCrypt2	encrypts or decrypts data directly or indirectly using a *CmContainer*.
CmCryptEcies	encrypts a specified byte sequence with the ECIES (Elliptic Curve Integrated Encryption Scheme) algorithm.
CmCrypt Sim	encrypts or decrypts data directly or indirectly using the *Firm Security Box* entry of the desired *Firm Code*.
CmCalucalatePioCoreKey	calculates the core key for the encryption of the PIO *Hidden Data*. This operation requires a *Firm Security Box*.
CmGetSecureData	reads encrypted *Hidden Data* from the *CmContainer* using the *Product Item Option Enryption Key* (PIOEK).
CmDecryptPioData	decrypts a *Hidden Data* sequence read using the *Product Item Option Enryption Key* (PIODK).
CmGetPioDataKey	calculates the key required to decrypt *Hidden Data*.

8.4.1.4 Error Management API

This API covers functions required for handling error messages.

Command	Description
CmConvertString	converts the input in a specified string.
CmGetLastErrorCode	queries the last error code.
CmGetLastErrorText	queries the last error text.
CmGetLastErrorText2	queries the last error text as ***CmGetLastErrorText*** but provides extended functions.
CmSetLastErrorText	sets an error code in a internally used global error code variable.

8.4.1.5 Management API

This API covers all functions required for *CodeMeter* event-related operations.

Command	Description
CmCheckEvents	waits until a (local) event occurs, and returns the results.
CmGetBoxes	identifies all connected *CmContainer* which are connected to the same connection.
CmGetBoxContents	reads all entries of a *CmContainer*.
CmGetInfo	queries data in the *CmContainer*. Differently used query parameters result in different results.
CmGetServers	searches the local network for running *CodeMeter License Server* to which a *CmContainer* is connected.
CmGetVersion	calculates the version of the related *CodeMeter* module.

8.4.1.6 Programming API

This API covers functions required to program *CmContainer*.

Meanwhile, these functions have been replaced by the *Programming API* [*High Level Application Programming Interface* (HIP)]. Using the functions listed below is limited to rare cases.

Command	Description
CmReserveFirmItem	reserves a temporary *Firm Item* in a *CmContainer* for subsequent *Firm Item* and *Product Item* operations.
CmCreateProductItemOption	prepares a security sequence for adding or updating of a *Product Item Option*.
CmCreateSequence	calculates a signature to program a *CmContainer* entry.
CmProgram	programs different entries into a *CmContainer*.
CmValidateEntry	checks a specified sequence.

8.4.1.7 Remote Update API

This API covers all functions required for the remote programming of license request Context and Update Files (`*.WibuCmRaC` / `*.WibuCmRaU` files).

Command	Description
CmGetRemoteContext	saves the contents of a *CmContainer* in an encrypted and compressed Context File (license request) (`*.WibuCmRaC`).
CmSetRemoteContext2	saves contents as ***CmGetRemoteContext*** but has an extended functional scope.
CmSetRemoteUpdate	programs a *CmContainer* with the specified remote activation Update File (license update) (`*.WibuCmRaU`). This file holds all information to be program into a *CmContainer*.
CmSetRemoteUpdate2	programs a *CmContainer* as ***CmSetRemoteUpdate*** but provides extended functions.
CmListRemoteUpdate	analyzes a remote activation Update File (license update) (`*.WibuCmRaU`), and defines the serial numbers of all *CmContainer* referenced in the file.
CmListRemoteUpdate2	analyzes a remote activation Update File as ***CmListRemoteUpdate*** but provides extended functions.

The extended functions holding an suffix of 2 allow, for example, using buffer instead of file operations, or using encoding options for transferred file names.

8.4.1.8 Time Management API

This API covers the function required to use the certified time (for the synchronization scheme of different time in a *CmContainer* see here[357]).

Command	Description
CmSetCertifiedTimeUpdate	gets the current certified time and date stamp from the time server (Certified Time Creation Server, CTCS) and saves it into the *CmContainer*.

8.4.1.9 License Transfer API

This API covers the function required for the license transfer (see here[43]).

Command	Description
CmLtCreateContext	first step of a license transfer.
CmLtDoTransfer	actual license transfer. A Context File (`*.WibuCmRaC`) is processed and an Update File (`*.WibuCmRaU`) generated.
CmLtImportUpdate	imports the update data into the license transfer target.
CmLtCreateReceipt	generates and returns a signature of the *Firm Item* data.
CmLtConfirmTransfer	checks the receipt and completes the transfer.
CmLtCleanup	cleans up the no longer required license data.
CmLtLiveTransfer	performs the transfer in one go.

8.4.2 CodeMeter API Guide

CodeMeter API Guide represents an interactive program 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#, VB6, VB.Net, Delphi and Java are supported.

The generated source code fragments you easily transfer into the source code of an application by using the clipboard.

8.4.2.1 Structure and Navigation

You access *CodeMeter API Guide* using *CodeMeter Start Center*[48] or alternatively using the **"Start | All Programs | CodeMeter | Tools"** start menu item (Press "Windows" key to open Start screen | Type "CodeMeter API Guide" | Press "Enter" key).

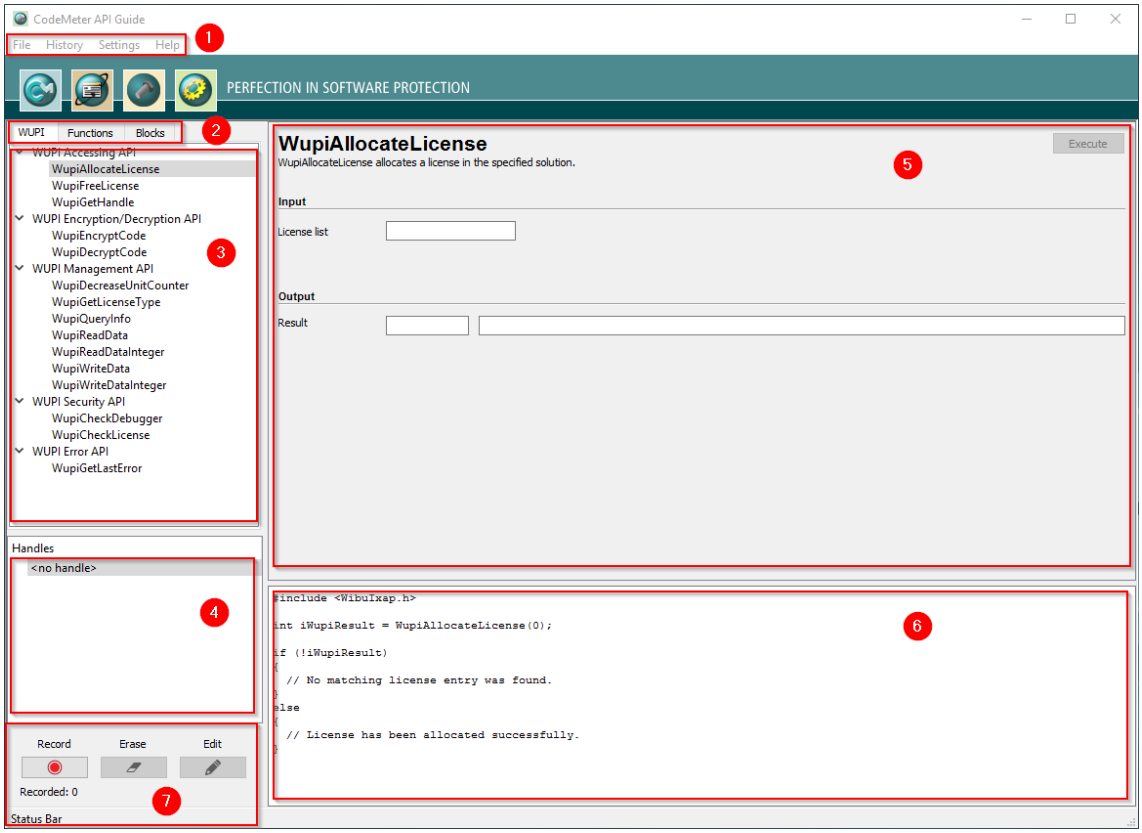


Figure 212: *CodeMeter API Guide* - Start GUI

The *CodeMeter API Guide* user interface consists of six separate areas:

- Menu Bar[301] (**1**)
- Tabs[301] to switch between WUPI, *Core API*, and Blocks (**2**)

- Tree view 302 Function Calls (**3**)
- Handle Display 302 Window (**4**)
- Interactive Area 302: Input and Output field (**5**)
- Source Code 302 Area (**6**)
- Record 302 Area (**7**)

8.4.2.2 Menu Bar

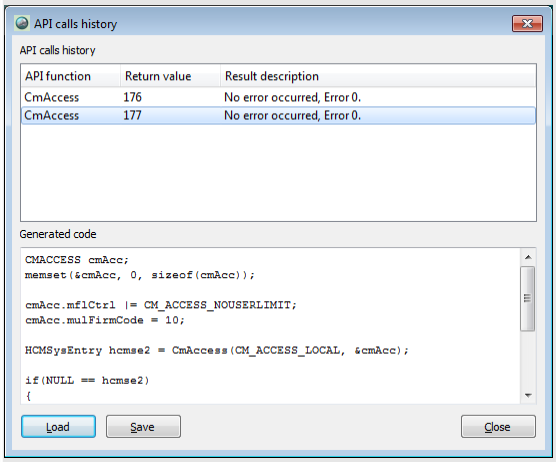
File Menu

Element	Description
Export Code	Select this menu item to save the generated code into a separate file.
Exit	Select this menu item to close *CodeMeter API Guide*.

History Menu

CodeMeter API Guide provides the option to save the history of your API calls for reusing purposes.

> The key combination `<CTRL><H>` opens the history window anytime.

Element	Description
Load	Loads the `*.WibuCmAPI`file including generated source code into the history window.
Save	Saves the history of API calls in a `*.WibuCmAPI` file you are free to name and save at a desired location.
Show	Shows the history of your API calls including the generated source code in the history window.  Figure 213: *CodeMeter API Guide* History

Settings Menu

Element	Description
UI Language	Select this menu item to set the language display of the user interface. The provided languages comprise German, English and Chinese.
Programming Language	Select this menu item to select the programming language of your software project. The provided programming languages comprise C++, C, C#, VB.NET, VB6, Java, and Delphi.

Help Menu

Element	Description
Context Help F1	Select this menu item to open the context sensitive *CodeMeter API Guide* online help. You also obtain information on the selected commands by pressing the `[F1]` key.
Info	Select this menu item to open a separate window holding *CodeMeter API Guide* version information.

8.4.2.3 Tabs

CodeMeter API Guide provides you the area **"Tab"** allowing you to switch between API calls for WUPI, *Core API*, and full function blocks.

Element	Description
WUPI	The functions of the *Software Protection API* or WUPI (WIBU Universal Protection Interface) are clearly arranged by single functional areas.
Functions	In this tab you find the predominant part of the *CodeMeter Core API* functions.
Blocks	Next to single API functions, *CodeMeter API Guide* also provides you full functions blocks. These function blocks comprise the reading and writing of data from and into *CmContainer*, the execution of different encryption operations, and the activation of the *CmStick* LEDs.

Element	Description	
	Block	Description
	Read Data	Reading *Product Item Options Text*, *User Data*, *Protected Data* and *Protected Data*.
	Write Data	Writing of data into a *Product Item*. Only operations are supported which do not require a *Firm Security Box* (FSB). Most of the write operations are limited to the license container with the *Firm Code* 0.
	Sign Message	Use of ECDSA (Elliptic Curve Digital Signature Algorithm) to sign data.
	Verify Signature	Verification of signed data.
	Symmetric Encryption	Symmetric encryption and decryption of binary data.
	Asymmetric Encryption	Asymmetric decryption of binary data.
	Asymmetric Decryption	Asymmetric decryption of binary data in the *CmContainer*.

8.4.2.4 Tree View

CodeMeter API Guide provides you a controllable tree view for a clear and structured display of single API calls. Depending on the tab you select, the calls are topically structured by areas. The single root nodes you can easily collapse and expand by using the ⊟ and ⊞ controls.

8.4.2.5 Handle Display Window

In this area *CodeMeter API Guide* shows you existing handles. A handle identifies and refers to a specific object, i.e. an entry in the communication process between the *CmContainer* and the *Core API* interface. Objects with a reference to an entry comprise *Product Items*, *Firm Items*, *CmContainer* or subsystems.

Then the call you execute by selecting an API function relates to the handle displayed or selected.

8.4.2.6 Interactive Area

The interactive input area allows you to enter parameters and structures for previously selected API functions. In some cases, additional windows and dialogs open for more specific input. The input is transferred into the source code area.

Click the **"Execute"** button to start the function call. Then the output area shows you the results of the function calls, e.g. whether an error occurred or not, or the protection result.

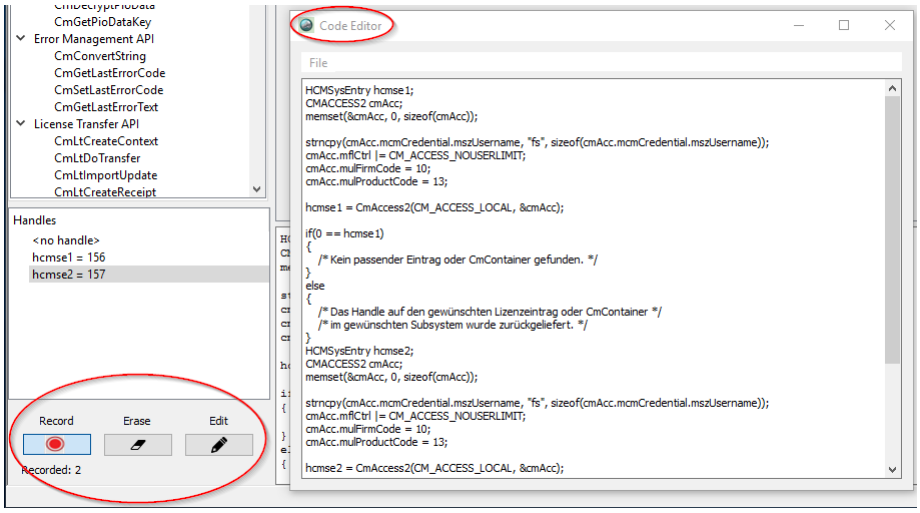
8.4.2.7 Source Code Area

The source code area automatically adapts to the specification you selected in the interactive input area. Now you can select the source code and paste it into your own software project.

> Alternatively, you can export the adapted source code in a separate file using the **"File | Export Code"** menu item or save the history of function calls as file using the **"History | Save"** menu item.

8.4.2.8 Record Area

The record area provides the option to record source code fragments generated using handles and actions in *CodeMeter API Guide*. A click on the "**Record**" button starts recording and each time after a new action the field "**Recorded**" is incremented. A click on the "**Edit** button opens the Code Editor including the recorded events. The result may be further processed. Using the "**Erase**" button clears the Code Editor.



%FIGURE%> 214: *CodeMeter API Guide* Record Area

8.4.3 Sample Applications: CmDemo, CmCalculator, WupiCalculator

The *CodeMeter Development Kit* ships with example applications for different programming languages (C++, C#, VB6, VB.NET, Delphi and Java). The applications are intended to ease introduction, and help you getting familiar with *CodeMeter* functions.

> You find the examples "CmDemo" and "CmCalculator" after installing the *CodeMeter* SDK for respective programming languages in the directory "`%Users%\Public\Documents\WIBU-SYSTEMS`"
> The example "WupiCaculator" you find for respective programming languages in the directory "`%\Program Files%\WIBU SYSTEMS\AxProtector\DevKit\Samples\IxProtector\…\WupiCalculatorIndex`".
>
> Alternatively, find the samples using the start menu item **"Start | All Programs | CodeMeter | Samples"** or via *CodeMeter Start Center* 49 (Press "Windows" key to open Start screen | Type "CodeMeter Samples" | Press "Enter" key).

8.4.3.1 CmDemo

The example application "CmDemo" represents a project showing the implementation of the most frequently used *Core API* functions. By default, after installing you find the file `CmDemo.exe` in the specified directory. The functions including the source code you find in the same directory in form of respective programming files for related programming languages.

The example in C++ is also available in a commandline version with project data for macOS or Makefile for Linux.

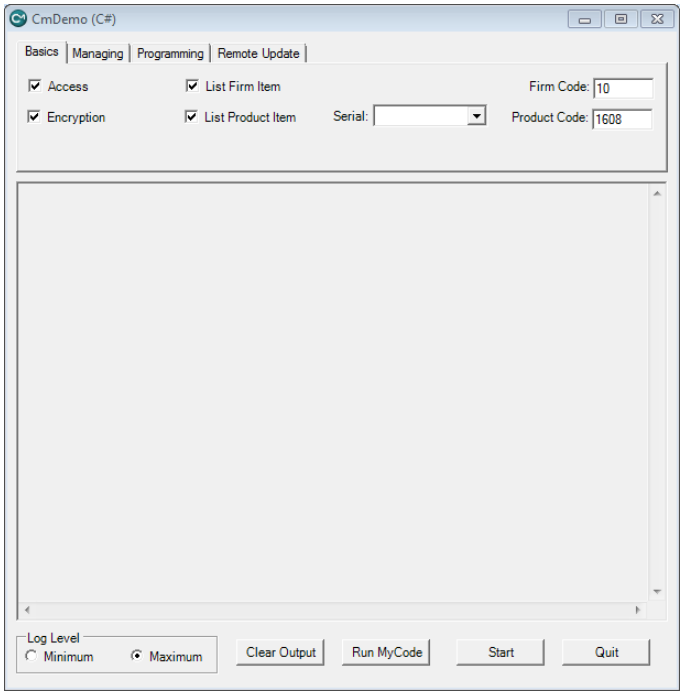


Figure 215: *CmDemo* - Overview

For a clear overview the API functions are topically structured and summarized in separate tabs.

Element	Description
Basics	This page shows the access to entries, the reading of entries, and the encryption characteristics of *CodeMeter*. This section holds the source code you require for most of the *CodeMeter* implementations.
Managing	This page demonstrates the complete read-out of a *CmContainer* and the querying of internal information of the *CmContainer*, e.g. version of *CodeMeter License Servers* or version of the hardware. In addition, the access to the LEDs, *CodeMeter* on the network, and exception handling are displayed.
Programming	This page shows how to program and delete the different entry types. The source of this area can be used for programming own applications for *CmContainer*. Note that you require a connected *Firm Security Box*. Meanwhile, these functions have been replaced by the *Programming API* [*High Level Application Programming Interface* (HIP)].
Remote Update	This page demonstrates the *CodeMeter Field Activation Service* (CmFAS), i.e. the remote programming of *CmContainer* without having to re-send altered *CmContainer*.

Additional buttons comprise:

Element	Description
Log Level	Click the "**Minimum**" or "**Maximum**" checkboxes to set the log level for the display window.
Run MyCode	Click the "**Run MyCode**" button to start the related event including the source code which is part of the separate secondary function "MyCode".

Element	Description
	Insert self programmed code, or code copied from other parts of "CmDemo" into the function "`MyCode`". After successful compilation you are able to test it using the interface
Start	Click this button to start the functionalities you specified in the selected tab. Depending on the log level you set, the information is displayed in the window.
Quit	Click this button to close "CmDemo".
Clear Output	Click this button to delete the content of the display window.

8.4.3.2 CmCalculator

The example application "CmCalculator" represents a project showing the use of some essential *CodeMeter Core API* functions and structures on the basis of a simple calculator example.

8.4.3.3 WupiCalculator

The example application "WupiCalculator" shows how to implement modular software protection in combination with a pay-per-use license model using WUPI. See here 293.

9 Programming of CmContainer and Licensing Management

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

> As a matter of fact, in *CodeMeter* it does not make a difference which step for mapping your license strategy you take first. Whether you already map your license models when encrypting using *AxProtector* or *IxProtector* and then program your *CmContainer* or whether you first program license information into the *CmContainer* and then later encrypt using *AxProtector* or *IxProtector* - both options work.
> In the case of *CodeMeter Core API*, you also have this option by using the necessary "handle" not at the runtime of the application, but instead using the WUPI function ***WupiGetHandle*** within *IxProtector* reading out the entry, and further using it for *Core API* functions.

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* 337.

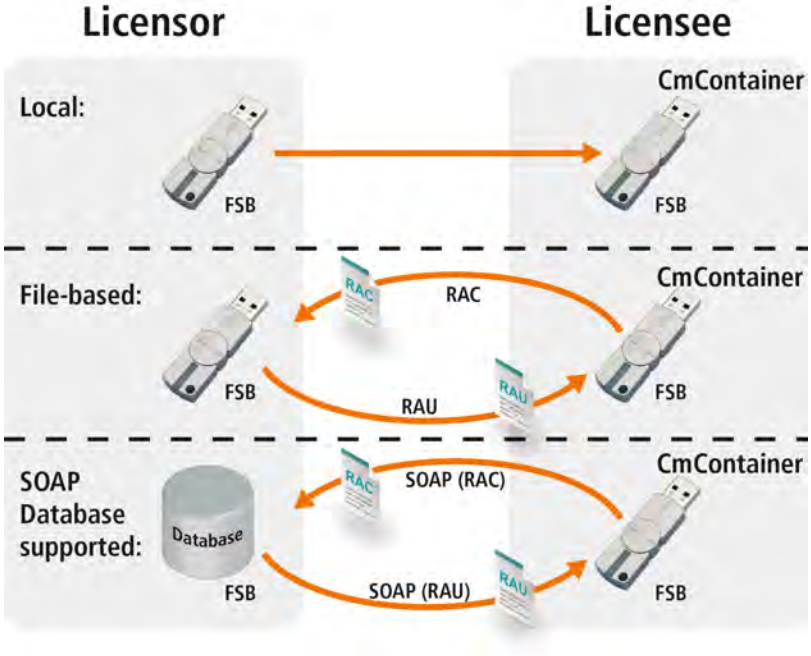


Figure 216: *CmContainer* Programing Options

For all three methods *CodeMeter* provides several tools:

- ***CmBoxPgm*** 315: commandline tool for batch programming of *CmContainer* in production.
- ***CodeMeter License Editor*** 306: graphic tool to program *CmDongles* or local tests of licensing strategies.
- ***CodeMeter License Central*** 337: 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 342, 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).

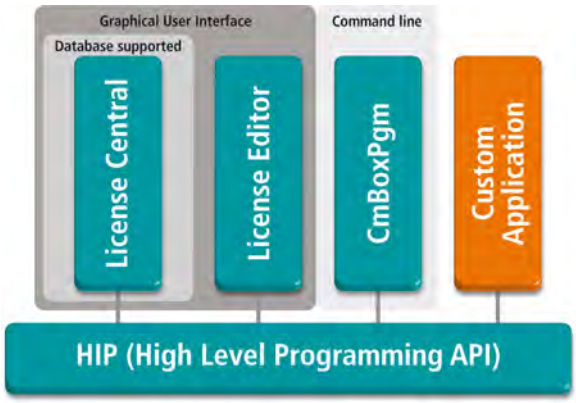


Figure 217: Tools for *CmContainer* Programming

9.1 CodeMeter License Editor

CodeMeter License Editor is an application which allows you to create, edit or delete licenses and their license components (***Firm Item***, ***Product Item***, and ***Product Item Options***) in a *CmDongle*. Next to programming of locally connected *CmDongles*, *CodeMeter License Editor* also supports file-based remote programming[342] (*CodeMeter Field Activation Service*, *CmFAS*).

> Please use *CodeMeter License Editor*, if only a small number of *CmDongles* are used, e.g. while developing or while testing license strategies.

You access *CodeMeter License Editor* either by using *CodeMeter Start Center*[48] or by using the **"Start | All Programs | CodeMeter | Tools"** start menu item (Click Windows key to open start screen | Type "CodeMeter License Editor" | Press Enter key).

> Currently, not all functions of *CodeMeter License Editor* are available.
>
> For the direct programming mode the following holds:
>
> *CmDongle* and *CmActLicense* licenses can be programmed with the exception of:
>
> - license transfer options (*CmBoxPgm* options `-plt`...)
> - ***Module Items***
> - ***Named User***
> - ***Access Password***
> - ***Maximum Encryption Rate***
>
> > If you want, however, still use *CodeMeter License Editor* to program, Wibu-Systems recommends first to create these currently not supported license options (***Product Item Options***) using *CmBoxPgm*.
>
> For the file-based remote programming[342] the following file types are supported:
>
> - Context Files (`*.WibuCmRaC`) and Modified Context Files (`*.WibuRaM`) for *CmDongle* and *CmActLicense* (except of empty *CmDongles*, i.e. no information on the ***Firm Code*** is available) (***Firm Codes*** bigger than 600000).
> - Context Files (`*.WibuCmRaC`) and Modified Context Files (`*.WibuRaM`) (***Firm Codes*** smaller than 600000)
> - *CmActLicense* Context Files (`*.WibuCmRaC`) in read only mode (***Firm Codes*** smaller than 600000).

9.1.1 Structure and Navigation

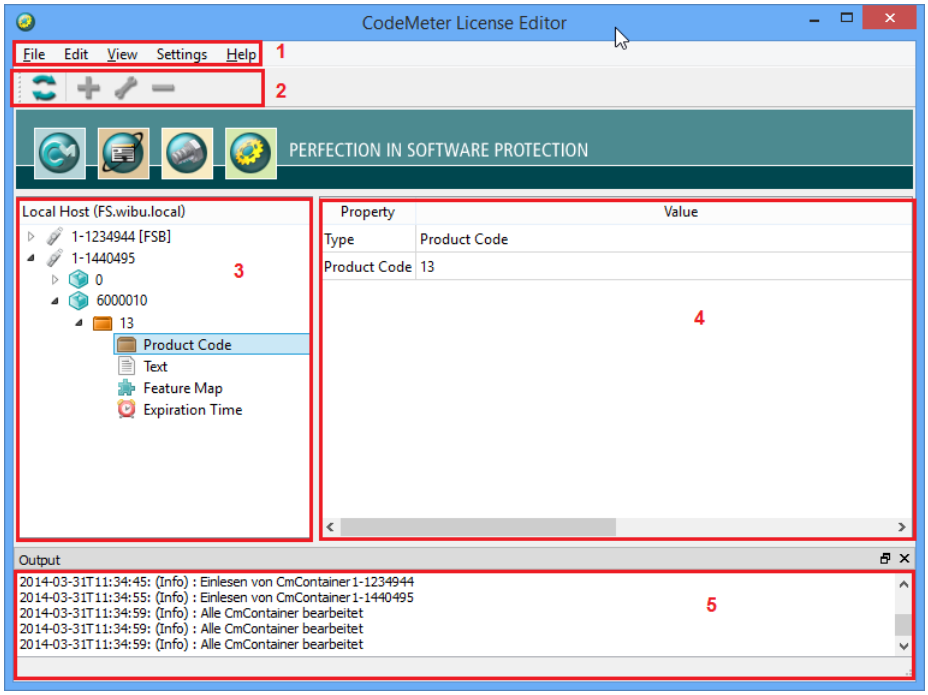


Figure 218: *CodeMeter License Editor* - Start Screen

The *CodeMeter License Editor* graphical user interface consists of five separate areas:

- Menu Bar 307 (**1**)
- Symbol Bar 308 (**2**)
- Tree View Window 308 (**3**)
- Display Window 309 (**4**)
- Output Window 309 (**5**)

9.1.1.1 Menu Bar

File Menu

Element	Description
Direct Programming Mode	Loads license information in *CmDongles* to *CodeMeter License Editor*. This menu item corresponds to the command "Execute".
Open Context File...	Loads license information in *CmDongles* to *CodeMeter License Editor* by using Context Files (`*.WibuCmRaC`) or Modified Context Files (`*.WibuCmRaM`).
Close Context File...	Closes license information in *CmDongles* to *CodeMeter License Editor* by using Context Files (`*.WibuCmRaC`) or Modified Context Files (`*.WibuCmRaM`).
Recent Files	Displays file list recently used in *CodeMeter License Editor*.
Exit	Closes *CodeMeter License Editor*. In order to exit *CodeMeter License Editor* using the keyboard, press the `<ALT+F4>` key combination. Alternatively, you may also close the window using the ✕ control. Before exiting you are prompted to save the changes you have made.

Edit Menu

Element	Description
Add Item	Adds a new *Item*. In order to add an *Item* using the keyboard click the `<CTRL+A>` key combination.
Modify Item	Opens a dialog to modify an *Item*. In order to modify an *Item* using the keyboard click the `<CTRL+M>` key combination.
Delete Item	Deletes an *Item*. In order to delete an *Item* using the keyboard click `<CTRL+D>` the key combination.
Execute	Saves changes of the licenses in the *CmDongle*. In order to save changes using the keyboard click the `<CTRL+X>` key combination.
Refresh	Refreshes the view of the licenses in a *CmDongle*. In order to refresh the *Item* view using the keyboard click the `<CTRL+R>` key combination.

View Menu

Element	Description
Output	Allows you to reactivate the output window you deactivated by using the ✕ control.

Element	Description
Tools	Hides or shows tool bar of *CodeMeter License Editor*.

Settings Menu

Element	Description
Language	Allows to set language of *CodeMeter License Editor* graphical user interface.

Help Menu

Element	Description
About	Selecting this menu item opens a window informing about the *CodeMeter License Editor* version you use.

9.1.1.2 Symbol Bar

Figure 219: *CodeMeter License Editor* - Symbol Bar

The symbol bar is freely to move and consists of a set of shortcut symbols allowing for standard functions. Please click on a symbol to perform the function.

9.1.1.3 Tree View

This windows displays the contents of the *CmDongles* connected to your computer.

Using the ◢▷ controls collapses or expands the root nodes of single *CmDongles*, *Firm Items* levels, and license entries (*Product Item Options*).

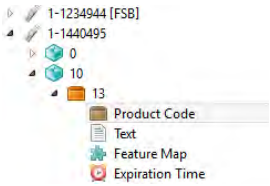


Figure 220: *CodeMeter License Editor* - Tree View

The following figure shows an overview of symbols used and their meaning.

Symbol	Object
	CmDongle
	Firm Item
	Product Item (license entries)
	Product Item Options
	Text
	Unit Counter
	License Quantity
	Usage Period
	Feature Map
	Activation Time
	Expiration Time
	Maintenance Period
	Hidden Data
	Secret Data
	Protected Data
	Extented Protected Data
	User Data

Table 7: *CodeMeter License Editor* - Entry Symbols

9.1.1.4 Display Window

The display window shows you details on objects (*CmDongle*, *Firm Item*, *Product Item*).

Property	Value
Product Code	54321
Product Item Reference	37
Unit Counter	1.000
Usage Period	518400 seconds (= 6 days, 0 hours, 0 minutes, 0 seconds) - not activa
License Quantity	5

Figure 221: *CodeMeter License Editor* - Display Window

9.1.1.5 Output Window

The output window informs you on actions executed in *CodeMeter License Editor* and issues error messages if required.

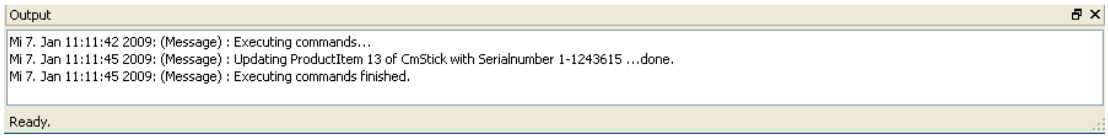


Figure 222: *CodeMeter License Editor* - Output Window

Using the control allows you to move the output window to a favored place on the desktop. This may increase clarity. Using the control you are also able to deactivate the output window. You reactivate it by using the file menu item **"View | Output"**.

9.1.2 Working with CodeMeter License Editor

The following section shows you how to work with *CodeMeter License Editor*.

9.1.2.1 Starting CodeMeter License Editor

You access *CodeMeter License Editor* either by *CodeMeter Start Center* 48 or by the start menu **"Start | All Programs | CodeMeter | Tools"** (Click Windows key to open start screen | Type "CodeMeter License Editor" | Press Enter key).

9.1.2.2 Display of connected CmDongles

For the display of contents in connected *CmDongles* you have two options. You read in license details from *CmDongles* either by the function **Refresh** or using the menu item **File | Open Context File...** you load a Context File (`*.WibuCmRaC`) or Modified Context File (`*.WibuCmRaM`). which holds the license details.

9.1.2.2.1 Refreshing Display

Using the Edit Menu item **"Edit | Refresh"** or the symbol you re-read the license details of all *CmDongles* connected to your computer.

9.1.2.2.2 Open Context Files

For the file-based remote programming 342 only the following file types are supported:

- Context Files of *CmDongles* (`*.WibuCmRaC`)
- Modified Context Files of *CmDongles* (`*.WibuCmRaU`)
- Context Files of activated *CmActLicenses* (`*.WibuCmRaC`)

Using the File Menu item **"File | Open Context File..."** you load the respective Context File (`*.WibuCmRaC`) or Modified Context File (`*.WibuCmRaM`) holding the license details for further editing.

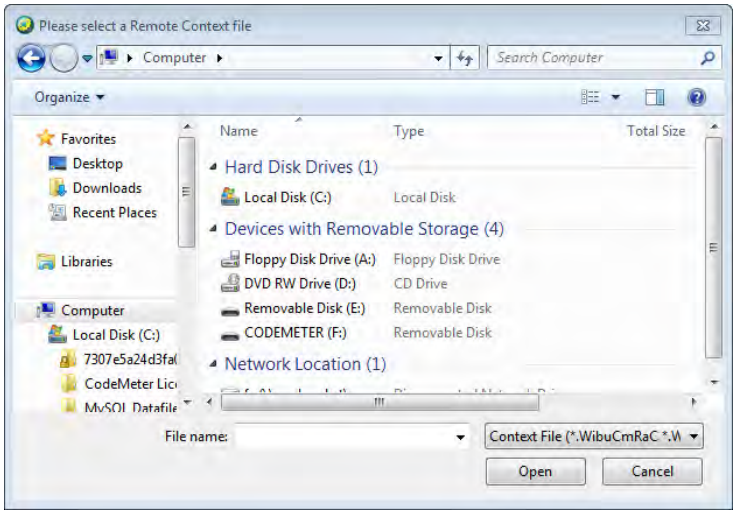


Figure 223: *CodeMeter License Editor* - Context File

9.1.2.3 Creating, Editing and Deleting a Firm Code

Creating and editing

For creating and editing a *Firm Item* (*Firm Code*), please proceed as follows:

1. Select the desired *CmDongle*.
2. Select the item **"Add Item"** or **"Modify Item"** via:
   - or symbol in the context menu (right mouse-click) or in the symbol bar
   - the **Edit** menu item of the same name.

   The following dialog allows you to enter a new or specify data for an already existing *Firm Item* (*Firm Code*).

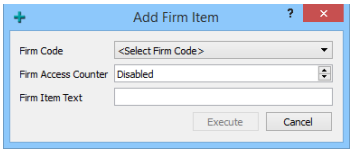


Figure 224: *CodeMeter License Editor* - Create Firm Item and Editing

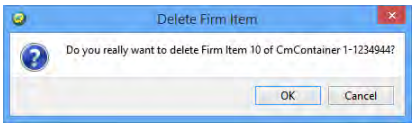
Element	Description
Firm Code	Specify a *Firm Code* or select on from a list of available *Firm Codes*.
Firm Access Counter	Specify a numeric value for the *Firm Item*. The *Firm Access Counter* locates at the *Firm Item* level of a *CmContainer*. This counter allows you to control whether a *Firm Item* can be used for encryption and decryption operations. By default, the FAC is deactivated and has a value of `65535` (`0xFFFF`). A software vendor is able to program it to any other value between `1` and `65534`. On detecting a debugger the FAC is decremented by a value of `1`. If the FAC reaches a value of `0`, the *Firm Item* is locked [318] . The owner / end-user of the locked *Firm Items* must contact the software vendor for unlocking codes. This can be done by remote programming.
Firm Item Text	Specify the text which describes the *Firm Item* in greater detail.

Deleting

For deleting a *Firm Item*, please proceed as follows:

1. Select the desired *CmDongle*.
2. Select the item **"Delete Item"** via:
   - symbol in the context menu (right mouse-click) or in the symbol bar
   - the **Delete** menu item of the same name.

   The following dialog asks to confirm deleting the object.



Depending on the license type of your *Firm Code* it may be the case that you cannot delete a *Firm Item* using your FSB. This forecloses an accidental deletion. However, this option may be featured any time later on request and free of charge.

9.1.2.4 Creating, Editing and Deleting a Product Code

Creating and Editing

For creating and editing a *Product Code*, please proceed as follows:

1. Select the desired *Firm Item* level.
2. Select the item **"Add Item"** or **"Modify Item"** via:
   - or symbol in the context menu (right mouse-click) or in the symbol bar
   - **Edit** menu item of the same name.

   The following dialog allows to create a new or edit an already existing *Product Code*.

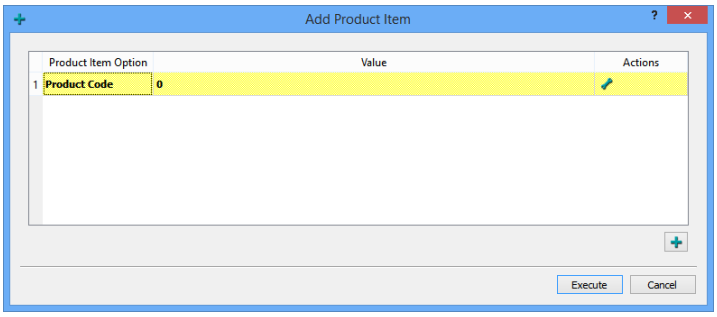
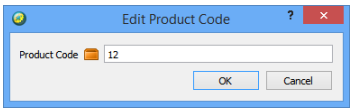


Figure 225: *CodeMeter License Editor – Product Code* create or edit

3. Click the symbol.

   An input dialog displays:



4. Enter desired *Product Code*.
   The specified *Product Code* is transferred into the *Product Item* dialog.
5. Click the "**Execute**" button.

   The newly created *Product Code* is integrated into the tree view.

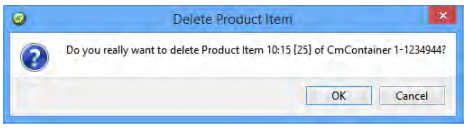
   Using the button adding of further 311 *Product Item Options* is available.

Deleting

For deleting a *Product Code*, please proceed as follows:

1. Select the desired *Firm Item* level.
2. Select the item **"Delete Item"** via:
   - symbol in the context menu (right mouse-click) or in the symbol bar
   - **Edit** menu item of the same name.

   The following dialog asks to confirm deleting the object.



9.1.2.5 Creating, Editing and Deleting a License Option

Please note that *Product Item Options* can be created, edited or deleted exclusively at a *Product Item* level.

Creating

For creating a *Product Item Option*, please proceed as follows:

1. Select the desired *Product Item* level.
2. Select the item **"Modify Item"** via:
   - symbol in the context menu (right mouse-click) or in the symbol bar
   - **Edit** menu item of the same name.

   The following dialog allow to create a new *Product Item Option*.

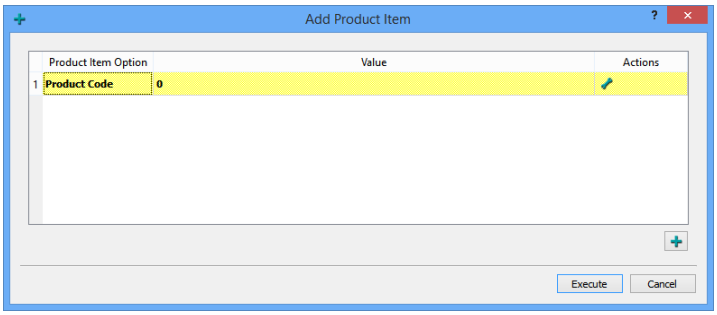
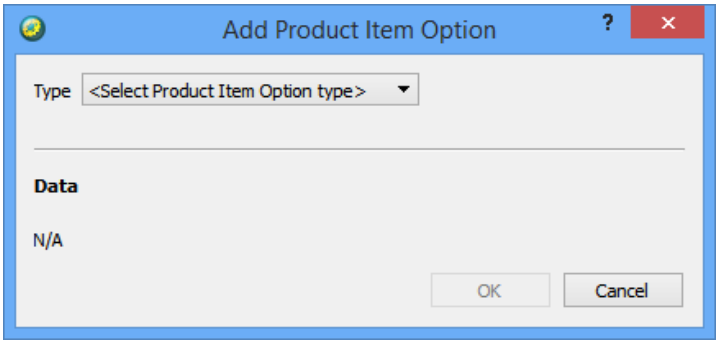


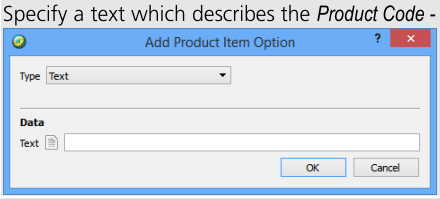
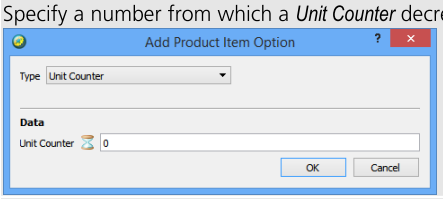
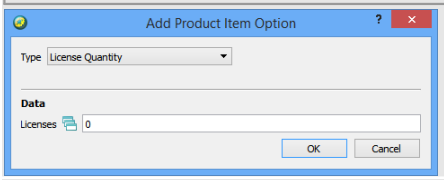
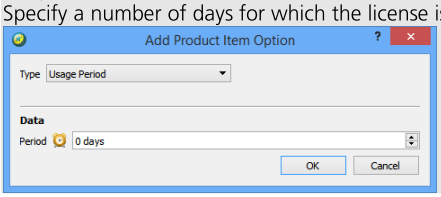
Figure 226: *CodeMeter License Editor – Product Item Option* create

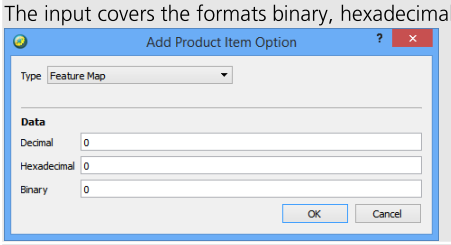
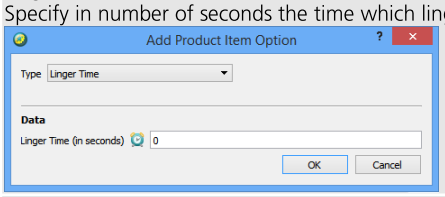
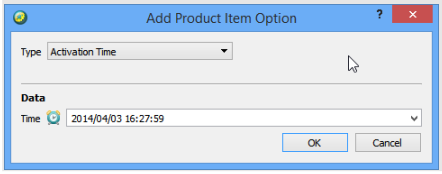
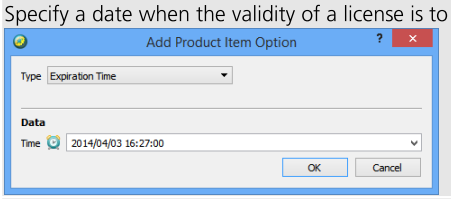
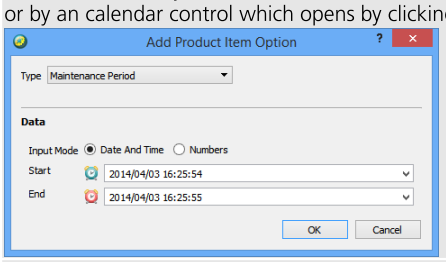
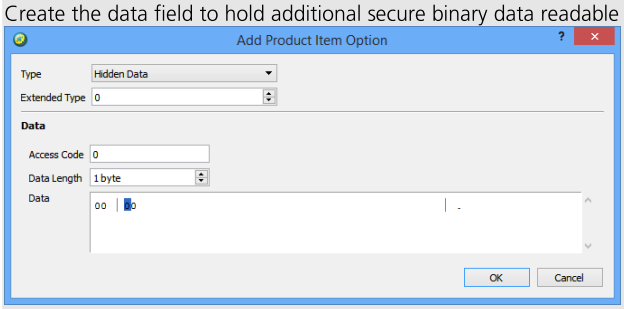
3. Click the button.
An input dialog displays.

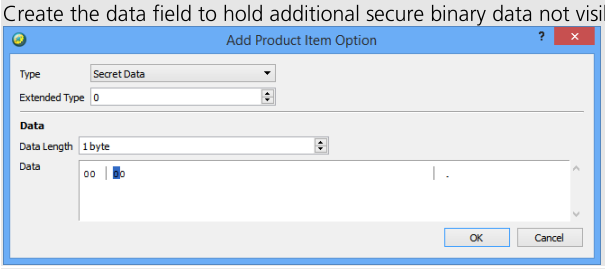
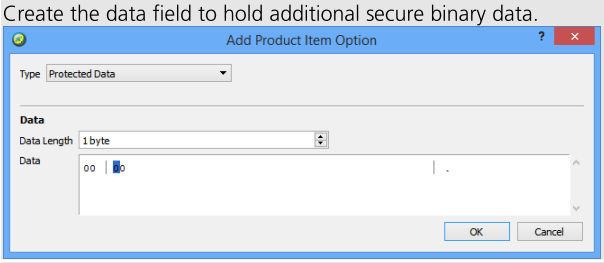
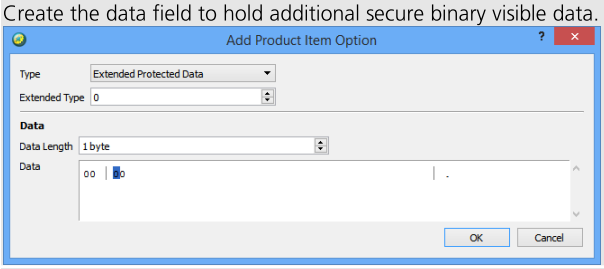
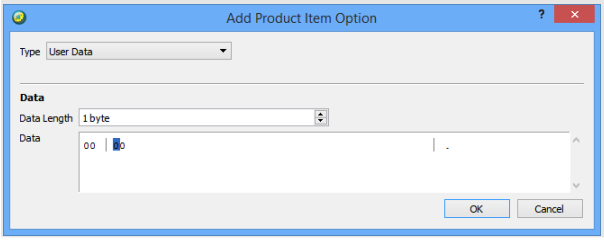


4. Select in field "**Type**" the desired *Product Item Option*.
The following *Product Item Options* are available.

> The dropdown exclusively provides *Product Item Options* not yet created.

Symbol	Product Item Options
	Text Specify a text which describes the *Product Code* - the actual product - in greater detail. 
	Unit Counter Specify a number from which a *Unit Counter* decrement is to start. 
	Specify a number defining how many licenses are simultaneously accessible. The default setting sets a pure local single user license. The license allocation within a network you have already specified in *AxProtector*. 
	Usage Period Specify a number of days for which the license is to be valid. 

Symbol	Product Item Options
	Feature Map Specify the desired combination of features to be activated. Feature may comprise modules, functions, or different versions. The input covers the formats binary, hexadecimal or decimal. 
	Linger Time Specify in number of seconds the time which lingers before another access on the same license is feasible. 
	Activation Time Specify a date when the validity of a license is to begin using the calendar control. Note that when the *Activation Time* has been reached a *Certified Time* update via Internet is required. 
	Expiration Time Specify a date when the validity of a license is to expire using the calendar control. 
	Maintenance Period Specify in the date fields start and end of the *Maintenance Period* for which the license is to be valid. Requires *CodeMeter* Firmware 1.18 or higher. In both fields either specify time dates or integer values in the format used in *CodeMeter*, i.e. seconds since 1.1.2000. This covers the currently valid time horizon in *CodeMeter* until the maximum of February 2136. You specify the data either directly or by an calendar control which opens by clicking on the left arrow symbol. 
	Hidden Data Create the data field to hold additional secure binary data readable only by using a password. 

Symbol	Product Item Options
	Please note, that you can change only either the *Access Code* or the *Data* section but not both at the same time.
	Secret Data Create the data field to hold additional secure binary data not visible. 
	Protected Data Create the data field to hold additional secure binary data. 
	Extented Protected Data Create the data field to hold additional secure binary visible data. 
	User Data Create the data field to hold visible data. 

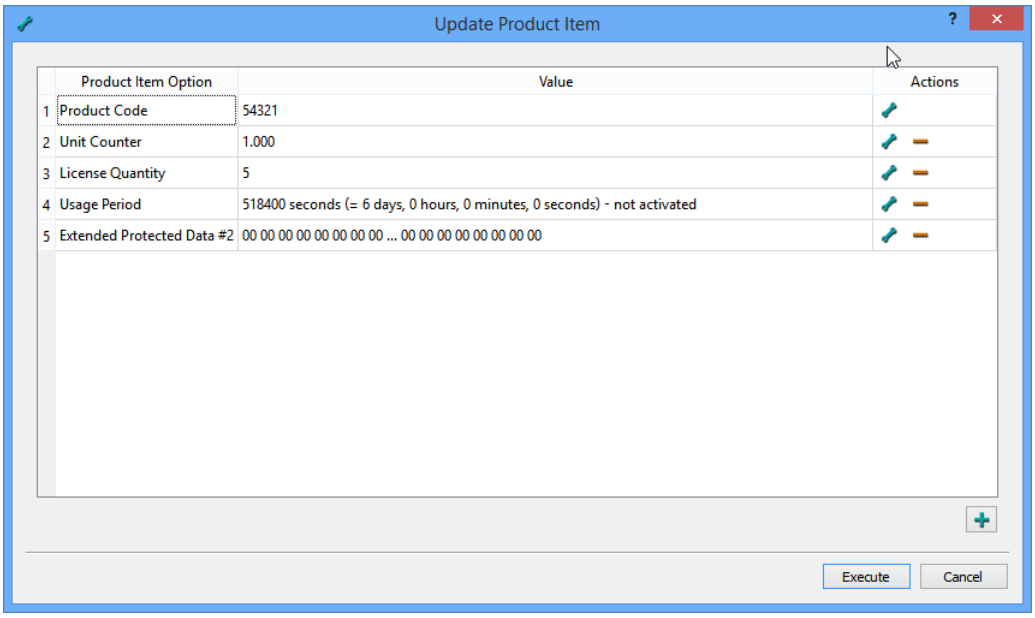
5. Click the "**Execute**" button.
   The newly created *Product Item Options* are integrated in the left display.

Editing

For editing a *Product Item Option*, please proceed as follows:

1. Select the desired *Product Item* level.
2. Select the item **"Modify Item"** via:
   - symbol in the context menu (right mouse-click) or in the symbol bar
   - **Edit** menu item of the same name.

   The following dialog allows to edit an already existing *Product Item Option*.



3. Select the symbol in the respective table row to edit the option:
   The respective option dialog opens. Modifications can be made and saved.
4. Click the "**Execute**" button.

Deleting

For deleting a *Product Item Option*, please proceed as follows:

1. Select the desired *Product Item* level.
2. Select the item **"Modify Item"** via:
   - symbol in the context menu (right mouse-click) or in the symbol bar
   - **Edit** menu item of the same name.
3. Select the symbol in the respective table row to delete the option.
   The symbol displays indicating that the option is registered for deletion.
4. Click the "**Execute**" button to finalize deleting the *Product Item Option*.

9.2 CmBoxPgm

Besides programming of *CmDongle* using *CodeMeter License Editor* 306 and *CodeMeter License Central* 337, *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.

9.2.1 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

Table 8: Time Zones in *CmBoxPgm*

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

9.2.2 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 '/'.

9.2.3 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 319 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*).
`/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*.

9.2.4 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. 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*.

9.2.5 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>`

Command	**`/fltperm`** -Firm Item License Transfer Permissions
	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.

Command	**/fptr** - Firm Precise Time, relative
	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 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.

9.2.6 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 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.

Command	**/papwd** - Access Password
	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. 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.

Command	**/ped** - Extended Protected Data	
	`/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`
	`/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.>]` 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>-`

Command	**/phd** - Hidden Data	
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 362, 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.

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 362, 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.

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

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 362, 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.

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:

Command `/plttype` - License Transfer Typ

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 316 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.

Command	**/pmpi** – Maintenance Period (Integer)
	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.>]` 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.

Command	**/psd** - Secret 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.>]` 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
	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. The key can be given directly as a block of hexadecimal data preceded by '0x'. 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:
	• 128
	• 256

e.g.	`/puvdaes256:r` generates a AES random key in the *CmContainer* with the key length 256 bits.
Command	**/puvddata** - Universal Data Custom Data
	Option for the setup of a *Universal Data* PIO as storage for custom data. Expects either the specification of the data itself or the path to the input file that holds the data as argument. When given directly the data has to be specified as a block of hexadecimal data preceded by `'0x'`.
Syntax	`/puvddata[:]<data>|<file>`
Command	**/puvdperm** - Universal Data User Permissions
	Option for the setup of a *Universal Data* PIO's read, write and use permissions. Expects at least one permission argument. A permission argument consists of the permission type either followed by a • '+' character to grant the permission, or a • '–' character to deny the permission or • up to two password references if password protection (`/puvdpwd`) is activated. Valid permission type specifications are: `r` : (`r`)ead permission `w` : (`w`)rite permission `u` : (`u`)se permission A so-called password reference is the index of a *Universal Data* PIO that holds the password to use. This *Universal Data* PIO must be located at the same *Product Item* level as the PIO to be protected.

Command	**/puvdperm** - Universal Data User Permissions

Syntax	`/puvdperm:<permission #1>[:<permission #2>[:<permission #3>]`
	Each permission has to be composed in the following way:
	`(r|w|u)(+|-|<password ref #1>[,<password ref #2>]]`

e.g.	Write protection by the password stored in *Universal Data* #123
	`/puvdperm:w123`
	Deny read, write protection by the passwords stored in *Universal Data* #123 and #7, grant use permission
	`/puvdperm:r-:w123,7:u+`

Command	**/puvdpwd** - Universal Data Password
	Option for the setup of a *Universal Data* PIO as password storage. The password may be given as text or as a block of hexadecimal data preceded by `'0x'`.
Syntax	`puvdpwd:<password>|0x<hex data>`
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:
	• 2048
	• 3072
	• 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.
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.

9.2.7 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 IDs` as argument. *CmActLicense* license activation request files with a `CmAct ID` that does not match one of these `CmAct IDs` will be rejected.

Command	**/lacids** – Allowed CmAct Ids
	This option only can be used in combination with /laf 328 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` 330 `:none`, the input file argument may be omitted. If option `/lbind` 329 is used, the input file argument must be omitted. The modified license context file argument must be omitted, if the option `/lpo:noramfile` 331 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` 330 and `/laf` 328. 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 general information about a *Universal Firm Code* LIF (License Information File), Context or Modified Context File. The `/l` option can be used to display more detailed license content. Specifying the `/f` (*Firm Code*) and `/p` (*Product Code*) parameters determines the license item level.
Syntax	`/ldf:<file> [/f<Firm Code> [/p<Product Code>]] /l /ldf:<RAC file>|<RAM file>`
e.g.	`CmBoxPgm /f6000010 /L /LDF:"LtModifiedContextFile.WibuCmRaM"` lists the *Firm Item* contenr contained in the file and the general information.

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. *CmActLicense* 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 becomes invalid. • 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 becomes invalid except the tolerance level has been set to a value of `"3"` (loose). • If a previously created snapshot of the VM is reverted, the license becomes invalid.

Command	**/lfs** – *CmActLicense* Binding Scheme (License Feature Set)

***SmartBind* and Azure**

For Windows systems running on the Azure cloud computing platform, newly created CmActLicense licenses of Version 6.90 with the CodeMeter SmartBind binding scheme are now explicitly bound to the cloud computing platform. For Linux systems running on Azure this feature requires at least CodeMeter Version 7.0.

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
	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
`'container'`	Licenses can be used in a container, e.g. Docker (*Universal Firm Code* licenses only). The licenses can also be used, if they use the binding scheme 'none-bind 330 ' together with the binding option 'reimport'.
`'ewffbwf'`	allows EWF/FBWF mode (*Universal Firm Code* licenses only).
`'vm'`	*CmActLicense* license can be used on a (V)irtual (M)achine). Please note the information 329 on *CmActLicense* with *CodeMeter SmartBind* for licenses in VMs.
`'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. The following operating systems are supported:

Operating System	Description
`'Win'`	(Win)dows, all supported Windows versions, Windows 2000 or higher
`'Mac'`	(Mac) macOS
`'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 SmartBind* provides 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.
- 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 43)
`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 --serial 130-1781635890 --context 6000010 --file "Context.WibuCmRaC"
```

In the context of a license transfer use instead:

```
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 323 and Module Items 324 introduced with version 6.00 you may also use all other Product Item Options 319 to map your desired license strategy.

```
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.

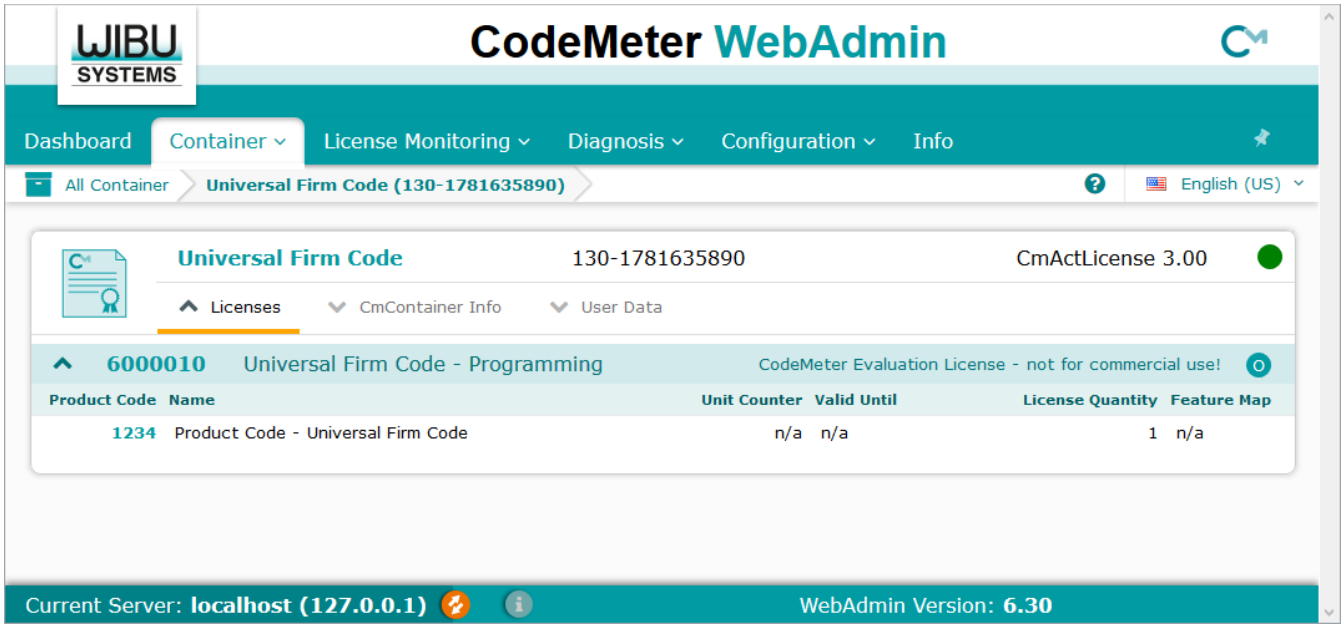
5. Import the Update File (license update, `*.WibuCmRaU`) into the *CmActLicense CmContainer*.
Type the following example commandline:

```
cmu32 --import --file "Universal_FC_Programming.WibuCmRaU"
```

In the context of a license transfer use instead:

```
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[22], 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.WibuCmRaU
" /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.

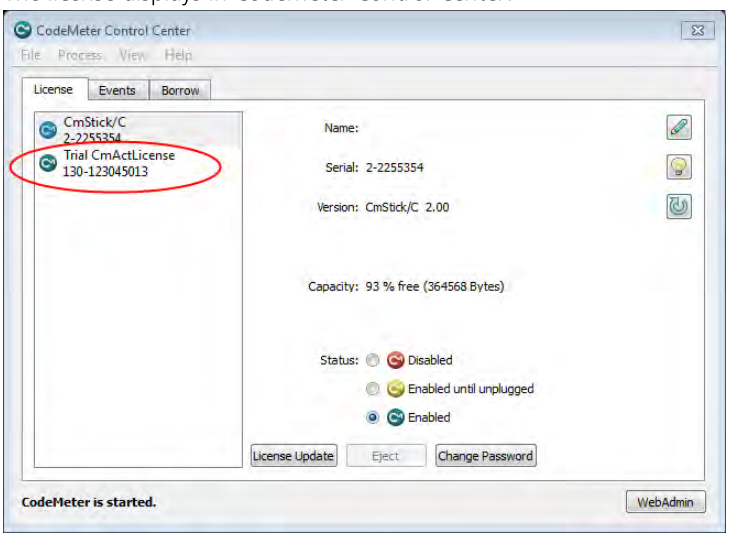
Output

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 [22] 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

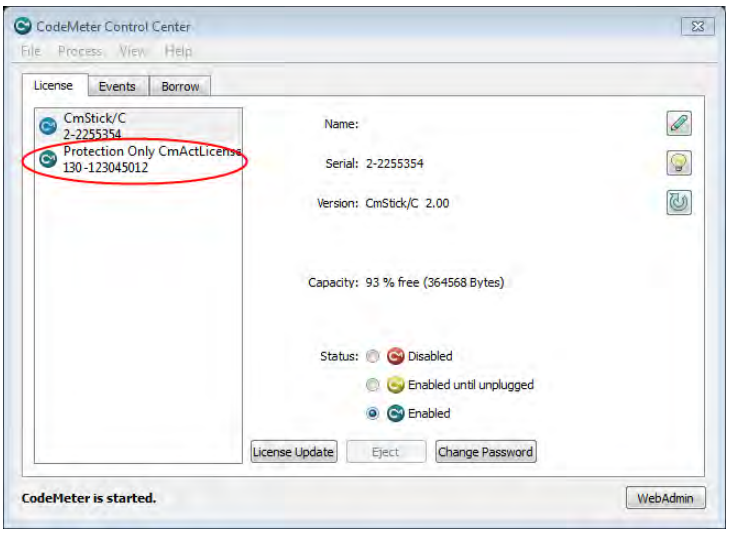
A *CmActLicense* license container with a `F`(irm Code) `f6000010` and a `f`(irm Item)`t`(ext) `"MyFirma"` is updated (`/cau`) and a `p`(roduct Code) `2002` added (`/ca`).
The additional *CmActLicense* options comprise the license activation file (`/laf`) `"UpdateProtectionOnlyLicense.WibuCmRaU"` with 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*.



9.2.8 CmCloud options

This section describes operations related to the licensing system *CodeMeter Cloud*.

The following options are available:

Command	**/clkc** -- Creation of the *CmCloud* Key Certificate
	Use this option to create a *CmCloud* Key Certificate. As first argument the path to the request file (`*.ckcr`) to use is expected. The path to the certificate file (`*.ckcc`) to create may be specified as second argument. Additionally, the following specifications are required: • *Firm Code* by option `/f` and • *CmAct ID* by option `/lpid`
Syntax	`/clkc:<request file>[,<certificate file>]`
e.g.	Create certificate for *Firm Code* = `6000010`, *CmAct ID* = `'3XYZ'`: `/f6000010 /clkc:"RequestData.ckcr","Certificates.clkc" /lpid:3XYZ`

9.2.9 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>`

9.2.10 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).

Command	**/rcl** - Cleanup Registry (*CmDongle* only)
	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`
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.

9.3 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).

9.3.1 The Principle

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*.

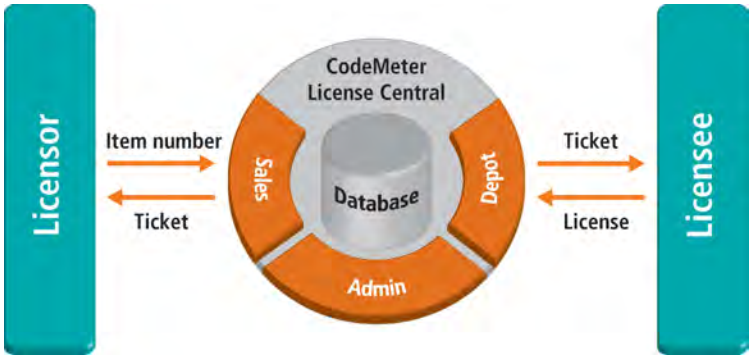


Figure 227: License Collection by Licensee

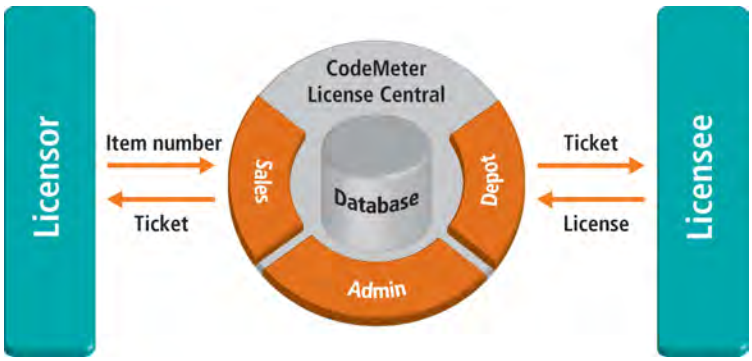


Figure 228: License Collection by Licensor

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*.

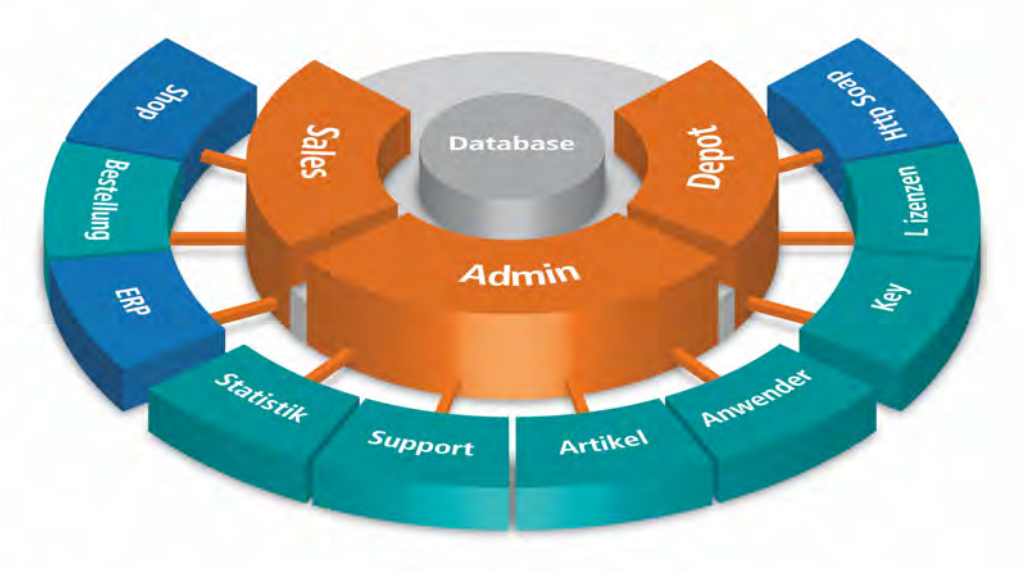


Figure 229: *CodeMeter License Central* - Interfaces and Functions

9.3.2 The Architecture

The core of *CodeMeter License Central* consists of a database and web services for the *Sales*, *Depot*, and *Admin* Interfaces. The web services are cross-platform and available in Java. A Tomcat application server is a prerequisite. The web services provide a SOAP based interface to *CodeMeter License Central*. The complete communications is handled by those web services and the web services have a separate internal interface to the database. Databases supported include MySQL (Windows / Linux) and MSSQL (Windows). On request, other databases can be integrated.

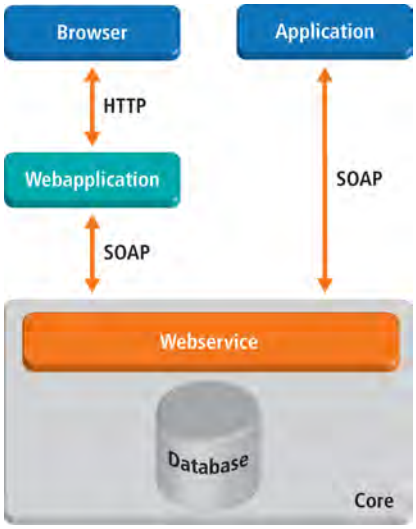


Figure 230: SOAP Access to *CodeMeter License Central*

A web application (Apache/PHP) provides that you are able to instantly use *CodeMeter License Central* without having to make any changes. When integrating the *Sales* Interface into your ERP/CRM system or into your own application a web service is provided.

9.3.3 Functions

The interfaces cover the following functions. Which one of the two *CodeMeter License Central* editions you are allowed to use to integrate your ERP/CRM system or to activate licenses via ticket from within your own software depends on the licensing terms.

9.3.3.1 Sales Interface

The *Sales Interface* accepts activities. You send the item number of the license to be delivered, optionally customer data, and an order number. The *Sales Interface* returns the matching ticket.

In the case of recurring activities (checkpoints, license extension), you are able to send along the original order number. In this case, the existing ticket is extended by a collection activity. This means, the user is able to extend the license with his established ticket. This saves management efforts for new tickets, and eases handling for the user. You can also extend or renew the license with the existing ticket directly via SOAP from within your application.

Depending on the item configuration, you are also able to dynamically transfer parameters on an activity. For example, you transfer the number of network licenses, or the purchaser's name written to the property: *Customer Owned License Information*.

You are also able to easily and efficiently integrate your sales processes into online shops or CRM/ERP systems (Connectors).

9.3.3.1.1 Connectors

On creating tickets *CodeMeter License Central* allows the integration of existing systems by using interconnected adapter, the so-called Connectors.

This option is not available for the *Desktop* edition; here the ticket is manually created via the web interface of the browser.

In contrast, in the *Internet* Edition tickets can also be automatically created using a SOAP interface. This is feasible because in *CodeMeter License Central* the actual sales process is separated from the generation of the license. On selling a license *CodeMeter License Central* creates a ticket which allows collecting the license at a later time. This allows the integration into existing systems, such as, for example, online shops or ERP/CRM systems.

In a technical perspective, Connectors are flexible adapters allowing the data transfer between the SOAP interface and other existing systems. The transfer is based on data mapping. A Connector is, however, not merely an adapter which maps data formats; it also intervenes in the process as an active component. That is, it saves additional information to separate tables that is delivered by the online shop but not required for the ticket generation in *CodeMeter License Central*. The Connector then is able to read data from those tables, and use it for the creation of the ticket. Or, it can execute the ticket delivery via e-mail.

Connectors are implemented by addressing a standardized web service allowing automated communication.

Online Shops

For online shops, such as, for example, asknet, Cleverbridge, Digital River, element 5, or ShareIt which provide web-based license generators basic Connectors (pure data mapping tools) are already provided in PHP.

ERP/CRM Systems

While most online shops provide easy to configure web-based interfaces, integration into an ERP/CRM system is much more customizable. However, the principle integration process is the same. The ERP/CRM system calls a Connector which in turn processes the data, serves the SOAP interface from *CodeMeter License Central*, and returns the ticket to the ERP/CRM system.

In addition to having the ERP/CRM system starting the Connector, you also have the option of developing an individual Connector that periodically reads data from the ERP/CRM system, or imports exported data. This scenario is quite common. And it can be used if the

ERP/CRM support team does not wish, or is not able to customize processes. A disadvantage of this solution is that the ticket data is not available in the ERP/CRM system. In the case of support incidents, you have to search for information from two different systems: the ERP/CRM system and *CodeMeter License Central*. The advantage of the solution, however, is an easy straight-forward implementation; a periodical automatic export of the data from the ERP/CRM system has been possible in all projects so far.

WIBUconcepts supports you in individual integration with consulting and professional services from design to implementation. Do not hesitate to contact us.

9.3.3.1.2 Gateway

With a Gateway you can collect licenses directly from within your protected application.

A completed Gateway for the automatic collection of licenses via the Internet is part of and available for the *Internet* Edition of *CodeMeter License Central*. The Gateway is written in PHP.

Why do you need a Gateway?

When completing a sale, *CodeMeter License Central Internet Edition* lets you decide whether your customer should use our web interface or our SOAP Gateway to collect the ticket.

In most cases, *CodeMeter License Central* will not be available from the Internet directly. But for security reasons, it is available from an internal network. That is why a direct access from outside via SOAP (from the software installed on the customer's side) is not possible. You need a special kind of software, which is located in the DMZ (demilitarized zone) and replies to inquiries from outside and forwards them to CodeMeter License Central. We call this kind of software a Gateway.

Personalized additional information

Like a Connector, the Gateway can do more than just forward inquiries. The Gateway allows you to link advertising messages to license information, and then deliver it "piggy-back" to the customer for up-selling, cross-selling, or other marketing campaigns.

You can also deliver software updates via the Gateway, because the Gateway can access the license information of the corresponding customer and filter individual offers or updates.

Remote and Update file

Whether collecting the license via a browser or Gateway, the basic principle is the same. A remote context file is created by using the desired *CmContainer*.

Together with the ticket, the remote context file is sent to *CodeMeter License Central*, which in turn checks whether the ticket is valid (does the ticket exist, and does it remain uncollected), creates the appropriate remote update file for this *CmContainer* or PC, and then transmits this file as a reply. The update file is then copied into the *CmContainer* or onto the PC. If you use the web interface, an ActiveX Plug-in or Java Applet creates the remote update file, and copies it on the customer´s side. Optionally the customer can create the remote update file manually, and upload it to the web interface. This is especially appropriate, if the PC to which the license has been bound, or to which the *CmContainer* is attached, does not have any access to the Internet. In this case, neither the ActiveX nor the Java Applet is necessary. In fact, the *CodeMeter* Runtime does not even need to be installed.

The Standard Gateway

Using a Gateway, you can create the remote context file by yourself, send it together with the ticket to *CodeMeter License Central*, and copy the remote update file to your system.

This Gateway is alternatively accessed by HTTP/POST or by HTTP/GET and collects all open licenses coupled to the corresponding ticket. The same mechanism is used on the Internet when you send a form to a server. The only difference is that the Gateway does not reply as an HTML site, but with a remote update file.

Calling the Gateway

There are class libraries available in many programming languages that are used to send a HTTP request.

Remote Context- and Update file

Use the function ***CmGetRemoteContextBuffer*** and ***CmSetRemoteUpdateBuffer*** from the CodeMeter Core API to create and deliver the remote update file. These functions are available with version 4.0.

Where to generate the request?

Collecting licenses via a Gateway is pretty simple and straight forward. But where is the ideal place to generate a request? Within your protected application? Within your error handling DLL file, which is called up from the protected application? Maybe you want to use an additional application to activate it?

Depending on your scenario, one of these three solutions is the proper way. Experience has shown that providing an additional application for activation has proven to be the most flexible solution. If a customer already owns a basic version of the software and wants to activate another module, you can start the application for activation from your protected software. If the customer does not have a license yet, you can start the application for activation from your error-handling DLL file.

Even if your customer wants to activate a network license, the application for activation is the ideal solution. The customer only needs the *CodeMeter* Runtime and your application for activation on the server.

9.3.3.2 Depot Interface

The *Depot Interface* features license collection. The collection involves the upload of a content file and the download of an update file. Optionally, after an update file is activated, a new context file may be uploaded in order to send a receipt for the license activation. Of course, this process can be achieved in one step so that the user is collecting the license only.

The *Depot Interface* offers two options to collect licenses:

- direct (PC with the *CmContainer* to be programmed has Internet access)
- indirect (Activation data is transferred to another PC via file transfer)

Next to the license collection, the *Depot Interface* also provides for methods for returning licenses. After returning the license, the user receives a new ticket. He receives it only after uploading the receipt. Using this new ticket he or she is able to transfer the license to another PC, or is able to resell it passing the ticket to the new user. If you allow the reselling of licenses, then simply activate the option: License Returning. By default, license returning – and thus reselling – is disabled.

Moreover, in the *Depot Interface* you are able to retrieve information on sold and activated licenses.

Depending on the product configuration, you are able to preset the licensing system for the end user or let the user opt for hardware-based or software-based protection.

License Collection by the Licensee

In the case the licensee is to directly collect the licenses, then s/he requires access to *CodeMeter License Central*. Depending on the envisaged access – directly via SOAP from within the application or via a website – place the web server or the web server and application server into the DMZ (Demilitarized Zone). For security reasons, in this case we recommend to span the installation using several PCs and to locate the remaining modules (database and eventually the application server) behind the inner firewall.

9.3.3.3 Admin Interface

The *Admin Interface* consists of the following parts; license configuration, evaluation, support, and user management.

In license configuration you are able to manage license properties and the related item numbers. Here you individually define for each license which parameters are preset, and which are dynamically transferred to the *Sales Interface*.

In the statistics module you are able to evaluate data from *CodeMeter License Central*, for example licenses on *CmContainer* per customer.

For closing open processes (e.g. receipt not uploaded), the release of further activations, and the editing of blacklist entries, you use the support module.

User management provides you with the option to configure the "access privileges" to *CodeMeter License Central*. Those include user name, password, IP range, and *CmContainer*. For example, you can set it up so that a sales partner with changing IP addresses has to authenticate using a CmContainer, while a sales partner with a web portal must log on using an authorized IP address.

9.3.4 Application Scenarios CodeMeter License Central

Using *CodeMeter License Central* for example may span the following scenarios.

Scenario	Description
Single User	Here *CodeMeter License Central* is locally installed on a single user PC as VM image and runs within the VMware Player or the VMware Workstation. Using a browser the user accesses *CodeMeter License Central*. The advantage in this case: all required components are already installed, and database management is not required.
Small Network/ Intranet	Here the *Desktop* Edition of *CodeMeter License Central* is installed on a server and the staff is able to access *CodeMeter License Central* using a browser. The advantage in this case: all staff is accessing a central database.
Online Accessibility	Here the *Edition* Edition of *CodeMeter License Central* is installed on a server. The Gateway locates in the DMZ. The customer is able to activate licenses from within the protected application running on his/her PC. The advantage in this case: the customer is able to activate licenses from within the protected application running on his/her PC.

Online Shop	Here the *Edition* Edition of *CodeMeter License Central* is installed on a server. Via a Connector located in the DMZ the Online Shop and *CodeMeter License Central* communicate. The advantage in this case: you are able to use web-based license generators of popular online shops to create tickets.
ERP/CRM Integration	Here the *Internet* or *Edition* of *CodeMeter License Central* is installed on a server. The ERP/CRM system calls an internal Connector which processes the data and forwards it to *CodeMeter License Central*. The ticket generated this way is sent back to the ERP/CRM system. The advantage in this case: license information can be combined with information on customer data, order processing, accounting, etc.

9.4 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 391 .

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.

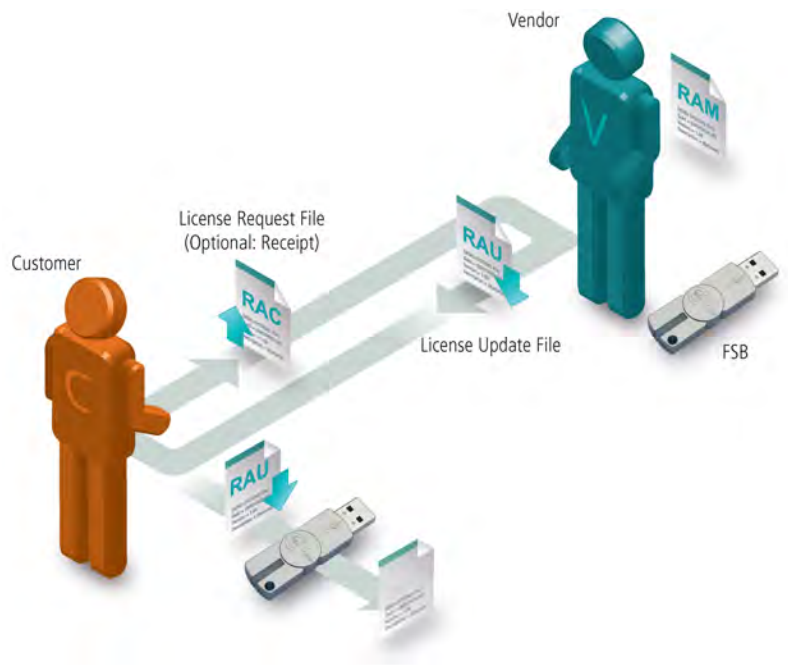


Figure 231: *CmFAS* - File-based Remote Update *CmDongle*

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 21 and additional activation options 22 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.

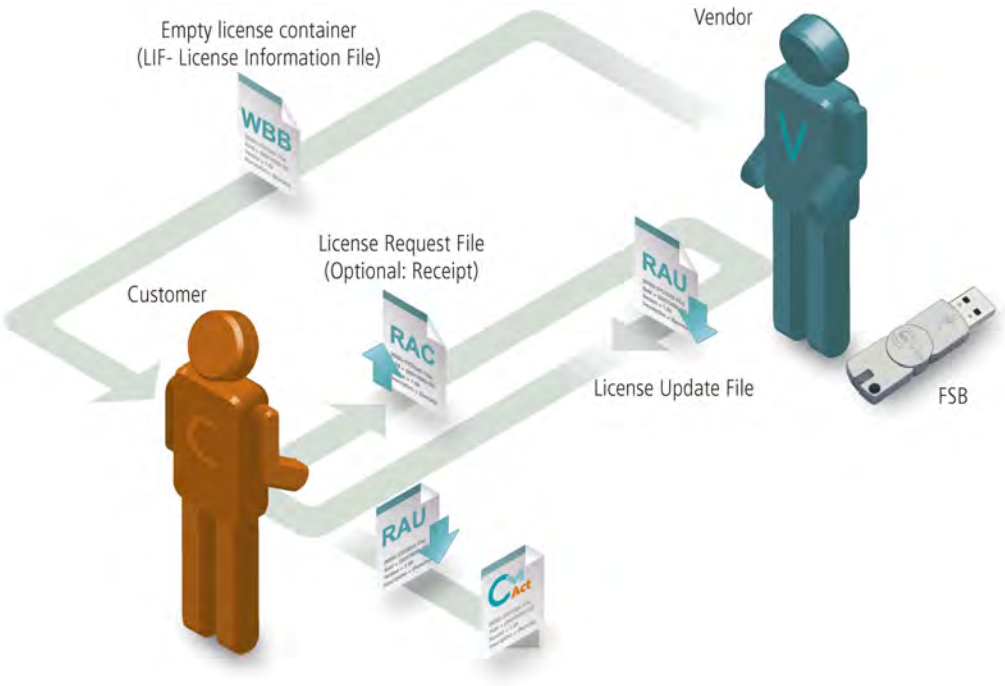


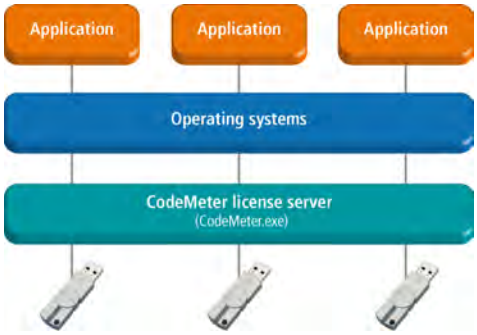
Figure 232: *CmFAS* - File-based Remote Update *CmActLicense*

10 Deployment

After you successfully protected your software with the matching license information, you deliver it to the end-user. The deployment process covers a manageable number of elements you send to your customers providing an optimal and trouble-free execution of your protected software.

No separate driver installation required

Many dongle manufacturers provide separate Kernel drivers for directly accessing the dongle. Wibu-Systems takes another path. Wibu-Systems relies on the proprietary *CodeMeter License Server* to act as a central turntable providing all communication tasks for *CmContainer*. *CodeMeter License Server* communicates between the *CmContainer* using USB, Mass Storage Device or other drivers provided by the operating system and the interface to the protected software you deliver.



A separate driver installation for different operating systems is not required.

Respective operating system updates automatically include these drivers. So you do not have to wait for an Wibu-Systems update with the latest drivers. The software you protected using Windows Vista immediately works under Windows 7 with the same *CodeMeter* Runtime update.

Recommendation

Nevertheless, Wibu-Systems recommends the use of the installation packages available in the download area on the website (www.wibu.com). This avoids version conflicts eventually caused by the simultaneous installation of several products requiring *CodeMeter License Server*.

When installing an update, you as software vendor guarantee that always the latest *CodeMeter* works with the latest *CodeMeter License Server* version. Moreover, the installation packages include some additional support tools.

Wibu-Systems does not recommend deployment of your protected application by simply copying the required elements. By copying them to a separate application directory, and the simultaneously use of several *CodeMeter* protected applications may lead to version conflicts.

Wibu-Systems in this case, does not assume responsibility for version conflicts at runtime of the application.

An exception exists in the case of complete systems, i.e. no other software vendor uses a *CmDongle*, and no *CodeMeter* protected applications are installed except the own software. For example, in the case of pre-installed computer of cash or central fire alarm system.

10.1 Installation packages for Non-Windows Operating Systems

The minimum requirement for the deployment of your protected software consists of the *CodeMeter* License Server, i.e. `CodeMeter.exe`. Wibu-Systems recommends for installing the existing installation packages for different operating systems.

As software vendor, you are allowed to transfer these complete packages free-of-charge to your end-customers. Alternatively, your customers are also able to directly download the packages from the user area of the Wibu-Systems website (http://www.wibu.com/en/downloads-user-software.html) to install the files - free-of-charge, without password and the requirement to register.

For Non-Windows operating systems the following installation packages exist:

macOS
CodeMeter Runtime Kit (macOS starting with 10.9)
Linux
RPM packages, e.g. SuSe, Red Hat
CodeMeter Runtime 64-bit - for PC on AMD64 basis
AxProtector/Java Runtime - for *AxProtector* protected Java applications
CodeMeter Runtime - contains all required file for the end user
CodeMeter Lite - driver only installer for systems without GUI
DEB packages, e.g. for Debian, Ubuntu

Linux
CodeMeter Runtime 64-bit - for PC on AMD64 basis
AxProtector/Java Runtime - for *AxProtector* protected Java applications
CodeMeter Runtime - contains all required files for the end user
CodeMeter Lite - driver only installer for systems without GUI

10.2 Deployment on Windows Operating Systems

The minimum requirement for the deployment of your protected software consists of the *CodeMeter* License Server, i.e. `CodeMeter.exe`. For the deployment Wibu-Systems provides pre-configured installation packages [346] for Windows.

As software vendor, you are allowed to transfer these complete packages free-of-charge to your end-customers. Alternatively, your customers are also able to directly download the packages from the user area of the Wibu-Systems website (http://www.wibu.com/en/downloads-user-software.html) to install the files - free-of-charge, without password and the requirement to register.

Also separate merge modules [347] are available which comprise files, registry entries and settings of specific runtime components. Setup developer are able to use them for own installer.

Start of *CodeMeter License Server* after installing

At the end of installing, the service `CodeMeter.exe` is started and checked, if it runs.

> If the embedded product installer has been created using WiX, the error case provides a GUI message on setting the property `WIXUI_EXITDIALOGOPTIONALTEXT` issuing "*The CodeMeter service (`CodeMeter.exe`) has been installed but cannot be started successfully.*"

However, this behavior after installing can be changed. Then the installation is canceled, if the service `CodeMeter.exe` could not be started and a roll back of the installation is performed.

For this the installation property `CM_CMSVC_START` must be set to the value `"Standard"`. Then the start behavior of *CodeMeter Licence Server* corresponds to the standard Windows operating system procedure for starting services.

The installation property `CM_CMSVC_START` may also assume the following values:

- `"none"` - if set, then the *CodeMeter Licence Server* service is not started.
- not set (default) or `"custom"` - the service is started as via the `CustomAction "StartCmSvcXX"`. The installation is not canceled. if `CodeMeter.exe` could not be started.

In addition, a property `CM_RETURNCODE` is set with a default value of `0`. If the service does not run, momentarily this value is set to a value of `110`.

10.2.1 Pre-configured Installation Packages

Full Installation Package

This package holding all necessary components of the *CodeMeter* Runtime is available for 32- and 64-bit operating systems.

It is available as executable file (`CodeMeterRuntime32/64.exe`) and as separate package for Managed Software Installation using the Windows Installer service `msiexec.exe` (`CodeMeterRuntime32/64.msi`).

Reduced Installation Package

This package also available for 32- and 64-bit operating systems presents a reduced functional scope of *CodeMeter* Runtime. Not included are the relevant files of *CodeMeter Control Center*, the separate User Help, and the entries is the Windows start menu (shortcuts).

It is available as executable file (`CodeMeterRuntime32/64Reduced.exe`) and as separate package for Managed Software Installation using the Windows Installer service `msiexec.exe` (`CodeMeterRuntime32/64Reduced.msi`).

> The executable file of the reduced installation package is not downloadable in the user section of the Wibu-Systems website but in the developer section.

> If you use the reduced installation package, please note that the *CmDust* entry of the start menu is no longer available. Creating the log file then alternative must be triggered [450] using the commandline tool *cmu*.

Installation Package for applications using FSB functions

This package available for 32- and 64-bit operating systems contains the *CodeMeter* Runtime and the module `CmRuntimeInternal` with FSB functionalities. This allows, for example, to use a FSB License Server on a network or to provide *CodeMeter* encryption in a integrated developer environment (IDE).

It is available as executable file (`CodeMeterRuntimeLicensor32/64.exe`) and as separate package for Managed Software Installation using the Windows Installer service `msiexec.exe` (`CodeMeterRuntime32/64Licensor.msi`).

***CodeMeter* Merge Modules**

For single components of *CodeMeter* Runtime Wibu-Systems also provides merge modules you are able to build into own separate installer.

These `*.msm` files are not independently installable comprise files, registry entries and settings of single runtime components.

Download these modules from the password-protected developer section at the Wibu-Systems website (http://www.wibu.com/de/software-development-kit.html).

The following files are part of the Wibu-Systems *CodeMeter* Runtime Distribution for Windows:

File	Merge Module
`CmRuntimeMerger.msm`	CodeMeter Runtime (Win 32)
`CmRuntimeMergerReduced.msm`	CodeMeter Runtime with reduced scope (Win 32)
`CmRuntimeMerger64.msm`	CodeMeter Runtime (Win 64 / x64)
`CmUserHelp.msm`	CodeMeter User Help
`ShellExtMerger32.msm`	Wibu-Systems Shell Extension (Win32)
`ShellExtMerger64.msm`	Wibu-Systems Shell Extension (Win 64 / x64)
`WibuCmNet.msn`	Holds .NET policies

The *CodeMeter* Runtime merge modules hold all necessary parts of *CodeMeter* Runtime Kit, such as, *CodeMeter License Server*, *CodeMeter Control Center* and the runtime libraries.

The merge modules `CmRuntimeMerger.msn` or `CmRuntimeMerger64.msn` must be installed in each system. In the reduced merge module not are the relevant files of *CodeMeter Control Center*, the separate User Help, and the entries is the Windows start menu (shortcuts).

The merge module `CmRuntimeMerger64.msn` is required for *CodeMeter* accessed to 64-bit applications. If no 64-bit application is delivered, it is not necessary to install this module.

The merge module `CmUserHelp.msn` installs the User Help to the target system helping you customers to get familiar with *CodeMeter*.

The merge module `Wibu-ShellExtMerger32/64.msn` hold, among other things, the extension to execute remote update files by double-clicking.
The merge module `WibuCmNet.msn` is required when delivering .NET applications. It holds, among other things, references of the Global Assembly Cache (GAC).

Downgrade behaviour (valid since Version 6.30)

CmRuntimeMerger

If single merge module component integrated in your installer hold an older runtime version than the one installed on the target system of your customer. then the installer continues. However, no elements of the merge module component are installed. In the installer log a comment is added. The older runtime version is not replaced.

If the downgrade case occurs, a merge module-specific property is set for the merge module elements which are then not installed. The value of `PROP_CM_NODOWNGRADE` contains the downgrade note.

z.B. "A newer CodeMeter Runtime Kit Version XXXX is already installed. Downgrading CodeMeter Runtime components to the older Version YYYY was skipped. To downgrade you have to uninstall your CodeMeter Runtime Kit manually."

If you want, however, to allow the downgrade, you can do that using the property `CM_ALLOW_DG`. This property must be set externally using `CM_ALLOW_DG.[ID` des Merge-Moduls`]`.

z.B. `CM_ALLOW_DG.A961A077_4BD0_4C98_86BC_EE4A98CE550D="1"` for *CmRuntimeMerger*
`CM_ALLOW_DG.1992E333_D17A_448B_8484_ED047109D182="1"` for *CmRuntimeMerger64*

If the installer runs in UI mode including a user interface, the product installer property `WIXUI_EXITDIALOGOPTIONALTEXT` is set with the message text above. If the product installer has been generated using Wix, in the exit dialog this message text is included. For üproduct installer other than Wix (InstallShield, Wise etc) this UI feature is not supported.

Firewall Settings

By default, *CodeMeter* uses TCP/IP for communicating with protected applications and for displaying information in *CodeMeter WebAdmin*. To ensure that this also works with an activated "Windows Firewall", please specify the *CodeMeter License Server* merge modules into the private and public profile as exception for *CodeMeter License Server* (`CodeMeter.exe`). On 'mobile' use of *CodeMeter*, i.e. without use of merge modules *CodeMeter License Server* checks for itself for an exception entry in the actual firewall profile and set exceptions in case they do not exist. This, however, only if *CodeMeter License Server* has been started with administrator privileges.

Firewall applications of vendors other than Microsoft are currently not supported. Eventually, here you have to specify the exceptions manually.

10.2.2 Customizing Options for Installation Packages

In majority of the cases the pre-configured installation package of *CodeMeter License Server* in Form of executable files (`*.exe`), Windows installation packages (`*.msi`) and merge modules (`*.msm`) meet the delivery and installation requirements of software protected and licensed using *CodeMeter*.

In exceptional cases, however, it may be required to further customize the pre-configured installation packages.

For this purpose Wibu-Systems provides several procedures: installing options 348, directed installing of features 348 and using of central configuration parameter on integration of merge modules 349 into own installer.

Installing Options

In the case of Windows operating systems, you have the option to configure the executable *CodeMeter* Runtime installation package by specifying additional parameter. For listing all available commandline options, please proceed as follows:

1. In an open commandline prompt window type in the following commandline:
   `CodeMeterRuntime64.exe /?`
   A separate window opens displaying available options.

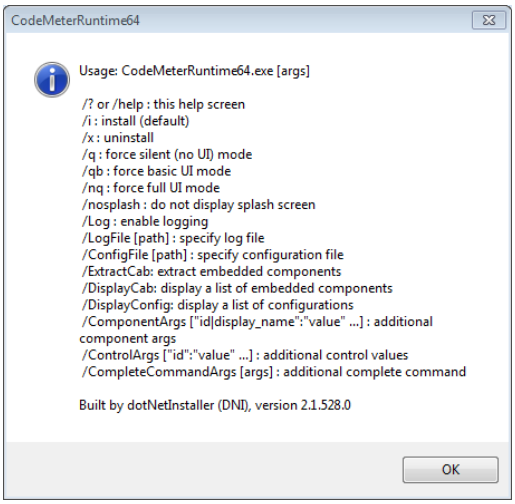


Figure 233: `CodeMeterRuntime.exe` - Commandline options

> Please note that the options '`/q`, `/qb`, `/nq`' for the GUI display control and "silent" procedures are obsolete with *CodeMeter* version 5.0a.
>
> This is the consequence of changing default behaviours of the EXE- and MSI-Installer.
>
> Since the EXE-Installer (Bootstrapper), by default, now starts in "silent" mode, changes can be made only by addressing the MSI-Installer which, by default, starts in GUI mode.

Starting with *CodeMeter* Version 5.0a the GUI display control and "silent" installing procedures are controlled by using the commandline flag `ComponentArgs`.

e.g.

The commandline input
`CodeMeterRuntime.exe /ComponentArgs "*":"/qn"`
allows a "silent" installation without user interaction.

The commandline input
`CodeMeterRuntime.exe /ComponentArgs "*":"/qn REINSTALLMODE=ocmusv REINSTALL=ALL"`
performs a "silent" repair installation.

The commandline input
`CodeMeterRuntime.exe /x /ComponentArgs "*":"/qn"`
performs a "silent" uninstalling.

Directed Installing of Features

By specifying additional options of the commandline flag `ComponentArgs` you are also able to explicitly define which features are to be installed.

Note the following rules when integrating explicitly features to be installed using the flag `ComponentArgs`:

- `ADDLOCAL` installs the features
- `REMOVE` removes already existing features
- Specify the Feature ID names
- Single Feature-ID names are separated by comma.

> The features to be installed follows the `ADDLOCAL` part. Features not listed are not installed.

The following table lists all Feature ID name of the full executable installation package:

Feature ID	Description
`Complete`	Main feature, holds the `CmRuntimeMerger` module and the following secondary features

Feature ID	Description
DotNET_Modules	holds the WibuCmNet.msn module and holds the file wibucmnet.dll, the language files. and the policy files
WibuShellExtension	holds the ShellExtensionMerger module
User_Help	holds the CmUserHelp module
EnableNetworkServer	activates the operation as network server
AutomaticServerSearch	standardmäßig ist die Automatische Server-Suche aktiviert, d.h. es wird zunächst lokal und danach im Netzwerk (Subnnetz) nach Lizenzen gesucht.

e.g. The commandline input:
```
CodeMeterRuntime64.exe /componentargs "*":"/l*v Runtime_msi.log
ADDLOCAL=Complete,WibuShellExtension,User_Help"
```
installs next to the CmRuntimeMerger module the features WibuShellExtension and the UserHelp but not the .NET_Modules.

The following table lists all Feature ID name of the reduced executable installation package

Feature ID	Description
Complete	Main feature, holds the CmRuntimeMerger module and the following secondary features
DotNET_Modules	corresponds to the WibuCmNet.msn module and holds the file wibucmnet.dll, the language files. and the policy files
WibuShellExtension	holds the ShellExtensionMerger module
EnableNetworkServer	activates the operation as network server
AutomaticServerSearch	by default, the automatic server search is activated, i.e. licenses are sought first locally and then on the network (subnnet).

e.g. The commandline input:
```
CodeMeterRuntime64Reduced.exe /componentargs "*":"/l*v Runtime_msi.log
ADDLOCAL=Complete,DotNET_Modules"
```
installs next to the CmRuntimeMergerReduced module the additional feature.NET-Module but not the WibuShellExtension module.

Integrating Merge Modules in separate Installer using Configuration Parameter

By introducing central configuration parameter you are also able to control for the merge modules CmRuntimeMerger and CmUserHelp whether, for example, on installing *CodeMeter Control Center* is to automatically start and whether entries in the Windows start menu (shortcuts) are to be created.

Here the parameter PROP_CMCC for the start behavior of *CodeMeter Control Center* and PROP_MAKESC for the creation of shortcuts exists.

CmRuntime Merger Modules

In the module CmRuntimeMerger the parameter PROP_CMCC and PROP_MAKESC with the following behavior and the following pre-defined values are available.

Parameter PROP_CMCC

Value	Description
None	• Preventing the start of *CodeMeter Control Center* at the end of the installation • Disabling the *CodeMeter Control Center* entry in the auto start directory
run	Disabling the CodeMeter Control Center entry in the auto start directory
auto	Preventing the start of *CodeMeter Control Center* at the end of the installation
all	• Start of *CodeMeter Control Center* at the end of the installation • Enabling the *CodeMeter Control Center* entry in the auto start directory If parameter PROP_CMCC is not set, it corresponds to value all.

e.g. Preventing the start of *CodeMeter Control Center* at the end of the installation:
```
CodeMeterRuntime64.exe /componentargs "*":"/l*v rtk_install.log PROP_CMCC=""auto"""
```

Parameter PROP_MAKESC

Value	Description
no	Preventing the creation of any shortcuts (*CodeMeter Control Center*, *User Help*, *CmDust* etc.) Please note that the CmDust entry in the start menu no longer exists. Creating the log file then alternative must be triggered 450 using the commandline tool *cmu*.
yes	Creating all shortcuts (*CodeMeter Control Center*, *User Help*, *CmDust* etc.)

Value	Description
	If parameter `PROP_MAKESC` is not set, it corresponds to value `yes`.

Preventing that on *CodeMeter* Runtime Kit installation shortcuts are created:

```
CodeMeterRuntime64.exe /componentargs "*":"/l*v rtk_install.log PROP_MAKESC=""no"""
```

If the `CmRuntimeMerger` module is controlled via `PROP_CMCC` and `PROP_MAKESC` then the value `PROP_MAKESC="no"` also prevents the auto start entry since this also presents a shortcut.

CmUserHelp Module

In the module `CmUserHelp` the parameter `PROP_MAKESC` with the following behavior and the following pre-defined values is available.

Parameter `PROP_MAKESC`

Value	Description
`no`	Preventing the creation of a start menu entry for the *User Help*.
`yes`	Disabling the *User Help* entry in the start menu If parameter `PROP_MAKESC` is not set, this corresponds to the value `yes`.

10.3 Mobile Installation on CmDongle (Windows)

Optionally, *CodeMeter* ships with flash memory part (*CmStick/M*) in addition to the copy protection chip (*CodeMeter* chip). It allows you to deliver your software directly on the USB device without the installation of drivers and without giving up secure software protection.

For the mobile use of *CodeMeter* you only require *CodeMeter License Server* (`CodeMeter.exe`) located in the directory [`%Program Files%\CodeMeter\Runtime\bin`].

Copy this file together with your protected application to the same directory in the flash memory of a *CmDongle*. On starting the protected application, `CodeMeter.exe` auto-starts, and the application is able to communicate with *CodeMeter License Server*.

In order to provide your customers with the complete functional scope of *CodeMeter* copy the following files into the application directory of your application on the *CmDongle*:

File	Description
`CodeMeter.exe`	*CodeMeter License Server*
`CodeMeter.l**`	Language files for *CodeMeter License Server*
`CodeMeterCC.exe`	*CodeMeter Control Center* including support tool *CmDust*.
`CodeMeterCC_**.qm`	Language files for *CodeMeter Control Center*
`CmWebAdmin.exe`	*CodeMeter WebAdmin*
`WibuCm32.dll`	*CodeMeter* runtime library (from `%Windows%/system32`)
`WibuCm32.l**`	Language files for the runtime library (from `%Windows%/system32`)

For mobile installation add to the *CodeMeter* runtime a `CodeMeter.ini` file on the *CmDongle*. Then all settings are read from and written to the `CodeMeter.ini` file.

The result is that no residual traces are left on your hard disk, or in the PC registry.

The Configuration File `CodeMeter.ini`

The configuration file `CodeMeter.ini` holds all settings of *CodeMeter License Server*.

In order to create a `CodeMeter.ini` file with PC-independent default values, create an empty file with the name `CodeMeter.ini` in the same directory where `CodeMeter.exe` locates.

On restarting `CodeMeter.exe` all standard settings are written to this file. All changes to the configuration you now apply using *CodeMeter Control Center* or in *CodeMeter WebAdmin* are automatically saved to the `CodeMeter.ini` file.

What happens when *CodeMeter* is already installed on the PC? No problem again. When *CodeMeter License Server* is already installed and running, then this instance is used. Then all automated mechanisms for starting or exiting the *CodeMeter License Server* are suspended.

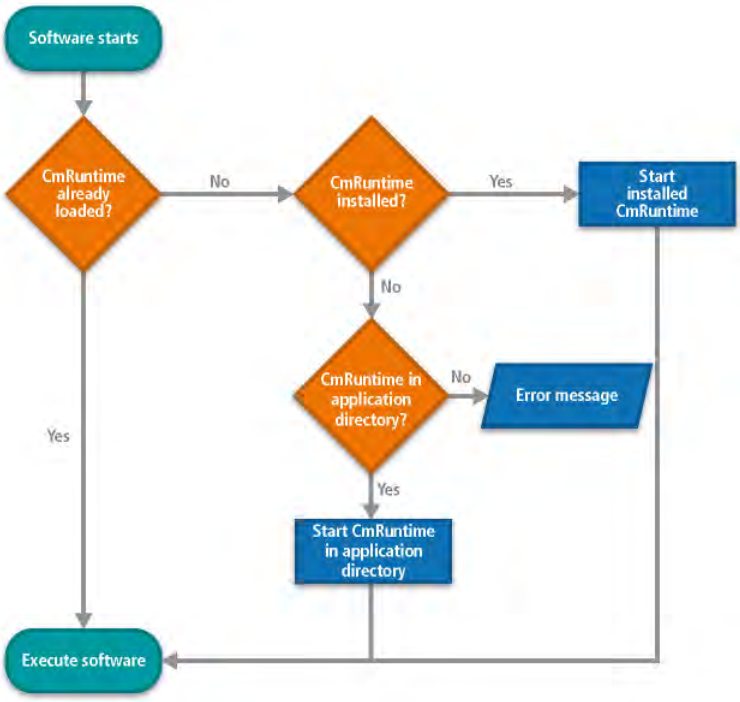


Figure 234: Behaviour *CodeMeter License Server*

UseMobileHandling

Starting with version 4.0, processes to exit *CodeMeter License Server* are automated. Enter "`UseMobileHandling=1`" into the `codemeter.ini` 350 file. This entry automatically closes *CodeMeter License Server* and *CodeMeter Control Center* when exiting your application. If several applications run and access *CodeMeter License Server*, *CodeMeter License Server* exits with the last running application.

```
[General]
ExePath=$(CODEMETER_HOME)
CleanUpTimeOut=120
UDPWaitingTime=1000
UDPCachingTime=20
ApiCommunicationMode=2
IsNetworkServer=0
NetworkAccessFsb=0
NetworkPort=22350
NetworkTimeout=100
MaxMessageLen=67108864
BindAddress=0.0.0.0
UseUmsDA=1
CmActDisabled=0
UseMobileHandling=1
```

Figure 235: Excerpt sample `codemeter.ini`

Shared Memory Mode

The *CmDongle* is not addressed by TCP/IP but the communication uses shared memory. Then the *CmDongle* also in cases when TCP/IP is deactivated (default on mobile installation).

Insert in the `codemeter.ini` 350 file the entry "`ApiCommunicationMode=2`".

> Wibu-Systems recommends this setting on mobile installation. For further questions contact Wibu-Systems support.

10.4CodeMeter Copy Installation on Windows

Wibu-Systems does not recommend the shipping of your protected application by simple copying the necessary components. By copying them to a separate application directory, and the simultaneously use of several *CodeMeter* protected applications, this may lead to version conflicts.

An exception exists in the case of complete systems, i.e. no other software vendor uses the *CmDongle*, and no *CodeMeter* protected applications are installed except the own software. For example, in the case of pre-installed computer of cash or central fire alarm system.

In justified single case the *CodeMeter* runtime installation may be also a copy installation of single components. The following table lists an overview of components, status, and description of related files.

> Please note that on non-installing single components some essential operating options and functions are no longer available.

Component	Status	Description
`CodeMeter.exe`	necessary	Executable of *CodeMeter License Server*

Component	Status	Description
		Can be implemented as service with option /i if privileges are sufficient.
`CodeMeter.l**`	optional	Language files for `CodeMeter.exe` If no language file is installed the default language English is available.
`CodeMeterCC.exe`	recommended	Executable of *CodeMeter Control Center*
`CodeMeterCC**.qm`	optional	Language files for *CodeMeter Control Center* If no language file is installed the default language English is available.
`cmu32(64).exe`	recommended	Executable of *cmu* commandline program including support tool *CmDust*. If not installed commandline-based *CodeMeter Control Center* functions are not available. If not installed an access to information gathered by support application *CmDust* is not possible which limits support assistance.
`CmWebAdmin.exe`	recommended	*CodeMeter WebAdmin* localized in several languages.
`WibuCm32(64).dll`	recommended	Includes *CodeMeter* API functions, e.g. support application *CmDust*. The default installation path is [`%\Windows\System32`].
`WibuCm32(64).lXX`	optional	Language files for the `WibuCm32(64).dll`; Installation path: [`\Windows\System32`]. If no language file is installed the default language English is available.
`WibuCmTrigger32(64).dll`	optional	Is required by Microsoft Internet Explorer, e.g. online collection of *CodeMeter License Central.*
`WibuCmTrigger32(64).lXX`	optional	Language files for the `WibuCmTrigger32(64).dll`. If no language files are installed the default language English is available.

11 Advanced CodeMeter Features

The following sections of the Developer Guide describe additional features of the protection and licensing system *CodeMeter*.

11.1 Implicit Firm Item (IFI)

The *Implicit Firm Item* level in a *CmContainer* features the same characteristics as a usual *Firm Item* levels. It simply has some distinct features.

While all other *Firm Item* levels are characterized by the existence of an exclusive *Firm Codes*, which is unique for each licensor, the *Implicit Firm Item* has a *Firm Code* 0.

> This implies that each owner of a *CmContainer* has licensor privileges for the *Implicit Firm Item* level. Thus s/he has read and write access to "his/her" license container.

For this reason, it makes sense to store applications in the *Implicit Firm Item* container to which each owner of a *CmContainer* has access.

> When using the *Implicit Firm Item* level for OEM products, note that the *Product Codes* up to 1000 are reserved for Wibu-Systems. In the case you as a software vendor want to free-of-charge reserve *Product Items* at the *Implicit Firm Item* level, contact Wibu-Systems.

CmDongle Password instead of *Firm Security Box*

The write access to the *Implicit Firm Item* level is special because instead of the *Firm Security Box* - required for all usual *Firm Item* - here the *CmDongle* password is used, the so-called *User Individual Key* (UIK).

> The default password for *CmDongle* is "CodeMeter".

11.2 Enabling

The *CodeMeter* feature *Enabling* allows you by using an access code to activate or deactivate the complete *CmContainer*.

If the *Enabling* refers to the *Implicit Firm Item* (*IFI*) level, the complete *CmContainer* can be activated or deactivated.

> Please note, that a *Universal Firm Code* is still accessible although it has been disabled or temporarily disabled.

This feature may be used, for example, to optionally split the flash memory of a *CmStick* into separate partitions.

Please note, that on *CmDongle* default delivery the access code for this feature in the *IFI* license container corresponds to the *CodeMeter* password, the so-called *User Individual Key (UIK)*, (see above 353). This would enable any owner of the *CmDongle* and the *CodeMeter* password to programm the access control in the case of configured partitions.

> If you as the ISV do not wish so, you must in a first step change this access code (see the separate document "*CodeMeter* Flash Disk Handling: Partitions of the *CodeMeter* Flash Memory" availabe from Wibu-Systems).

The controlled enabling or disabling comprises the interaction of several constituent parts:

- on/off switches (*Enabling Blocks*) and
- mapping (*Lookup*) between *Enabling Blocks* and license entries.

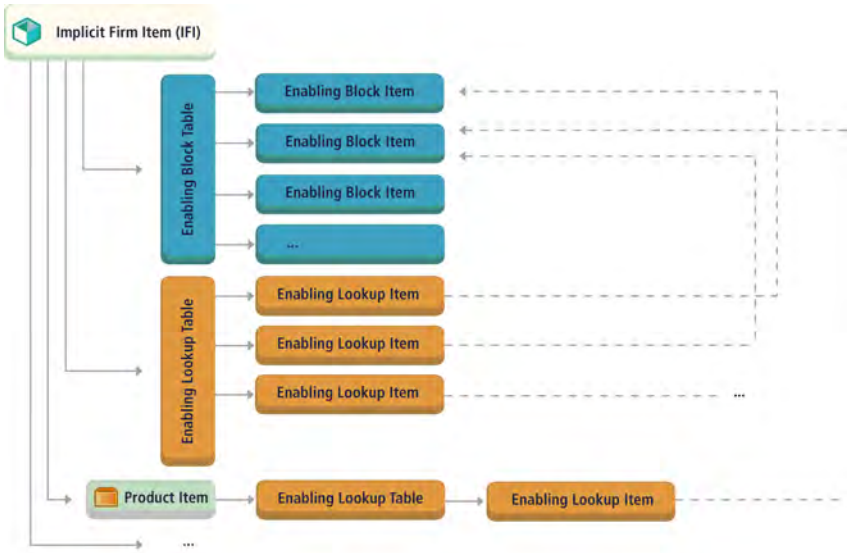


Figure 236: *Enabling* Structure in *CmContainer*

Single *Enabling Blocks* (on/off switches) and *Lookups* (mapping) are pooled in tables. The figure above shows the location and mapping of the different constituents.

11.2.1 Enabling Blocks as On/Off switches

An *Enabling Block* represents a kind of an on/off switch enabling or disabling *Firm Item* levels or license entries.

An *Enabling Block* is enabled or disabled as a whole. It locates as item in a table able to hold up to 31 items. This *Enabling Block Table* may locate at the level of the IFI and other *Firm Item* levels.

The parameter you set in an *Enabling Block* table item comprise information on:

- the access code (*Enabling Access Code*)
- the access type (*Simple PIN* or *Time PIN*).
  For the access type *Time PIN* you have the additional options to describe the *Enabling Block* (Enabling Text) and to specify a time when the activation expires (*Disable Time*).
- the activation mode (*Enabling Mode* with the modes *Enabled*, *Disabled* or *temporary Enabled*).

11.2.2 Access Type - Simple or Time PIN

The access type is defined either as *Simple PIN* comprising only of the *Enabling Access Code* (EAC) or as *Time PIN* additionally holding a describing text and a period defining the valid duration of the *Enabling* feature

Enabling Text

The *Enabling Text* in the case of the access type *Time PIN* allows the labeling of an *Enabling Block*.

Disable Time

The *Disable Time* defines a point in time at which an *Enabling Block* automatically deactivated (disabled). It represents a kind of an expiration date.

The *Disable Time* can assume split-second values up to `December 31 2099, 23:59:59`. In the *CmContainer* the *Disable Time* is compared to the System Time (for details on the synchronization of single time components see page here 357). If the System Time exceeds the *Disable Time*, the complete *Enabling Block* is automatically deactivated.

> A *Disable Time* may also be left out for an *Enabling Block*, i.e. you use the parameter `Disable Time=Never`.

11.2.3 Enabling Mode

Using several modes the activation mode defines whether an *Enabling Block* as a whole is permanently activated, deactivated, or temporarily activated (*Enabling Mode*).

Permanent Enabling Status

The *Permanent Enabling* status defines whether a complete *Enabling Block* is activated or deactivated. It can be set to enabled (activated), disabled (deactivated), or *temporary enabled*.

Temporary Enabling Status

The *Temporary Enabling* status activates or deactivates a complete *Enabling Block* depending on the electrical power supply of a *CmDongle*. It is set to enabled (activated) as long as the *CmDongle* is supplied with electrical power.

If the *CmDongle* is disconnected and replugged, the licensed access requires again the input of the access code. This corresponds to the option "enabled until unplugged" to be set in *CodeMeter Control Center* 389.

The relation between *Permanent Enabling*, *Temporary Enabling* and *Disable Time*

For the relation between the elements *Permanent Enabling* status, *Disable Time*, and *Temporary Enabling* status in a *CmContainer* the following statements are valid:

- in the case of an expired *Disable Time*, always the complete *Enabling Block* is deactivated,
- in the case the *Temporary Enabling* status activated (Global Enabling), it overwrites the *Permanent Enabling* status.

11.2.4 Deleting and Editing Enabling Blocks

For deleting an existing *Enabling Block* not the *Enabling Access Code* required but depending on where it is saved:

- the *CmContainer* password in the case of the *Implicit Firm Item* level and
- the *Firm Security Box* in the remaining cases.

In the *Enabling* process the *Enabling Access Code* controls access options but not the security.

An *Enabling Block* can be deleted only when after a check of the *Enabling Lookup* entries no attachments to *Firm Item* levels or license entries exist.

11.3 Mapping (Lookup) of Enabling Blocks

An *Enabling Block* is not directly saved in the *Firm Item* or license entries. Rather a mapping of the *Enabling Block*, *Firm Item* or license entry takes place.

This mapping process is labeled as *Lookup*. A so-called *Lookup Table* is provided for pooling up to 31 single table items.

Optionally, you attach to or detach items from this table.

Within a *Lookup* table a *Lookup* entry must have only a single attachment to a specific *Enabling Block*. Multiple attachments are prohibited. If the attachment process double-uses an *Enabling Block*, the existing entry is overwritten.

The parameter you set for an *Lookup Table* item next to addressing the license container or the license entry comprise:

- valid access privileges for the activated and deactivated status of an *Enabling Block* (Enabling Level),
- a flag which defines whether activating or deactivating of an *Enabling Block* is mandatory required or not (*required Flag*).

11.3.1 Privileges - Enabling Level

The *Enabling Levels* define tiered privileges for operations valid for activated or deactivated *Implicit Firm*, *Firm Item*, and *Product Item* levels. The following *Enabling Levels* exist:

Enabling Level	Privilege
`Locate`	Valid operations at the level `Locate` allow only the reading of the *Firm* and the *Product Codes* but of no other information.
`Read`	The level `Read` allows the complete reading of all non-hidden information at the *Product Item* level. Not allowed are operations addressing a *Firm Item* or *Product Item* level involving encryption, authentication, or calculation a public key from the saved private key.
`Encrypt`	The level `Encrypt` allows the encryption, authentication and the calculation of a public key – but only when the encryption operation does not decrement an *Unit Counter* and no *Firm Access Counter* at the *Firm Item* level is changed. This level you have to set when the user is to keep the *Unit Counter* reading saved in the *CmContainer*.
`UnitUse`	The level `UnitUse` allows unlimited encryption and decryption, authentication, and the calculation of public keys. However, adding, updating or deleting *Firm Item* or *Product Item* levels are not allowed. This level you have to set to prevent unintentional or unauthorized modifications of local contents in the *CmContainer*.
`Modify`	The level `Modify` allows all operations including modifications at the *Firm Item* or *Product Item* level. No other restrictions exist.

The default setting does not attach an IFI, *Firm Item*, or *Product Items* to an *Enabling Block.* via the *Enabling Lookup*. In this case, no restrictions exist for using these *Items*. This setting is identical to the *Enabling Level Modify*.

11.3.2 Required Flag

The mapping of *Enabling Blocks* using entries of *Lookup* tables may involve several attachment targets at the same time, i.e. different *Firm Item* levels or license entries. In this case of several existing attachment targets, setting the *Required Flag* serves to avoid conflicts in activating or deactivating including different *Enabling Levels*.

In the case, at least one *Required Flag* is set when several attachment targets exist, a logic AND conjunction defines that all settings of attachments having a *Required flag* must match before a defined operation is allowed to access a complete *CmContainer*, a *Firm Item* level, or a license entry.

This is the default setting starting with Firmware Version 1.18. On attaching *Enabling Blocks* using entries of Lookup tables the *Required Flag* is set as default.

When programming the attachment process you are able to explicitly set a *Non-Required Flag*. However, this will have no effect because the default setting ignores *Non-Required Flags* in the case at least one *Required Flag* exists (logical OR conjunction). This is because of the global enabling settings concerning the complete *CmContainer*. In the case you would like to change the global *Enabling* for own purposes, please contact Wibu-Systems Support.

The following reference shows you which *CodeMeter* tools and interfaces you use for *Enabling* operations.

Enabling Block Options	
CmBoxPgm	Create, edit and delete *Enabling Blocks*
Core API	Enabling options
Programming API	Call corresponding classes

11.4Using Own Keys

Together with your *Firm Security Box* you received an initial *Firm Key*. However, in the case you feel a higher security need, and want to define the *Firm Key* for yourself, you are free to do so.

However, when changing the initial value of the *Firm Key* you must ensure that you very safely store this *Firm Key* . In the case you lose this key, even Wibu-Systems is not able to restore this

Hidden and Secret Data

Moreover, at the *Product Item* level you have also the alternative to replace the *Firm Key* by own keys valid for single license entries. These keys you store either in a *Secret Data* or *Hidden Data* field.

However, seen from the perspective of the *CodeMeter* security architecture, this alternative does not provide additional security.

For example, the *Secret Data* field has the same security standard as the *Firm Key*, i.e. it can be read out only. In the case you already use an individual *Firm Key*, then this alternative does not yield additional security.

As the screenshot below from *CodeMeter API Guide* shows, then you have the choice, alternatively to the *Firm Key*, to select a *Secret Data*, or *Hidden Data* field.

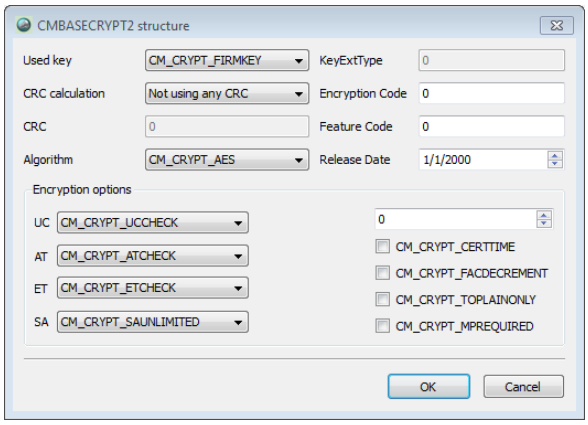


Figure 237: Encryption Alternatives

Hidden and Secret Data

For the symmetric encryption and decryption – i.e. the same key exists in the *Hidden* or *Secret Data* field and in the *CmContainer* – you have the option to encrypt and decrypt using AES, either by AES indirect with a minimum of 16 byte (CBC recommended), or by AES direct with exactly 16 bytes).

An application scenario could be the security-relevant separation of different user-groups of an application, where encryption and decryption operations work with different *Secret* or *Hidden Data* fields. Then a contractor is able to use an application which separately forwards orders to different agents, which in turn cannot access order data of other agents. This provides additional data security. Or you want to ensure that the communication between different technical devices (telephones, fire control center) to which a *CmContainer* is connected, is possible only with specific devices holding identical keys. Then the use of *Secret* or *Hidden Data* fields makes perfect sense.

Figure 238: Application scenario: *Secret Data*, *Hidden Data*

Moreover, you have the option to directly encrypt and decrypt *Secret* or *Hidden Data* fields using the "AES direct" algorithm (see Figure below). his option makes sense, for example, when you want to execute calculations within a protected software but outside the *CmContainer*. This encryption then takes place without the complete *CodeMeter* key derivation 44 , and only the parameters *Firm Key* and *Black Key* are encrypted or decrypted, i.e. without the visible parameters *Firm Code*, *Product Code*, *Feature Code* and without the *Encryption Code*.

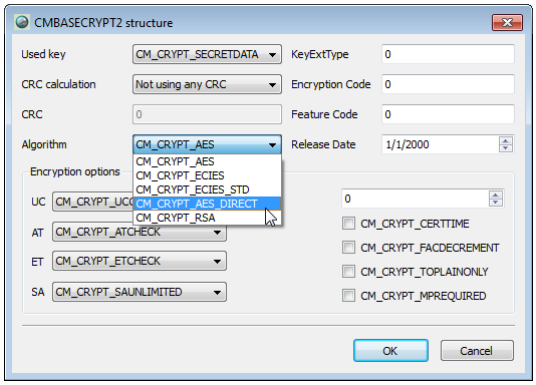


Figure 239: AES direct for *Secret Data*, *Hidden Data* direct encryption

Please contact Wibu-Systems support for further questions.

Asymmetric Encryption

Next to symmetric encryption *CodeMeter* also provides the option to asymmetrically encrypt or decrypt data using private and public keys, and to generate and verify signatures for authentication.

Again you can use own keys stored in *Secret* and *Hidden Data* fields. As it is valid for the *Firm Key*, when encrypting using the *Elliptic Curves Cryptography* (ECC) algorithm the complete 32 bytes are used as a private key to calculate an ECDSA signature.

The public key matching this private key then is calculated using the *CmContainer* and is subsequently verified.

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).

CodeMeter API Guide 300 provides you the API commands and function blocks required: Authentication API 298 , Encryption API 298 , Blocks 298 for executing various encryption and decryption operations.

Please contact Wibu-Systems support for further questions.

11.5Time Server: System Times and Certified Time

Time references in *CodeMeter* play a vital role in a variety of license models, especially when the *Product Item* options Activation, Expiration Time or *Usage Period* are involved, but also in other respects.

There exist several time references which are stored in each *CmContainer*. In sum, they ensure a scheduled and safe use of time-limited software licenses. The meaning of the different times references, and how and when they change is described below. You find the current time references of each individual *CmContainer* in *CodeMeter WebAdmin* on the page **"Container | CmContainer Info** 407 **"**.

Since in a strict sense, the *CmContainer* has no conventional real time clock, it is replaced by a more fail safe and manipulation safe check mechanism:

Every PC comes with an internal clock, whether running Windows, macOS, or Linux. But it's easy to change the computer's system time either forward or backward. Software that enforces time-based licensing based solely on the operating system's time can be easily fooled. If the user's subscription ends on December 31, he can set his system clock back to November or October and get more usage out of his software, in violation of the terms of the license. So clearly something more un-crackable is needed. One possibility is to use a battery-powered clock in a dongle. But what happens when the battery is dead? How safe is a clock with a battery? Another possibility is to use an NTP server (Network Time Protocol) over the Internet. This raises the question of how to recognize and prevent the use of a manipulated NTP server, and what happens if the customer is not online all the time.

How *CodeMeter* knows what time it is?

Each *CmDongle* has a separate clock, located in the internal smartcard chip. This is called the **CodeMeter System Time** (note: this is not the same as the system time of the computer). For *CodeMeter* this is the only valid time. An encryption or decryption can only be made if the *Expiration Time* of the license has not been reached or exceeded in this internal clock.

To put the clock in the smartcard chip has an unbeatable advantage: it's almost impossible to manipulate. A clock placed in flash memory, as found in some other dongles, can be manipulated by a hobbyist with little effort. Unfortunately, the clock in the smartcard chip also has a disadvantage: it only works when the *CmDongle* is connected and has power.

The *CodeMeter* clock stops as soon as the *CmDongle* is unplugged or the computer is turned off. At the next power-on, either when you plug in the *CmDongle* or turn on your computer, the *CodeMeter* system time is synchronized with the time of the computer (**PC System Time**). But only to a later time (i.e. in the future), never to an earlier (past) time. If this is not possible, the *CodeMeter* System Time starts from the last stored time. The *CodeMeter* System Time only advances forward into the future and cannot be reset to the past by the end user. Because it does not rely on a battery, the *CodeMeter* time system is always available to the application, unlike a dongle with a dead clock battery.

Certified Time

In many cases, the accuracy and security of the internal clock is sufficient. For all other cases Wibu-Systems provides the ability to synchronize the internal clock with one of the Wibu-Systems Time Servers. The Wibu-Systems Time Servers get their time similarly to a NTP server from multiple trusted sources (atomic clocks, for example), but also provide a protected channel for the transmission of this time into the *CmContainer*. Manipulation of the transfer or faking a time server is impossible.

When synchronizing the *CodeMeter* System Time with a Wibu-Systems Time Server, the internal clock is set to the current date. In addition, this time is stored as a timestamp in the CmContainer. This timestamp is referred to as **Certified Time**. This time stamp is digitally signed by the Time Server and therefore cannot be manipulated.

What if the *CodeMeter* System Time gets set far into the future? This might be by accident or if you need to do some testing with the date in the future. The PC clock will never set the *CodeMeter* System Time backwards. It can only be corrected by the collection of a new Certified Time from the Time Server without intervention by the ISV.

Time Options

To use the *CodeMeter* System Time, you do not need to implement anything. This is automatically done by *CodeMeter*. If the license expires, whether by *Expiration Time* or *Usage Period*, then the software cannot be decrypted and will not start. If the license is still valid, or it has no *Expiration Time*, then the software runs. The use of the time server is an option that you can use as an additional safeguard. As a software developer, you have some options when setting up time-based licensing:

- Require a synchronization of the *CodeMeter* System Time with a time server since the last power up of the *CmContainer*
- Check if the last synchronization with a time server is not older than xx hours
- Try to connect to a time server, but software always starts regardless
- Do not require the application to ever connect to a time server.

See for example *AxProtector* encryption: Runtime settings | Advanced runtime settings):

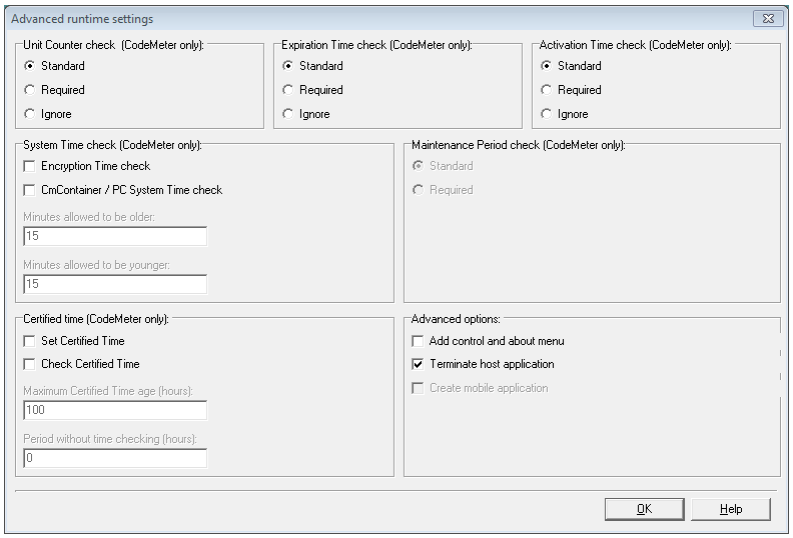


Figure 240: *AxProtector* - "Advanced Runtime Settings"

Times in *CodeMeter WebAdmin*

In *CodeMeter WebAdmin*, you see the *CodeMeter* System Time, PC System Time, and the Certified Time.

Figure 241: *CodeMeter WebAdmin* – "Content | *CmContainer*"

What about *CmActLicense*

CmActLicense uses time the same way *CmDongle* does. In *CmActLicense*, each license file has its own dedicated clock running towards the future. In contrast to *CmDongle* the last time is not stored in secure hardware, but encrypted and hidden on the computer.

In the case of using a previous copy of an older license file (not yet expired license), the check against the hidden time information fails and *CodeMeter* recognizes the fake. Like the *CmDongle*, you can't turn back the clock by changing the OS time on your PC.

CmStick/T

In a few cases, for example, if the licenses are used only rarely and for a short time, a continuously running clock is desirable, even when the *CmDongle* is not plugged in. To meet this need, Wibu-Systems provides the *CmStick/T*, which contains a battery. With the battery, the power off time is bridged. The next time the *CmDongle* is plugged in, this time is used as another source, like the time on the PC, to set the *CodeMeter* System Time. The concept of the secure clock on chip is therefore retained. If the battery is tampered with or fails, you still have the basic protections of system time and certified time listed above.

11.6 Locking a CmContainer

There exist several scenarios in which the licensor is interested in locking the use of a *CmContainer*. The locking can refer to single *Firm Item* levels or to complete *CmContainer*.

Locking a Firm Item

You lock a *Firm Item* level if:

- a manipulation attempt has been detected from within the software,
- the *CmDongle* is reported as lost or stolen,
- a specific licensee is prohibited to use the software, e.g. because there are late payment in the case of pay-per-use licenses.

Locking from within the Software

Locking a *Firm Item* level from within the software is done by the interaction of anti-debugging mechanisms and the *Firm Access Counter (FAC)*.

Firm Access Counter (FAC)

The *Firm Access Counter (FAC)* locates at the *Firm Item* level of a *CmContainer*. This counter allows you to control whether a *Firm Item* level can be used for encryption or decryption operations or not. 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`.

> If you want to use the FAC for hardware locking from within your software, you need to program this value to another value (recommended 1). Otherwise the hardware locking mechanism is deactivated. When the FAC has a value of 0 the *Firm Item* is locked.

In *AxProtector* this mechanism is implemented in the "Security Options" input window by activating the "Activates locking of hardware" option.

Using *Software Protection API* WUPI you implement this by the function ***WupiCheckDebugger***.

In *Core API* the function ***CmCrypt2*** provides the **Fac_Decrement** option which decrements the FAC by a value of 1 (setting the switch `CM_CRYPT_FACDECREMENT` within the `CMBASECRYPT2` structure).

If the software detects a manipulation attempt, a locking sequence is sent which decrements the FAC by the defined value. If the FAC reaches the value of 0, this license container at the *Firm Item* level is locked for further use. However, not the complete *CmContainer* is locked. In the case of *CmDongles* only the licenses which locate in the license container of the respective licensor. The user is still able to use software licenses of other licensors.

By remote programming the licensor is able to set the FAC to a higher value, and thus unlock the locked license container at the *Firm Item* level.

Theft or Losing - Individual Blacklist

When a *CmDongle* is reported as lost or stolen, the licensor has the option to create a separate individual list holding these reported *CmContainer*.

On the next update of the licensed software, in these reported *CmDongles* the *Firm Access Counter* is set to a value of `0`. In the case these *CmDongles* should be recovered, or eventually pending invoices paid, again by remote programming the FAC value can be increased and the locking is revoked.

> Wibu-Systems recommends the creation of such a list.

The `LicenseLock.log` file

The file `LicenseLock-*.log` is created, if the *Firm Access Counter* has been modified according to *AxProtector* encryption settings and the *Firm Access Counter* has reached a value of **0**.

The information is saved on a case-by-case basis and not appended to a single file.

By default, the log file is written on Windows into the directory `C:\ProgramData\CodeMeter\Logs` , on macOS into `/Library/Logs/CodeMeter` and on Linux into `/var/log/CodeMeter`.

The name of the file follows the pattern:

`LicenseLock-YYYY-MM-DD-hhmmss-TimeStamp`

Parameter	Description
`YYYY-MM-DD`	year-month-day specification.
`hhmmss`	current time of the machine on which the license is available.
`TimeStamp`	milliseconds value since `Thu Jan 1 00:00:00 1970` (64-bit value without padded zeros).

The file is partly plaintext partly encrypted.

Wibu-Systems analyzes the encrypted information. The software vendor then is informed on the conclusions drawn from reasons and trigger for and of the locking and is able to take appropriate action.

Locking the complete *CmDongle*

The locking of a complete *CmDongle* is possible if a *CmDongle* is reported as lost or stolen.

Then the licensor has the option to globally lock the complete *CmDongle* via Wibu-Systems.

> This process is exclusively managed online.

The locking is managed by the use of the Wibu-Systems Time Server and the *Certified Time* update of a *CmContainer*. This process involves a global Wibu-Systems blacklist holding the reported *CmDongles* to be locked the next time when a *Certified Time* update is requested. You are also able to integrate this update request in the licensed software. It requires a licensee to regularly access the Wibu-Systems Time Server for a *Certified Time* update.

Then Wibu-Systems locks the respective *CmDongle* if an update request is sent. Naturally, Wibu-Systems implements this only for *CmDongles* for which an unique identity is ensured.

> Locking a *CmDongle* this way is irreversible.

11.7 Backup of CmDongle Content

Backup Mechanism

CodeMeter stores all licenses into the *CmDongle*. Thus the hardware has a special value defined by the sum of the prices paid for software licenses located in the *CmDongle*. When a *CmDongle* is lost or stolen also this value is lost. This can mean a great loss for the owner of the *CmDongle*, but also for the owner of the licenses. Thus *CodeMeter* provides a backup mechanism which writes and saves the contents of a *CmDongle* in a separate binary `*.wbb` file on the PC.

Creating a backup

CodeMeter WebAdmin allows you to specify the location to which this file is saved, and by which backup intervals. By default, a backup is created every 24 hours.

> This file is encrypted and is attempt safe and manipulation safe stored.

This backup file holds all license information from the *CodeMeter* SmartCard memory – with the exception of the *Secret Data* field - that is:

- the complete *CmDongle* information structure (serial number, serial key, *CmDongle* version, etc.),
- the *Implicit Firm Item* level, and
- the contents of all *Firm Item* level.

Importing a backup

Currently, *CodeMeter WebAdmin* however supports only the data restoring of the *Firm Item* level with the *Firm Code* 0, i.e. the *Implicit Firm Item*. This allows to transfer the saved data into another *CmDongle*, as long as the second *CmDongle* uses the same *CodeMeter* Password (*User Individual Key*). For restoring the other data at the other *Firm Item* and *Product Item* levels currently no separate *CodeMeter* tool exists.

However, in most cases, software vendors log their own histories of programming operations for *CmDongles*, or use other *CodeMeter* tools in a way that an analysis of programming operations is possible.

Sending to Wibu-Systems

In the case a *CmDongle* is lost and a backup file has been produced, and the software vendor wants to read out important information - for example, the verification of specific software action by a *Unit Counter* status, etc. - the backup file has to be send to Wibu-Systems. Then this file can be manually edited using a matching *Firm Security Box*. Of course, again the *Secret Data* field cannot be read out.

If at the customer the variable data has been locally re-programmed, such as *Usage Period* or *Unit Counter*, a proof of the latest status (days or reading) it can be analyzed by using *CmDongle*-internal time stamps.

11.8CodeMeter in a Wide Area Network (WAN)

By default, the licensing system *CodeMeter* supports the access to licenses stored on a network server based on the communication between two instances of the *CodeMeter* runtime environment (*CodeMeter License Server*).

In the case of a local network (Local Area Network, LAN), the communication takes place between a local *CodeMeter License Server* and a network *CodeMeter License Server* via the TCP/IP protocol and the communication type CmLAN.

Since *CodeMeter* Version 5.0 the communication type *CmWAN* for Wide Area Networks, WAN is available. A WAN is a computer network which in contrast to a LAN may be geographically dispersed and is not limited in the number of connected computer.

In the case of a Wide Area Network (WAN), then the communication takes places between *CodeMeter License Server* on clients and a network *CodeMeter License Server* via the HTTPS protocol and the communication type CmWAN.

The following sections give an overview of a WAN infrastructure 361 using CmWAN and describe the necessary steps required for *CodeMeter* sided implementing 362.

11.8.1 WAN Infrastructure

Using the *CodeMeter* communication type CmWAN in a WAN requires a special infrastructure. An essential role plays a proxy server installed in the demilitarized zone (DMZ) behind a firewall.

The reverse proxy serves as a communication turntable for *CodeMeter* clients accessing licenses stored on an internal server on which also a *CodeMeter License Server* runs. Here a *CodeMeter* client always communicates with the reverse proxy via a TLS/SSL-secured and encrypted connection (HTTPS). This single access point connects the *CodeMeter* clients not directly with the internal server and the server's identity is not visible.

At the communication level, the reverse proxy forwards the HTTPS request as HTTP request to the *CodeMeter License Server* on the server. Conversely, the reverse proxy returns the HTTP response of the *CodeMeter License Server* on the server back to the *CodeMeter* clients secured by HTTPS.

In addition, the reverse proxy can perform authentication tasks. Then a client must authenticate with the proxy server (user, password) and/or the reverse proxy issues a client certificate which is used by the *CodeMeter* client for authentication.

> Currently, *CodeMeter* supports Digest Authentication for client access. It is planned to integrate the use of client certificates in future *CodeMeter* versions.

The following figure sketches this infrastructure.

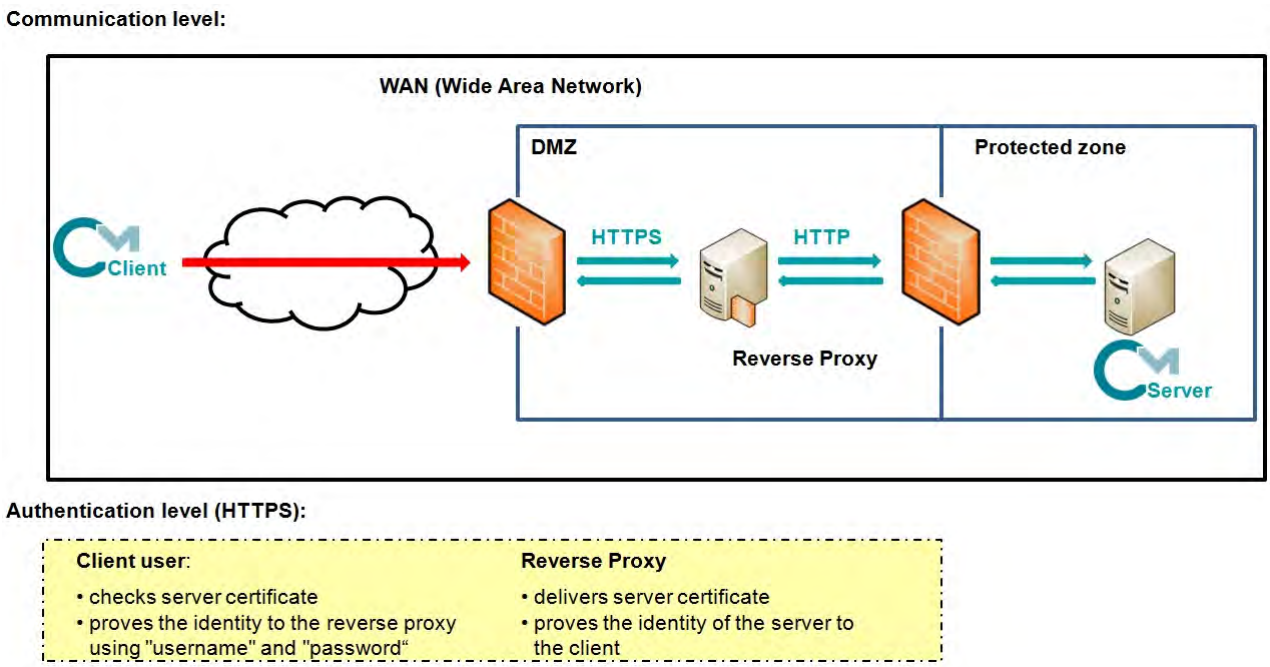


Figure 242: CmWAN: Network communication and authentication

Installing and configuring the WAN including the reverse proxy is not done by Wibu-Systems. This rests with the customer. However, if you require support in installing and configuring, WIBU Professional Services is glad to assist you.

If testing and using a self-created test certificate at the reverser, please note that you import this certificate as root certiificate on the client. The root certificate the client is to use validating the server certificate must locate in the client system's certificate memory to be valid for the complete system.

Requirements:

Proxy Server	Server on which licenses are stored
support of TLS/SSL-secured connections (HTTPS)	installed *CodeMeter License Server* Minimum Version 5.0
transforming from HTTPS to HTTP and vice versa support of authentication tasks	

11.8.2 CodeMeter-sided Implementation

For using *CmWAN* you have to configure *CodeMeter* in the following areas and the following tools:

- license programming [362] (*CmBoxPgm*)
- license usage via API [363] (*CodeMeter API*)
- license usage via Automatic Encryption [364] (*CodeMeter Encryption Suite*)
- CmWAN network communication [365] (*CodeMeter WebAdmin;* registry or `Server.ini` entries)

11.8.2.1 Programming of licenses (CmBoxPgm)

Firm Security Box-license entry

In order to program licenses for using *CmWAN* you first require a separate license entry [`100021:10000:1`] in your *Firm Security Box* (FSB).

This separate FSB license entry you will receive by Wibu-Systems.

Programming a license entry using *CmBoxPgm*

With the tool *CmBoxPgm* using the *License Quantity* option [322] `wan` you can define for each license entry whether is used with the communication type *CmWAN*.

By default, you find *CmBoxPgm* as executable command line program `cmboxpgm.exe` in the Windows directory "`%\Program Files%\CodeMeter\DevKit\bin`". For other operating systems you find *CmBoxPgm* in the usual directories.

The programming sequence follows the pattern:

```
cmboxpgm.exe /[CmContainer] /f [...] /p[...] /plq<Number>:wan
```

`/[CmContainer]` addresses the *CmContainer* to be programmed (see here[317])

`/f [...] /p[...]` specifies the license entry (*Firm Code/Product Code*)

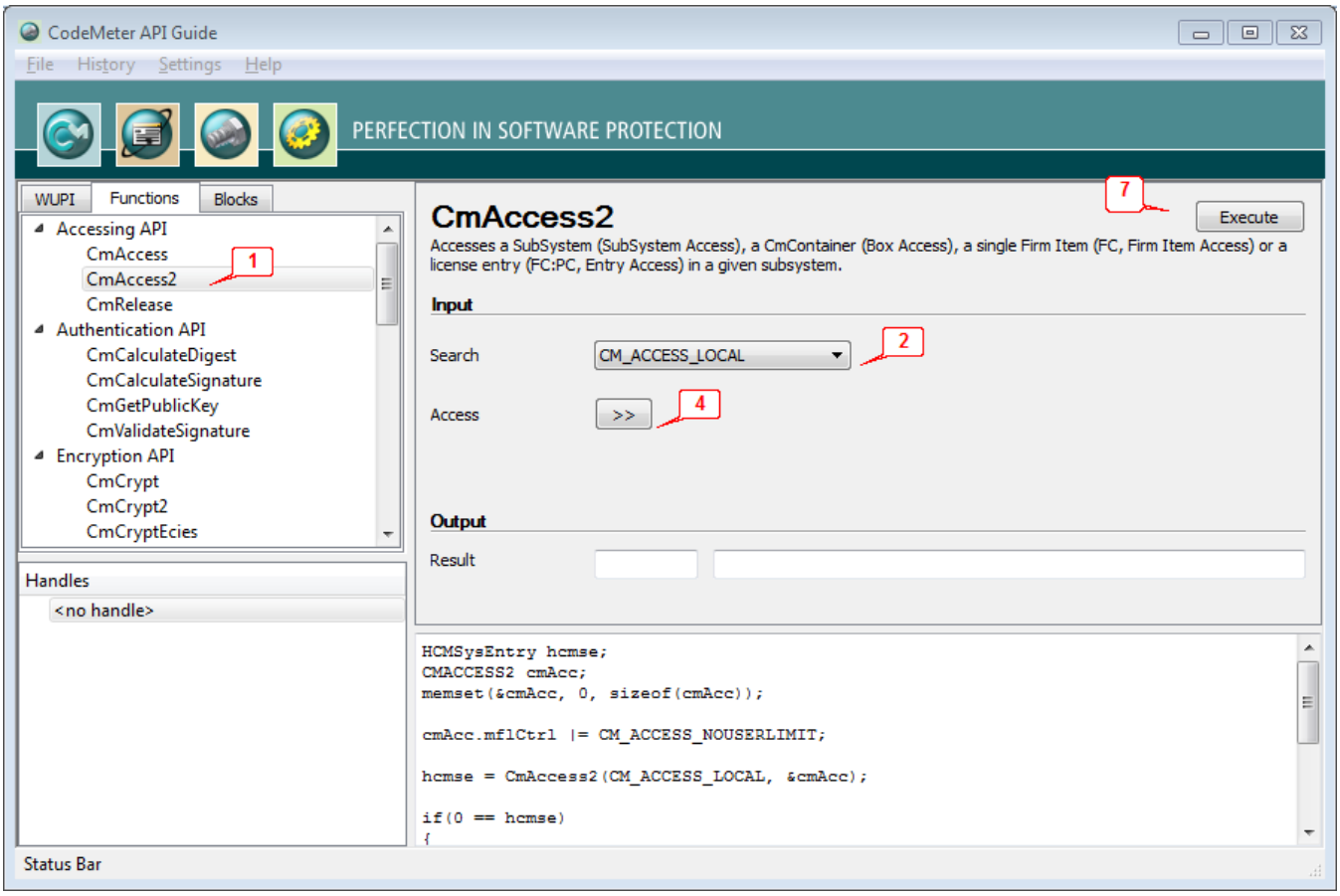
11.8.2.2 License usage via API (CodeMeter API Guide)

In order to use the required license access via the ***CmAccess2*** structure using the tool *CodeMeter API Guide*[300], open the *CodeMeter API Guide* via:

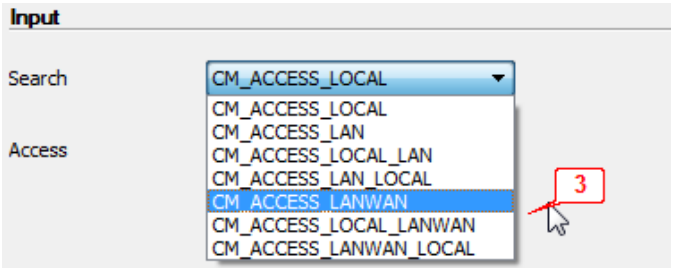
- start menu item **"Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt"**.
- "Windows" key to open Start screen | Type "CodeMeter Command Prompt" | Press "Enter" key.

and then proceed as follows:

1. Select **CmAccess2** item via tab "**Functions**" and item **Accessing API**.
   You get more information on this function by pressing the F1 key.



2. Open **search** list.
3. Select a WAN item on the list.



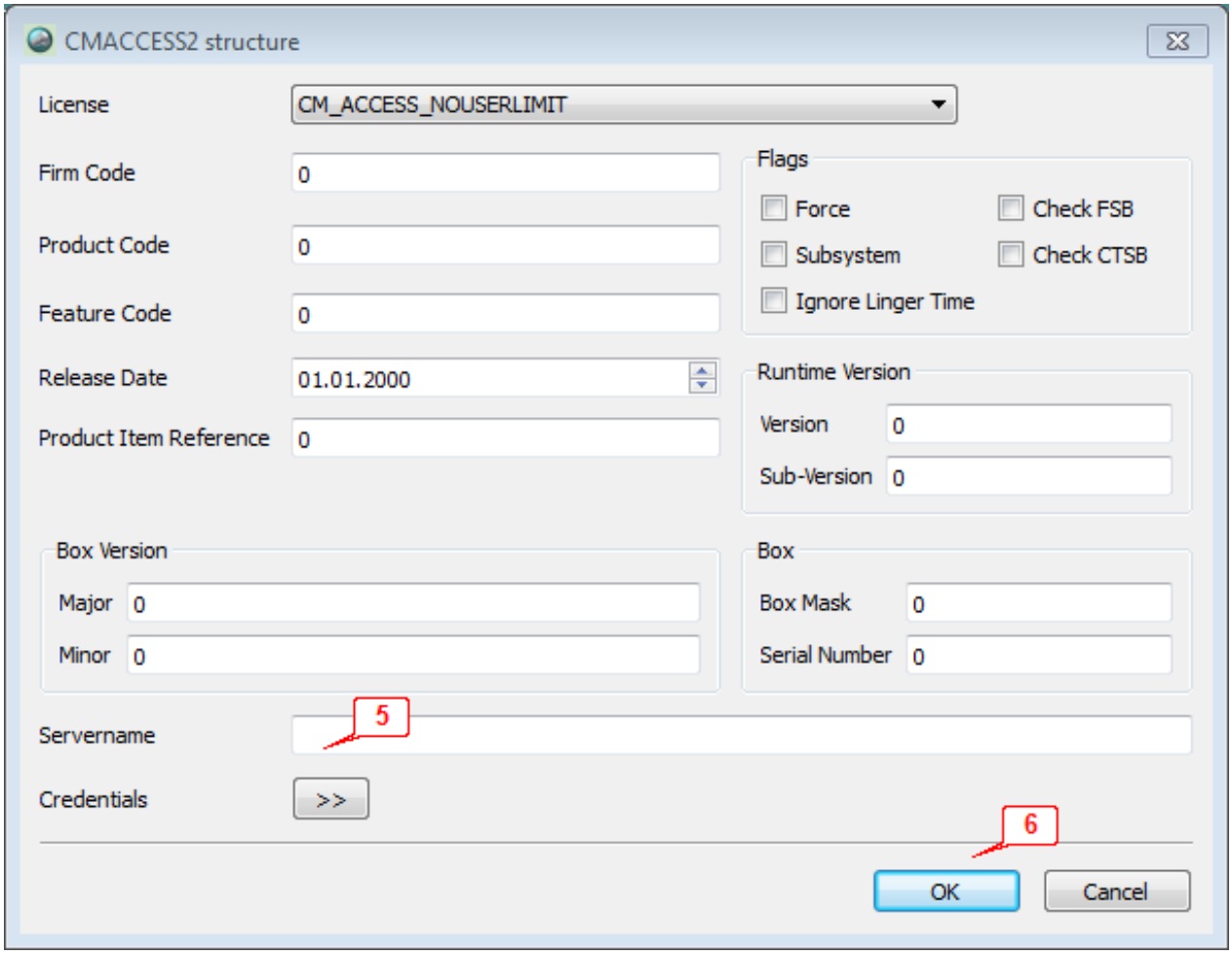
In general, *CmWAN* is handled similar to a usual *CodeMeter* network access. The following WAN flags exist:

- *CM_ACCESS_LANWAN*:
  If a *CmWAN* address is specified in the server search list or in the member `mszServername` of the ***CmAccess2*** structure, the address is used for a license access on a network.
- *CM_ACCESS_LOCAL_LANWAN*:
  Behavior as specified above according to the usual search order: first local, then CmLAN and CmWAN according to the server search list.

- *CM_ACCESS_LANWAN_LOCAL*:
  First using the usual search order: first CmLAN and CmWAN according to the server search list, otherwise local.

> If you allow *CmWAN* by using one of the flags, automatically also the access via *CmLAN* is activated. Conversely, ***CmAccess2*** will not involve an *CmWAN* license access, unless none of the three flags is specified.

4. Click the button "**>>**" to open the **CMACCESS2 structure**.
   Specify or activate the desired license details and parameter.



5. Optionally, specify the address (**Servername**) for the *CodeMeter* runtime environment (*CodeMeter License Server*) on the server.
   The address pattern is as follows:
   `https://user1:password1@reverse proxy address/servername`
   for example `https://user1:password1@cmwantest1.wibu.local/cmwan/test`

> Please note that you must set the prefix `https://`.

> Currently, *CodeMeter* supports Digest Authentication for client access. It is planned to integrate the use of client and server certificates in future *CodeMeter* versions.

6. Click the button **"OK"** to save the parameter of the CmWAN license access.
7. Click the button **"Execute"**.
8. Copy the generated content of the output window into the source code of the application to be protected.

11.8.2.3 license usage via API Automatic Encryption (CodeMeter Encryption Suite)

Using *CmWAN* must explicitly be activated when encrypting the application to be protected. Use the commandline variant of the tool AxProtector for automatic software protection as follows:

1. Call the respective AxProtector version 263 for the project type to be used.
   The call follows the general pattern:
   `AxProtector call -<options> <path and name of the application to be protected>`
2. Set the option `-s` 265 in the licensing system settings according to the WAN requirements.The following parameter are available:

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).

Please note that once you use WAN automatically LAN is activated since WAN represents an extension of the LAN communication.

11.8.2.4 Configuring CmWAN network communication

For configuring the *CmWAN* network communication two alternative ways are provided: either by using *CodeMeter WebAdmin* 365 or by configuring the profiling 365.

11.8.2.4.1 CodeMeter WebAdmin Configuration

For setting up *CodeMeter*® in a WAN, please proceed as follows:

Configure Server

1. Start *CodeMeter WebAdmin* (see here 401).
2. Navigate to the page "**Configuration | Server** 425 ".
3. Activate the option **Run CmWAN Server** to use the computer in a Wide Area Network (WAN) and allow license accesses.
4. Specify a **CmWAN Port** in the field of the same name.
   Default port for the *CodeMeter*® communication via WAN is 22351.

   You are able to customize this value. In this case, make sure that:

   - all *CodeMeter License Servers* use this port, if *CodeMeter*® protected applications access licenses via WAN.
   - the configured reverse proxy has the same port setting.
4. Click the **"Apply"** button to save the settings.

   When you define network settings, in some cases, this requires the restart of the *CodeMeter*service. However, you do not have to eject or deactivate the *CmContainer*. After you specified the settings you are able to stop and then restart the *CodeMeter* service in *CodeMeter Control Center* 388. For non-Windows operating systems see here 384.

Configure Server Search List

5. Navigate to the page "**Configuration | Basis** 417 ".
6. Use in the **Server Search List** defined *CodeMeter* network (LAN) and WAN server and their order in responding to client requests.
7. Specify the IP address(es) for client requests to the defined *CodeMeter License Server* in the WAN.

   When specifying the IP address(es) please note that you are required to prefix a "`https:\\`"needed for the secured communication with a reverse proxy in the WAN.

   Please use the following pattern on specifying: `https://user1:password1@lc.codemeter.com/cmwantest`

   You edit the server search list by using the respective "**add**", "**remove**" buttons. You can also change the order by using the "**up**" and "**down**" buttons.
8. Click the **"Apply"** button to save the settings.

   When you define network settings, in some cases, this requires the restart of the *CodeMeter* service. However, you do not have to eject or deactivate the *CmContainer*. After you specified the settings you are able to stop and then restart the *CodeMeter* service in *CodeMeter Control Center* 388. For non-Windows operating systems see here 384.

11.8.2.4.2 Profiling in Registry or in server.ini File

By editing registry or `server.ini` (section [General]) entries you are able to configure the *CmWAN* network communication settings.

The following table shows you where for which operating system you find the profiling to configure *CmWAN* network communication settings.

Operating system	Registry / Server Entry
Windows	`HKLM\SOFTWARE\WIBU-SYSTEMS\CodeMeter\Server\CurrentVersion`
Windows	`%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini`

The configuration covers two steps:

- editing existing entries,

- creating new entries for new *CodeMeter* network servers if required and a defining their order in replying to client requests via a server search list.

Edit existing entries

1. Activate *CmWAN* by setting the entry `"IsWanServer"` to a value of `"1"`.
   By default, the value is `"0"` and *CmWAN* deactivated.
2. The default port number for which a *CodeMeter* server accepts *CmWAN* requests is `"22351"`.
   If you wish to define a different port, use the entry `"HttpPort"`.

   Please make sure that this port number you define is **not** the port number for the general network communication defined in the entry `"NetworkPort"`.

Create new entries

For each new *CodeMeter* network server you create a new server entry additionally to the existing ones.

This description refers to the `server.ini` file. In the Windows registry you must create new keys and string entries.

Currently, *CodeMeter* supports digest authentication for client access. It is planned to integrate the use of client certificates in future *CodeMeter* versions.

Navigate to the entry `"ServerSearchList"`. All server entries must exist below this entry.

When creating a new entry for a server using digest authentication you define the parameter `Address`, `User`, and `Password`.

```
[ServerSearchList]
[ServerSearchList\Server1]
Address=https://cmwanserver.example.org
User=user123
Password=...
```

When creating a new entry for a server using server certificate authentication you define the parameter `Address`, `User`, and `Certificate`.

```
[ServerSearchList\Server2]
Address=https://cmwanserver.example.org
User=user456
Certificate=...
```

Once you created the new server entries define a server search list, i.e. the order of these and eventually existing server entries in replying to client requests.

Currently, the server search list has the limitation that an automatic server search is not performed, if one or more *CmWAN* server entries exist. This means, when using *CmWAN* and *CmLAN* all LAN servers must be explicitly listed in order.

11.9 The use of write filters and CmActLicense

CodeMeter supports Independent Software Vendors (ISV) when using write filter mechanisms, for example to protect individual partitions of mass storage devices against unwanted write access. These mechanisms redirect write accesses to a file that is deleted during shutdown. This means that all changes that an application makes to the registry or file system disappear after a restart. The following table gives an overview:

Write filter	Description
EWF (Enhanced Write Filter)	This mechanism "protects" the complete volume without exception and locks the memory access.
FBWF (File Based Write Filter)	This mechanism allows you to define write accesses for specific files or directories and thus exclude individual files/directories from protection.
UWF (Unified Write Filter)	This mechanism intercepts all write attempts for a protected volume and forwards them to a virtual overlay. Using the *UWF* with *CodeMeter* does not require any special handling.

Write filter and *CmActLicense*

These write filter mechanisms directly affect *CmActLicense* licenses. With a *CmActLicense* license, parts of the *CmContainer* are usually stored at different locations on the system. These parts are then all read during loading and must match together for the license to be successfully decrypted.

If the *CmActLicense* information can no longer be read because write filters have deleted the 'memory' of this data distributed on the system, previously valid *CmActLicense* licenses become invalid after a restart.

To use *CmActLicense* licenses all the same, the ISV must:

- **first**, when programming a *CmActLicense* license explicitly agree to the use of write filters by setting the `'ewffbwf'` parameter 330.
  This will only load *CmActLicense* licenses that have been unlocked with this option.

- **secondly**, cause the person installing the system to specify a location in the profiling[376] using the parameter `CmActPath` where the *CmActLicense* data should be stored persistently in order to enable use with the write filter activated.

> Please note that *EWF* should select this location on a non-write-protected partition. With *FBWF*, this can also be an excluded directory on the partition protected with a write filter.

12 Manual

The following parts of this *CodeMeter* Developer Guide on installing and using many of the *CodeMeter* tools are also of interest for the administrator and thus part of a separate section.

12.1 First important Information

First connection of *CmDongle*

Connect your *CmDongle* with a free USB interface of your PC. The light diode of the *CmDongle* alternatively flashes red and green for 1-2 seconds. Your PC shows that a new USB device has been found. *CmDongles* with additional Flash memory, e.g. *CmStick/M*, are able to permanently hold any data on this drive.

> With *CmDongles* with additional flash memory it can happen that the Windows message "Do you want to scan and fix 'XYZ'?" displays.
> This always happens when the connection was disconnected during access and the write process was not completed by closing a special flag at the same time.
> Solution:
> - Using the safely remove option, always log off the *CmDongle* before removing it from the system.
> - Click on "Scan and fix". Please save all important data first!

Alternatively to the mass storage device status, *CmDongles* can also display as HID (Human Interface Device) without a drive status (for more details see here 461).

> *CmDongles* without Flash memory represent virtual drives, i.e. data you save on it will get lost once you disconnect the *CmDongle*!

By default, *CodeMeter License Server* is installed as service (Windows) or as daemon (Linux, macOS) and thus automatically starts on system startup. The behavior at system startup is optimized by using default values and prevents eventually occurring process access conflicts. In the case of problems, please contact Wibu-Systems Support.

If *CodeMeter License Server* should not be active, it can be manually started or stopped 384. The following table shows you start options for different operating systems

Operating System	Menu Control	Name
Windows	[Start | All Programs | CodeMeter | CodeMeter Control Center]	`CodeMeter.exe`
	Press "Windows" key to open Start screen | Type "CodeMeter Control Center" | Press "Enter" key	
macOS	[Programs | CodeMeter | CodeMeter Control Center	`CodeMeterMacX`
Linux	[Applications | System | CodeMeter Control Center] or [Applications | Accessories | CodeMeter Control Center]	`CodeMeterLin`

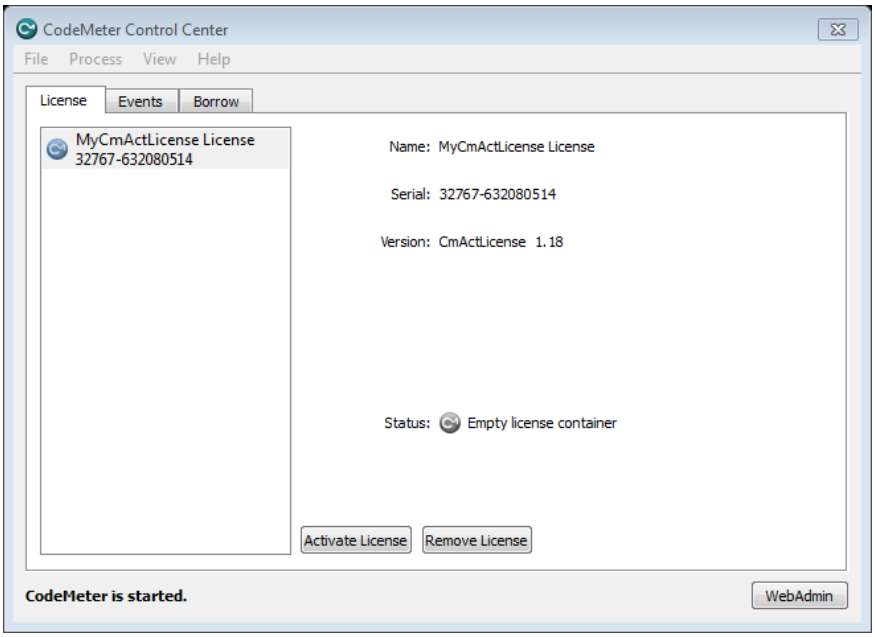
> On Linux (graphic desktop environment KDE 4) eventually connected *CmSticks/BMC* / *CmCards* are not automatically detected.
> If removable devices are connected to Linux systems, they have to be mounted, i.e. making the filesystem on the device accessible. Some desktop environments do this automatically, some do not.
> Check the settings for general automatic mounting and also for the *CmStick/BMC* / *CmCard* using the menu item "System Settings | Hardware | Removable Devices".
> Of course you cannot mount a file system whose underlying device is not connected.
>
> If on Linux *CmSticks/BMC* / *CmCards* cannot be detected, please proceed as follows:
> 1. Start `codemeter` a root user.
> If you use the `systemd-init` system, please enter the following shell commands:
>  ```
>  # mkdir /etc/systemd/system/codemeter.service.d
>  # printf '[Service]\nUser=root\n' > /etc/systemd/system/codemeter.service.d/as-root.conf
>  # systemctl daemon-reload
>  # systemctl restart codemeter.service
>  ```
> If you use the `sysvinit/sysv-rc`, please enter the following shell commands:
>  ```
>  # sed -i '/^USER/cUSER=root' /etc/init.d/codemeter
>  # service codemeter restart
>  ```
> 2. Read-Write mount the *CmStick/BMC* / *CmCard*.
> In the desktop environment either mount the filesystem in the `/etc/fstab` (using the service `udisks2`), or manually mount it.
> Eventually you must create an empty `'codemeter.io'` file: `<mntpnt>/CM-Device/codemtr.io`.
> If the mount takes place after the start of `codemeter`, either use *CodeMeter Control Center* "View | Refresh" - `CTRL+R` or restart using the shell command "`# service codemeter reload`".

CmActLicense the software- and activation-based *CodeMeter* variant requires no hardware token. Rather *CmActLicense* licenses are bound to hardware properties of the PC on which they are accessed.

> ⚠ Please make sure you activate a *CmActLicense* license only on the PC for which you want to use the license.

Before you are able to activate *CmActLicense* licenses for your PC you require a separate file you obtain from your software vendor. This licenses information file corresponds to an empty license container. It serves to collect hardware properties of your PC as a kind of 'finger print' for the subsequent activation. Please proceed as follows:

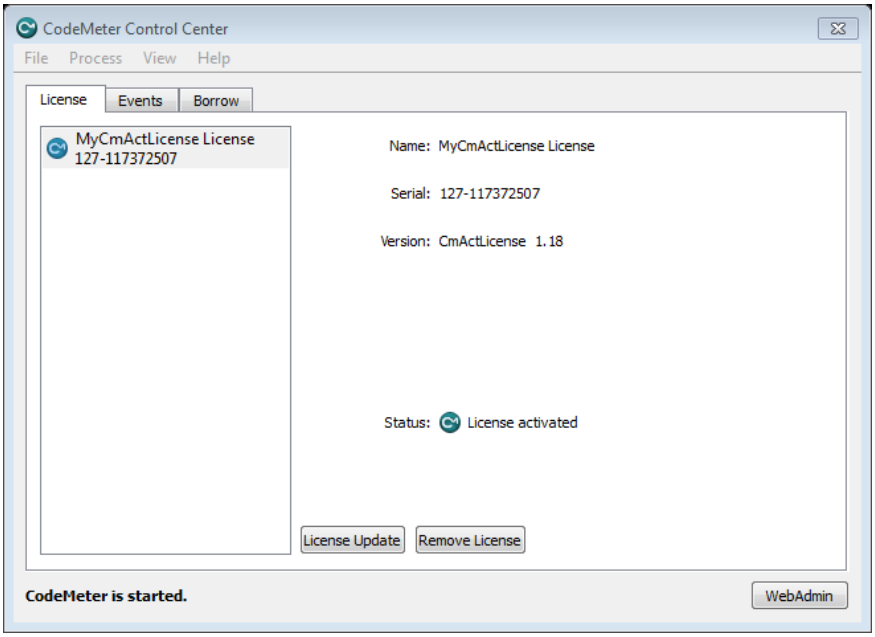
1. Drag & drop the `*.wbb` file, e.g. `MyCmActLicense.wbb`, you received from your software vendor onto *CodeMeter Control Center.*



   The **"Status"** field shows that is file is only an empty license container and not a license. At the same time, the *CodeMeter* symbol changes to red.

2. Click the **"Activate License"** button to create a license request file (see here 394) and to send it to your software vendor. Subsequently, your software vendor will send you a license update file.

3. Drag&drop the `*.wbb` file, e.g. `MyCmActLicense.WibuCmLIF`, you received from your software vendor onto *CodeMeter Control Center.*



   The **"Status"** field shows that the license has been activated. At the same time, the license has a serial number, and the *CodeMeter* symbol has switched to activated status.

CodeMeter FAQ

A comprehensive FAQ area on *CodeMeter* and on other additional products, you will find at our CodeMeter support page.

Please take a first look at the information on the *CodeMeter* support page before you contact our support team. In most cases, you will find quick answers to your questions and problems.

Support

You have several options to contact us:

E-Mail	Writes us an e-Mail at support@wibu.com Please describe your problem in detail and add the file `CmDust-Result.log` created with *CmDust* 450.
Telephone	Contact our *CodeMeter* Hotline at +49-721-93172-15. We are available in Germany (local Baden-Wuerttemberg non-holiday) workdays (Monday through Friday) from 8 a.m. to 5 p.m. 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).

12.2 Installation

The following section contains installing and uninstalling information of the *CodeMeter* for different operating systems.

While installing it is not required that a *CmDongle* is connected to the computer.

- Windows 32-bit/64-bit 370
- macOS 372
- Linux 373

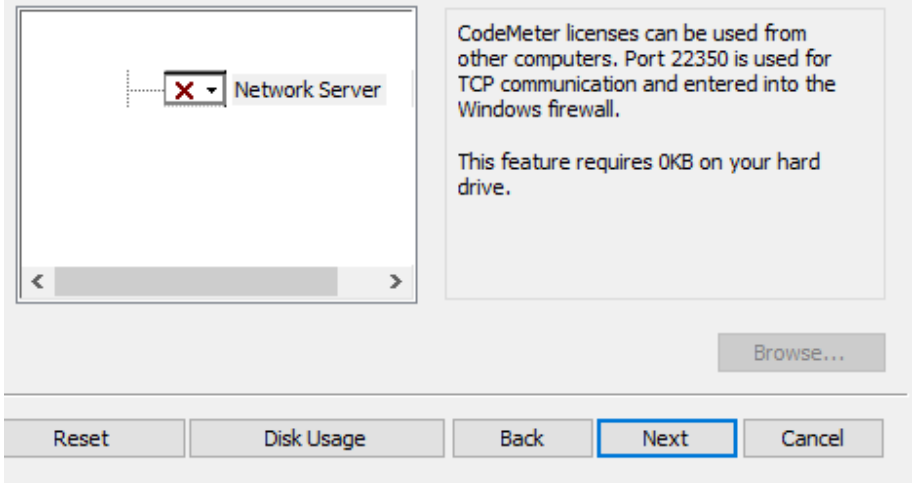
12.2.1 Installation on 32/64-bit Windows

For Windows 32- and 64-bit a *CodeMeter* Runtime Kit installation program is available:

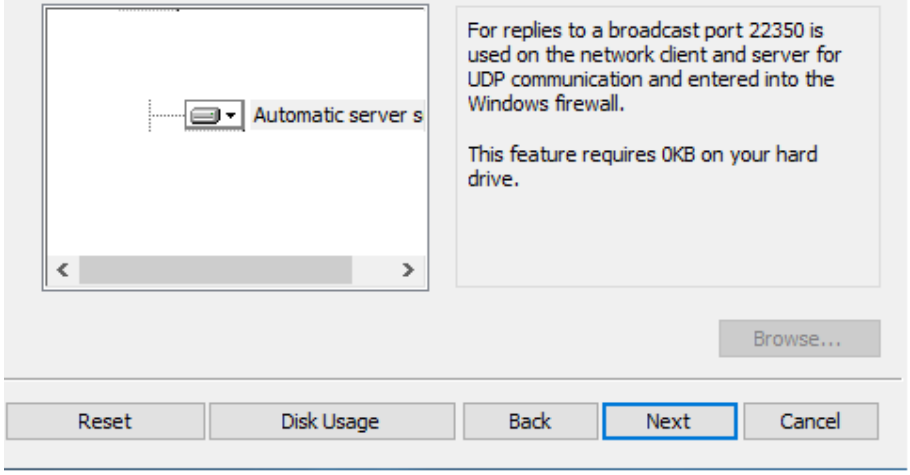
For installing the *CodeMeter* Runtime Kit start the respective installation program and follow the installation wizard.

Network Server and Automatic Server Search

On installing 370 it can be decided whether *CodeMeter License Server* is set up as a server in a network environment and the related TCP port 370 22350 is registered with the Windows firewall. By default, *CodeMeter License Server* is only available locally (local host).



Also during the installation an automatic search of network servers is set as default. This is implemented by a broadcast via UDP (User Datagram Protocol) (it is listened only at server search time and only until the end of the UDP Waiting Time) and for communication the related UDP port 370 22350 is registered with the Windows firewall.



In order to modify these default settings open the installation wizard again. In the following dialog click the button "**Change**" to allow the modifiactions to be made.

12.2.1.1 Installed files on 32/64-bit Windows

The files the *CodeMeter* Runtime Installation Kit installs on your PC you find in the installation directory usually in [`%\Program Files\CodeMeter\Runtime\bin`]).

For 32-bit Windows the following directory structure is valid:

```
%ProgramFiles%
  `--CodeMeter
     |-- Backup (starting with Version 4.30 as Shortcut)
     |-- Logs (starting with Version 4.30 as Shortcut)
     `-- Runtime
         |-- bin
         |   |-- CodeMeter.exe
         |   |-- CodeMeter.l*
         |   |-- CodeMeterCC.exe
         |   |-- CodeMeterCC.l*
         |   |-- CmWebAdmin.exe
         |   |-- CmRmtAct32.*
         |   |-- cmu32.exe
         |   |-- WibuCmId32.*
         |   `-- WibuCmTrigger32.*
         `-- help
              `-- CmUserHelp
%WINDIR%
  `-- System32
      |-- WibuCm32.lxx
      |-- WibuCm32.dll (CodeMeter Calling Driver)
      |-- WibuCmJni.dll
      `-- WibuXpm4J32.dll
%ProgramData%
  `-- CodeMeter
      |-- Backup
      `-- Logs
```

For 64-bit Windows the following directory structure is valid:

```
%ProgramFiles(x86)%
  `--CodeMeter
     |-- Backup (starting with Version 4.30 as Shortcut)
     |-- Logs (starting with Version 4.30 as Shortcut)
     `-- Runtime
         |-- bin
         |   |-- CodeMeter.exe
         |   |-- CodeMeter.l*
         |   |-- CodeMeterCC.exe
         |   |-- CodeMeterCC.l*
         |   |-- CmRmtAct64.*
         |   |-- cmu32.exe
         |   |-- WibuCmId32.*
         |   `-- WibuCmTrigger32.*
         `-- help
             `-- CmUserHelp
%ProgramFiles%
  `--CodeMeter
     `-- Runtime
         `-- bin
             |-- WibuCmId64.dll
             |-- CmWebAdmin.exe
             `-- WibuCmTrigger64.*
%WINDIR%
  |-- SysWOW64
```

```
  |    |-- WibuCm32.lxx
  |    |-- WibuCm32.dll (CodeMeter Calling Driver)
  |    |-- WibuCmJni.dll
  |    `-- WibuXpm4J32.dll
  `-- System32
       |-- WibuCm64.lxx
       |-- WibuCm64.dll (CodeMeter Calling Driver)
       |-- WibuCmJni64.dll
       `-- WibuXpm4J64.dll
%ProgramData%
  `-- CodeMeter
       |-- Backup
       `-- Logs
```

The following table shows an excerpt of installed files:

File	Description
`CodeMeter.exe`	Process of *CodeMeter License Server*
`CodeMeter.l**`	Language files for `CodeMeter.exe`
`CodeMeterCC.exe`	Process of *CodeMeter Control Center*
`CodeMeterCC**.qm`	Language files for *CodeMeter Control Center*
`cmu32(64).exe`	Process of *cmu* commandline program
`CmRmtAct32(64).dll`	Dynamic Link Library (DLL) required by `CodeMeter.exe` for license update.
`CmRmtAct32(64).l**`	Language files for license update.
`CmWebAdmin.exe`	*CodeMeter WebAdmin* in several languages.
`WibuCm32(64).dll`	Includes all *CodeMeter* API functions. This DLL must be installed on all PCs using a *CodeMeter* protected application; installation path: `[\Windows\System32]`.
`WibuCm32(64).lXX`	Language files for `WibuCm32(64).dll`; installation path: `[\Windows\System32]`.
`WibuCmTrigger32(64).dll`	Required by Microsoft Internet Explorer.
`WibuCmTrigger32(64).lXX`	Language files for `WibuCmTrigger32(64).dll`.
`CmUserhelp\*.*`	*CodeMeter* online help in several languages; installation path [`%CodeMeter%\Runtime\help`].

12.2.1.2 Uninstalling on 32/64-bit Windows

1. Select the "**Software**" option in the Windows System control start menu item.
2. Select the "**CodeMeter Runtime Kit**" item and the "**Remove**" option

All *CodeMeter* files as part of an installations package and registry entries are deleted. Only the log and backup directories remain.

12.2.2 Installation on macOS

For macOS a *CodeMeter* Runtime Kit installation programm is available:

File	Description
`CmRuntimeUser.dmg`	installs all required *CodeMeter* runtime components

1. Run the `CmRuntimeUser.dmg` file to install the *CodeMeter* Runtime Kit.
2. Select the file `CmInstall.mpkg` in the new directory `CmRuntime` and follow the instructions of the installation wizard.

12.2.2.1 Installed files on macOS

For macOS the following directory structure is valid:

```
/
|-- Applications
|    |-- CodeMeter.app
|    |    |-- CmUserHelp
|    |    |    `-- ...
|    |    |-- CodeMeterCn.wbb
|    |    |-- CodeMeterDe.wbb
|    |    |--  ...
|    |    |-- CodeMeterMacX
|    |    |-- CodeMeterUs.wbb
|    |    |-- Contents
|    |    |    |-- Info.plist
|    |    |    |-- MacOS
|    |    |    |    |-- CodeMeterCC
|    |    |    |    |-- CodeMeterCC_de.qm
|    |    |    |    |-- CmWebAdmin
|    |    |    |    |-- ...
|    |    |    `-- Resources
|    |    |         |-- CodeMeterCC.icns
|    |    |         |-- com.wibu.CodeMeter.Server.ini
|    |    |         |-- English.lproj
|    |    |         |-- ...
|    |    |         `-- zh_TW.lproj
|    |    `-- PkgInfo
|-- Library
|    |-- Application Support
|    |    `-- CodeMeter
|    |         |-- Backup
```

```
|   |       `-- CmAct
|   |-- Frameworks
|       `-- WibuCmMacX
|   |-- Logs
|   |   `-- CodeMeter
|   |-- Preferences
|   |   `-- com.wibu.CodeMeter.Server.ini (permissions
|   |                                      -rw-rw-rw-)
|   `-- Java
|       `-- Extensions
|           `-- libwibuKJni.jnilib
|-- System
|   `-- Library
|       |-- Extensions
|       |   `-- CmUSBMassStorage.kext
|       |           `-- Resources
|       |               `-- CodeMeter.icns
|       `-- PreferencePanes
|           `-- CodeMeter.prefPane
`-- usr
    `-- bin
        `-- cmu
```

The following table shows an excerpt of installed files:

File	Description
`CodeMeterMacX`	`[Applications/CodeMeter.app]`; *CodeMeter License Server* process.
`CmWebAdmin`	`[Applications/CodeMeter.app]`; *CodeMeter WebAdmin* in several languages.
`CodeMeterUserhelp`	`[Applications/CodeMeter.app/CmUserHelp]`; *CodeMeter* end user help
`CodeMeterCC`	`[Applications/CodeMeter.app/Contents]`; *CodeMeter Control Center*.
`CodeMeterCC**.qm`	`[Applications/CodeMeter.app/Contents/resources]`; Language files for *CodeMeter Control Center*.
`Cmu`	`[usr/bin/cmu]`; *cmu* commandline program.
`WibuCmMacX`	`[Library/Frameworks/WibuCmMacX.framework]`; includes all *CodeMeter* API functions.
`CodeMeterMacX`	`[Library/StartupItems]`; *CodeMeter License Server* startup item.
`libwibuKJini.jnilib`	`[Library/Java/extensions]`; *CodeMeter* Java extension.
`com.wibu.CodeMeter.Server.ini`	`[Library/Preferences]`; includes "Profille Basic Settings" for *CodeMeterMacX*.
`CodeMeter.prefPane`	`[System/Library/PreferencePanes]`; includes the system control for *CodeMeter*MacX."

Starting *WebAdmin*

You start *CodeMeter WebAdmin* in Mac/Linux:

- using the button **Web Admin** in **CodeMeterGUI** tool
- directly in your Internet browser specifying the URLs: `http://localhost:22350` or `http://127.0.0.1:22350`.

12.2.2.2 Uninstalling on macOS

To uninstall the *CodeMeter*® Runtime Kit, proceed as follows:

1. Re-open the `CmRuntimeUser.dmg` disk image.
2. Start in the directory `CmRunTime` the program `CmUninstall.mpkg` and follow the instructions of the wizard (in the commandline enter the following command: `$ sudo installer -pkg /Volumes/CmRuntimeUser/CmUninstall.mpkg -target`. Please note that path specification may vary.).

12.2.3 Installation on Linux

For Linux operating systems different installation packages are available in common formats:

File	Description
`CodeMeter-[CodeMeter-Version].[Package Number].i386.rpm`	Basic 32-bit drivers in RPM format (Red Hat Package Manager Format) (e.g. Suse 9x,)
`CodeMeter-[CodeMeter-Version].[Package Number]_i386.deb`	Basic 32-bit drivers in DEB format gcc3 based (e.g. Debian 3.0, Ubuntu 6.06)
`CodeMeter64-[CodeMeter-Version].[Package Number].x86_d64.rpm`	Driver extension 64-bit in RPM format (Red Hat Package Manager Format) (e.g. Suse, RHEL, FC)
`CodeMeter64-[CodeMeter-Version].[Package Number].amd64.deb`	Driver extension 64-bit in DEB format (e.g. Debian, Ubuntu)

To install *CodeMeter License Server*, proceed as follows:

1. Select the desired installation package, and
2. Install the package as usual, e.g. shell command or respective help programs.

rpm packages: [rpm -ivh CodeMeter-[CodeMeter-Version].[Package Number].i386.rpm]

deb packages: [dpkg -i CodeMeter-[CodeMeter-Version].[Package Number]_i386.deb]

For Linux the following directory structure is valid:

```
/
|-- etc
|   |-- hotplug
|   |   `-- usb
|   |       |-- codemeter.usermap    (obsolete)
|   |       `-- codemeter
|   |-- init.d
|   |   `-- codemeter
|   |-- udev
|   |   `-- rules.d
|   |       `-- 52-codemeter.rules
|   `-- wibu
|       `-- CodeMeter
|           |-- CmFirm.wbc    (permissions -rw-rw-rw-)
|           `-- Server.ini    (permissions -rw-rw-rw-)
|-- usr
|   |-- bin
|   |   |-- CodeMeterCC
|   |   |-- CodeMeterLin
|   |   |-- CmWebAdmin
|   |   |-- cmu
|   |   `-- codemeter-info    (permissions -rwsr-xr-x)
|   |-- lib (for 64-bit systems here the 64-bit libs locate with suffix 64; otherwise the 32-bit libs)
|   |   |-- libWibuCmWebLin[64].so
|   |   |-- libwibucmJNI[64].so
|   |   |-- libwibucmlin[64]-4.so
|   |   `-- libwibucmlin[64].so -> libwibucmlin[64]-4.so
|   |-- lib32 (directory exists only in the 64-bit Installer)
|   |   |-- libWibuCmWebLin.so
|   |   |-- libwibucmJNI.so
|   |   |-- libwibucmlin-4.so
|   |   `-- libwibucmlin.so -> libwibucmlin-4.so
|   `-- share
|       |-- CodeMeter
|       |   |-- CodeMeterCC
|       |   |-- CodeMeterCn.wbb
|       |   |-- CodeMeterDe.wbb
|       |   |-- CodeMeterFr.wbb
|       |   |-- CodeMeterIt.wbb
|       |   |-- CodeMeterJp.wbb
|       |   |-- CodeMeterLin
|       |   |-- CodeMeterUs.wbb
|       |   |-- WibuCmSTrigger.jar
|       |   |-- codemeter.rc     (copy of /etc/init.d/codemeter)
|       |   |-- getpath.class
|       |   `-- libWibuCmWebLin.so -> ../../lib/libWibuCmWebLin.so
|       |-- applications
|       |   `-- codemeter.desktop
|       |-- doc
|       |   `-- CodeMeter
|       |       |-- AppletExample.class
|       |       |-- AppletExample.html
|       |       |-- COPYING
|       |       |-- CmUserHelp
|       |       |   `-- ...
|       |       |-- License.rtf
|       |       `-- README
|       |-- man
|       |   `-- man1
|       |       `-- codemeter-info.1.gz
|       `-- pixmaps
|           `-- codemeter.png
`-- var
    |-- lib
    |     `-- CodeMeter
    |             |-- Backup
    |             |   `-- CM-Backup2-506426-10Aug04-16-40-40.wbb   (Sample)
    |             `-- CmAct
    |                     |-- CmActFI-5010.wbb     (Sample)
    |                     `-- 5010_ABCD-4711.wbb   (Sample)
    `-- log
        `-- CodeMeter
            `-- CodeMeterLin2010-08-04-170622.log   (Sample)
```

12.2.3.1 Uninstalling on Linux

Execute the respective shell command for uninstalling of the *CodeMeter*® Runtime Kit:

- for RPM based distributions, such as, Suse/RedHat/Fedora [rpm -e CodeMeter]
- for DEB based distributions, such as, Debian/Ubuntu [dpkg -r CodeMeter]

12.3 Profiling - CodeMeter License Server settings

The settings with which *CodeMeter License Server* is used are based on so-called 'profiling'. For Windows these settings are stored in the Windows registry, for macOS and Linux `*.ini` files contain this information.

Registry (Windows registry database)

> Please note that extensive problems might occur, if you modify the registry incorrectly. Thus, make sure that you change values very carefully. For added protection, back up the registry before you modify it. Then, you can restore the registry, if a problem occurs.

All settings for the administration of the system and all integrated system services and processes are stored here. Many application programs, such as the Wibu-Systems software, also store their settings here.

In order to check or edit registry entries, please proceed as follows:

1. Open Registry Editor.
   In the search box on the task bar, type **regedit**. Then, select the top result for Registry Editor (Desktop app).
   Alternatively, press and hold or right-click the **Start** button, then select **Run**. Enter `regedit` in the **Open:** box and select **OK**.
2. Advance to the node `[HKEY_LOCAL_MACHINE\SOFTWARE\WIBU-SYSTEMS\CodeMeter\Server\...]`.
   The following values and data types exist:
   - `REG_DWORD`: a binary data type in which 32-bit integer values are stored as 4-byte hexadecimal values.
   - `REG_SZ`: a string of Unicode characters. For names, descriptions, system paths, etc.

You can also create a local Windows `CodeMeter.ini` file holding the stored settings from the registry.

However, be aware that as soon as *CodeMeter License Server* starts, it checks if a `CodeMeter.ini` file exists. If there is a `CodeMeter.ini` file, all default information from the registry is stored there. From this moment on, *CodeMeter License Server* then will use only the information stored in `CodeMeter.ini`.

For creating the `CodeMeter.ini` file, please proceed as follows:

1. Create an empty file named `CodeMeter.ini` in the directory `C:\Program Files (x86)\CodeMeter\Runtime\bin`.
2. Stop and start the *CodeMeter* service in *CodeMeter Control Center* using the "**Action | ...**" navigation items.
   The `CodeMeter.ini` file is completed.

   > Changes made are saved only, if the *CodeMeter License Server* service has been stopped before and then restarted after modifications have been made.

`Server.ini` file

The settings of applications are stored here in configuration files separately for individual programs in respective directories (`*.ini` files).

> Changes made are saved only, if the *CodeMeter License Server* daemon has been stopped before and then restarted after modifications have been made.

On macOS this file locates in:
`/Library/Preferences/com.wibu.CodeMeter.Server.ini`.

On Linux this file locates in:

`/etc/wibu/CodeMeter/Server.ini`

In order to check or edit file entries open the file in a text editor of your choice.

12.3.1 General

`ActionTimeInterval`

specifies the time interval the `ActionHandler` is to scan for an open task to perform.

Value	Description
`[0, 1000]`	Interval between 0 and 1000 milliseconds (default: 10 milliseconds).

`ApiCommunicationMode`

specifies the communication mode between the `WibuCm.dll` library and *CodeMeter License Server* used by the library.

Modes may be combined.

Mode	Description
`1`	platform specific (default)

ApiCommunicationMode

2	shared memory
4	IPv4
8	IPv6

Platform specific default:

Windows	IPv6, IPv4, shared memory
macOS / Linux	IPv6, IPv4
WinCE	IPv4, shared memory

ApiCommunicationModeServer

specifies the communication mode between the `WibuCm.dll` library and *CodeMeter License Server* used by *CodeMeter License Server*.

Modes may be combined.

Mode	Description
1	platform specific (default)
2	shared memory
4	IPv4
8	IPv6

Platform specific default:

Windows	IPv6, IPv4, shared memory
macOS / Linux	IPv6, IPv4
WinCE	IPv4, shared memory

> If `ApiCommunicationMode` is set to a communication mode that is excluded by `ApiCommunicationModeServer`, the communication between DLL and *CodeMeter License Server* will not work. For example, `ApiCommunicationModeServer` is set to `'4'` (IPv4) and `ApiCommunicationMode` is set to `'8'` (Ipv6).

BindAddress

specifies the IP Address of the network adapter used by *CodeMeter License Server*.

Typical examples are `0.0.0.0` which binds to all network adapter (default) or `192.168.0.1`.

BorrowIdentifyByIpAddress

specifies the IP Address (server identification) for a prepared borrowing. `CmGetInfo(CM_GEI_SYSTEM)` returns the IP address as string in the parameter `CMSYSTEM::mszComputerName` and not the full qualified DNS-name.

CleanUpTimeOut

contains a timeout value (in minutes) used by *CodeMeter License Server* to clean up internal handles and threads. The default value is 120 minutes.

Starting with *CodeMeter* Version 6.70a this value only affects LAN handles. It is also possible to specify a `CleanUpTime` at the client by setting it directly using Core API function `CmAccess2`.

CmActPath

contains in the case of applied write filters the location (path information) to which the data of *CmActLicense* is to be saved persistently. If filters are used your software vendor is likely to inform you.

Please do not mix up this with the location of the default *CmActLicense* read and backup mechanism, e.g. `C:\ProgramData\CodeMeter\CmAct`.

> This configuration option applies only to ***Universal Firm Codes*** and must be explicitely set in conjunction with set write filter methods ***EWF*** (Enhanced Write Filter) or ***FBWF*** (File Based Write Filter). This is done using the *CmBoxPgm*-Option `/lopt:ewffbwf`.
>
> For ***EWF***, please note to select the location on a non-write-protected partition. With ***FBWF***, this can also be an excluded directory on the partition protected with a write filter.

`CmInstanceUid`

On starting, *CodeMeter License Server* calculates a unique ID used to discern client computers for station share accesses.

`CmWANPort`

specifies the port address for the *CmWAN* communication used by the server side only. The default value is the port address 22351.

`CmWebSocketApi`

specifies whether the *CmWebSocket* API is used or not.

Value	Description
`0`	disable
`1`	enabled (default)

`EnabledContainerTypes`

specifies the *CmContainer Types* activated. Currently, four flags are defined which can be bit-wise combined.

This setting is used to avoid technical problems on some systems.

Value	Description
`1`	Mass Storage (USB mass storage device class)
`2`	*CmActLicense*
`4`	HID (USB human interface device class)
`8`	*CmCloud*

`EnableWebAdmin`

This parameter is available for embedded systems only (Linux arm and Big Endian) and is ignored for all other platforms.

Value	Description
`0`	*CodeMeter WebAdmin* disabled (default).
`1`	*CodeMeter WebAdmin* enabled.

`ExePath`

specifies the current path information where *CodeMeter License Server* is stored.

On any start, *CodeMeter License Server* saves this path to the parameter allowing applications calling *CodeMeter* on request to start *CodeMeter License Server*.

`HelpFile`

specifies the complete path information and file name of the *CodeMeter* online help file.

`IsCmWANServer`

specifies whether *CodeMeter License Server* is running as a *CmWAN* server or not.

Value	Description
`0`	*CodeMeter License Server* as *CmWAN* server disabled (default).
`1`	*CodeMeter License Server* as *CmWAN* server enabled.

`IsNetworkServer`

specifies whether *CodeMeter License Server* is running as a network server on the network or not.

Value	Description
`0`	*CodeMeter License Server* as network server on the network disabled (default). All network requests from other computers are blocked.
`1`	*CodeMeter License Server* as network server on the network enabled.

`LastLogCleanup`

specifies when all *CodeMeter* `*.log` files have been deleted using a time stamp value.

LogCleanupTimeout

specifies how long the timeout value is for the log file deletion. The default value is 336 hours, i.e. 14 days.

LogCmActDiag

specifies whether *CmActLicense* diagnostic logging is activated or not.

Value	Description
0	*CmActLicense* diagnostic logging disabled.
1	*CmActLicense* diagnostic logging enabled (default).

Logging

specifies whether logging of the console or GUI output is activated or not.

Value	Description
0	Logging disabled (default).
1	Logging enabled.

LogLicenseTracking

specifies whether license tracking is activated or not.

Value	Description
0	License tracking disabled (default).
1	License tracking enabled.

LogLicenseTrackingPath

specifies the location to where the license tracking files are written to.

Default location is a directory `"Licensetracking"` parallel to the `"Logs"` directory specified in parameter `LogPath` 378.

LogLicenseTrackingLogRotationSizeInMb

`[1..3500]`

If a license tracking log file exceeds a size of `x` MB, the rotation starts.

The input range of `x` is from 1 to 3500 MB.

The default value is 1000 MB.

If the specified value is outside the input range, automatically the default value applies.

LogLicenseTrackingLogRotationTimeInMinutes

`[0..525600]`

If the oldest entry of a license tracking log file is older than `n` minutes, the rotation should start.

The default value is 0 hours.

The input range of `n` is from 0 to 525600 minutes (approx. 1 year).

If the specified value is outside the input range, automatically the default value applies.

LogPath

specifies the location where the log files are written to.

Default location is the directory generated platform dependently by the Installer.

Windows	`C:\ProgramData\CodeMeter\Backup\Logs`
macOS	`/Library/Application Support/Logs/CodeMeter`
Linux	`/var/log/CodeMeter`

LtClientsCleanupTime

specifies the period how long License Transfer client data will be stored.

The time is stored in hours to be able to test using small periods. In *CodeMeter WebAdmin* it is stored in days. The default setting is 100 days, i.e. 2400 hours.

MaxMessageLength

specifies the maximum length of TCP/IP requests in bytes. Default is 64 MB which should be sufficient for all known API calls.

NetworkAccessFsb

specifies whether an access to a Firm Security Box (FSB) entry is allowed via network or not.

Value	Description
0	Access to a FSB entry via network is not allowed (default).
1	Access to a FSB entry via network is allowed.

This value will be ignored if an Access Control List[381] (ACL) is used.

NetworkPort

specifies the network port address for the communication. The default value is the port address 22350.

The port 22350 is registered by Wibu-Systems at IANA (Internet Assigned Numbers Authority) and uniquely assigned for the *CodeMeter* communication. For a list of assigned ports see www.iana.org/assignments/port-numbers.

NetworkTimeout

specifies the TCP/IP network timeout value.

Values	Description
0	No timeout will be used.
[40, ∞]	Interval between minimum value of 40 and ∞ milliseconds (Default: 10 milliseconds).

ProxyPassword

specifies the password of the proxy server of the local network.

This must be set, if authentication is required through proxy or firewall.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

ProxyPasswordSecure

specifies the password for authenticated access to the proxy server.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

ProxyPort

specifies the port of the proxy server of the local network.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

ProxyServer

specifies the name of the proxy server of the local network.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

ProxyUser

specifies the user name of the proxy server of the local network.

This must be set, if authentication is required through proxy or firewall.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

StartAlways

specifies whether *CodeMeter License Server* immediately starts without any dialog request.

Value	Description
0	*CodeMeter License Server* will not start immediately.
1	*CodeMeter License Server* will be started immediately (default).

StartDaemon

specifies whether *CodeMeter License Server* will start as daemon on start [Linux only].

Value	Description
0	*CodeMeter License Server* will not start as daemon on startup.
1	*CodeMeter License Server* will start as daemon on startup (default).

SystemStartThreshold

specifies the threshold value in seconds after Windows start [Windows only].

If Windows has not been started since the specified value, the parameter StartAlways 379 is handled in a special way (see WaitForServiceAfterSystemStart 381).

TimeServerTimeout

specifies the timeout for the time server request in seconds. The default value is 20.

TimeServerURL1, TimeServerURL2, …

specifies the URL of the time sever to update the *Certified Time* in the *CmContainer*. The standard installer sets the following default values:

```
TimeServerURL1 = cmtime.codemeter.com
TimeServerURL2 = cmtime.codemeter.fr
TimeServerURL3 = cmtime.codemeter.de
```

UDPCachingTime

specifies the waiting time in seconds after a UDP (User Datagram Protocol) request is retried.

After a UDP search in the network the servers found are stored inside *CodeMeter License Server* for this time. The range of the parameter is `[1, 3600]`. The default value is 20 seconds.

UDPWaitingTime

specifies the waiting time in milliseconds in order to define the period in which a UDP (User Datagram Protocol) request for an existing *CodeMeter License Server* on the network has to reply.

This is the maximum time all other servers can answer to the request. The default value is 1000 milliseconds. The range of this parameter is `[100, 15000]`.

UseMobileHandling

specifies whether *CodeMeter* can be used mobile.

If enabled, a protected application and *CodeMeter License Server* can be started from a mobile disk and will terminate when the last handle of a protected application was released and the protected application has terminated.

Value	Description
0	*CodeMeter* Mobile Handling is disabled.
1	*CodeMeter* Mobile Handling is enabled.

UseMobileVars

specifies whether *CodeMeter* Mobile variables are used or not.

Value	Description
0	*CodeMeter* Mobile variables are disabled and absolute paths remain.
1	*CodeMeter* Mobile variables are enabled (default).

If *CodeMeter* uses a local CodeMeter.ini 375 file, the following variables will be used:

- `$(CODEMETER_HOME)` - contains the absolute path of `CodeMeter.exe`.
- `$(CODEMETER_DRIVE)` - contains the drive letter of the *CmStick*.

For example, the LogPath 378 may be set to `"$(CODEMETER_HOME)\logs"`.

If set to a value `0`, these variables will not be inserted into `CodeMeter.ini` but the absolute paths remain.

UseSystemProxy

specifies whether *CodeMeter* applies the system proxy settings or not.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

`UseSystemProxy`

Value	Description
`0`	*CodeMeter* does not apply the system proxy settings.
`1`	*CodeMeter* applies the system proxy settings.

`UseUmsDA`

specifies whether communication between *CodeMeter License Server* and *CmDongle* is direct or file I/O based.

Value	Description
`0`	Communication is based on file I/O.
`1`	Communication is based on direct access (PassThru) (default).

if the user starting *CodeMeter License Server* has no administrator privileges, then automatically file I/O applies.

`WaitForServiceAfterSystemStart`

specifies how long after system start `CodeMeter.exe` waits in seconds. After that, the `WibuCm.dll` library starts `CodeMeter.exe` as an application [Windows only].

`MaxBorrowDuration`

specifies the global maximum period a license is blocked by an borrowing server in minutes.

Please note, that Prepared License Borrowing supports only Firm Codes smaller than 6000000.

`MaxBorrowQuantity`

specifies the global maximum number of licenses which can be borrowed from an borrowing server.

Please note, that Prepared License Borrowing supports only Firm Codes smaller than 6000000.

12.3.2 AccessControl

`Enabled`

specifies whether Access Control Lists (ACL), i.e. using global and specific access rules for accessing licenses and reserving license access for single staff member or complete Active Directory groups, are used or not.

Value	Description
`0`	Use of ACL disabled (default).
`1`	Use of ACL enabled.

`ActiveDirectoryUpdateInterval`

specifies the interval in minutes that the Active Directory (AD) server request display is updated for user and group information. Changes to Windows Registry requires the CodeMeter License Server to be restarted to apply value changes.

Value	Description
`0`	no update intervals defined; disabled.
`[1, 1440]`	minimum waiting time is 1 minutes, maximum 1440 minutes (1 day) (default is 15 minutes).

12.3.3 Backup

`Interval`

contains a time interval (in hours) when *CodeMeter License Server* should automatically create and save a backup of the connected *CmDongle*(s).

Value	Description
`0`	Automatic backup disabled
`[1, ∞]`	Automatic backup enabled for specified hours (default is `24` hours).

`Path`

contains the path of the location where the backup file of the *CmDongle*(s) is to be saved. The default location for backup files depends on the operating system in use:

Windows	`C:\ProgramData\CodeMeter\Backup`
macOS	`/Library/Application Support/CodeMeter/Backup`
Linux	`/var/lib/CodeMeter/Backup`

`<serial_number>`

contains a time stamp when the last backup of the specified *CmDongle* was created in seconds passed since `01.01.2000`.

`UpdateCertifiedTime`

specifies whether a *Certified Time* update takes place before a backup is executed.

Value	Description
0	*Certified Time* update disabled (default).
1	*Certified Time* update enabled.

12.3.4 HTTP

`DigestAuthentication`

CodeMeter Versions smaller than 6.60:

The parameter specifies, if authentication via "User Name / Password" is required to change settings.

Value	Description
0	Authentication via "User Name / Password" is disabled.
1	Authentication via "User Name / Password" is enabled.

CodeMeter Versions equal to or newer than 6.60:

This parameter specifies, if write authentication (via `WritePassword`[383]) is required to change settings.

Value	Description
0	Write authentication disabled. If disabled, remote write operations, e.g. changing the configuration from a remote host, are not allowed.
1	Write authentication enabled. If enabled, the user needs to enter the `WritePassword`[383] before being able to change configuration or perform other 'writing operations.

`Port`

specifies the port *CodeMeter WebAdmin* is listening for HTTP requests. The default value is 22352.

`PreparedBorrowingConfiguration`

specifies that the configuration of Prepared License Borrowing in *CodeMeter WebAdmin* is enabled or not.

> Please note, that Prepared License Borrowing supports only *Firm Codes* smaller than 6000000.

Value	Description
0	Prepared License Borrowing configuration disabled (default).
1	Prepared License Borrowing configuration enabled. If enabled, the parameters MaxBorrowDuration[381] and MaxBorrowQuantity[381] can be specified.

`ReadAuthenticationEnabled`

specifies whether a read authentication is required or not.

Value	Description
0	Read authentication disabled (default). If disabled, remote write operations, e.g. changing the configuration from a remote host, are not allowed.
1	Read authentication enabled. If enabled, the user needs to enter the `ReadPassword`[383] before being able to see any *CodeMeter WebAdmin* page.

ReadPassword

contains the password required to be entered, if ReadAuthenticationEnabled 382 is set to a value of 1(enabled). The password is encrypted with bcrypt.

RemoteRead

specifies whether it is possible to read from a remote host.

Value	Description
0	Remote reading is disabled.
1	Remote reading is enabled (default).

UserAuthentication

CodeMeter Versions smaller than 6.60:

contains the hashed *CodeMeter WebAdmin* password, if DigestAuthentication 382 is enabled.

CodeMeter Versions equal to or newer than 6.60:

Obsolete parameter.

> Please use WritePassword 383 instead.
> UserAuthentication 383 is read only for transferring the configured password to WritePassword 383 the first time the user enters it.

UserName

contains the *CodeMeter WebAdmin* user name, if DigestAuthentication 382 has been enabled.

WritePassword

contains the password required to be entered, if DigestAuthentication 382 has been enabled. The password is encrypted with bcrypt.

12.3.5 HTTPS

CertificateChainFile

specifies the path to the certificate chain file required for HTTPS communication. The default value is platform dependent:

Windows	C:\ProgramData\CodeMeter\WebAdmin\SelfSignedCert.pem
Linux	/var/lib/CodeMeter/WebAdmin/SelfSignedCert.pem
macOS	/Library/Application Support/CodeMeter/WebAdmin/SelfSignedCert.pem

Enabled

specifies whether HTTPS communication is activated or not.

Value	Description
0	HTTPS is disabled (default).
1	HTTPS enabled.

Port

specifies the port the *CodeMeter WebAdmin* is listening for HTTPS requests. The default value is 22353.

PrivateKeyFile

specifies the path to the private key file required for HTTPS communication. The default value is platform dependent:

Windows	C:\ProgramData\CodeMeter\WebAdmin\key.pem
Linux	/var/lib/CodeMeter/WebAdmin/key.pem
macOS	/Library/Application Support/CodeMeter/WebAdmin/key.pem

12.3.6 ServerSearchList

UseBroadcast

specifies whether automatically licenses on servers are searched for, first locally and then in the network (subnet), i.e. whether a so-called broadcast is performed.

Value	Description
0	Broadcast disabled.
1	Broadcast enabled (`255.255.255.255` is automatically added) (default).

`Server1, Server2, ...`

specifies the access to and order of *CodeMeter* network LAN and WAN (Wide Area Network) servers in separate list entries. By default, a broadcast (`255.255.255.255`) is specified.

For LAN network servers, the IP address or name entry are possible.

```
[ServerSearchList\Server1]
Address=184.45.89.5

[ServerSearchList\Server2]
Address=185.55.78.6
```

For WAN servers next to the required "https:\\" IP address also the 'User' and 'Password' credentials are required. The password will automatically be converted to PasswordSecure.

```
[ServerSearchList\Server3]
Address=https://my.product.com/cmwantest"
PasswordSecure=****
User=user1
```

12.3.7 TripleModeRedundancy

`TmrEnabled`

specifies whether Triple Mode Redundancy (TMR) is active or not.

Only to be used in a TMR Server Setup for using a CodeMeter Backend Server.

Value	Description
0	disable (default)
1	enable

Please note, that if enabled, at the same time, local and shared memory accesses to this *CodeMeter* instance except for *CodeMeter WebAdmin* and *CodeMeter Control Center* are disabled.

Also *CmLAN* is not supported for TMR Servers.

12.4CodeMeter Control Center

CodeMeter Control Center serves to locally configure *CodeMeter License Server*. Software-sided, *CodeMeter License Server* as the runtime environment is at the heart of *CodeMeter*. It allows the access to *CmContainer*. In doing so, *CmContainer* can be locally connected or are available on a network. By default, *CodeMeter License Server* is installed as service or deamon (Linux, macOS) and automatically starts when the system starts.

When the service has been started, other programs are available to access licenses stored in *CmContainer* and use protected data areas in a *CmContainer*.

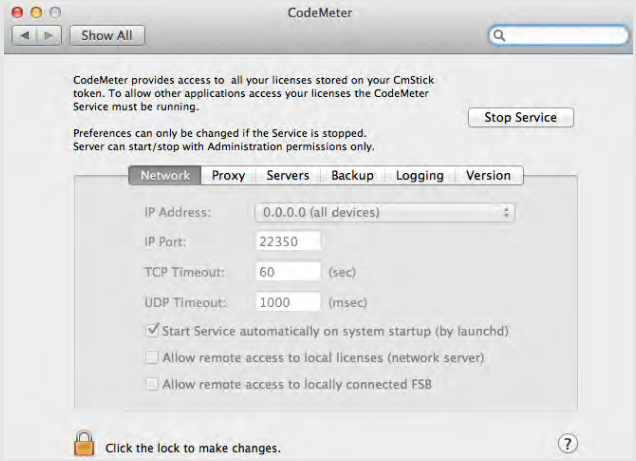
Operating System	Menu Control
Windows	[Start - All Programs - CodeMeter - CodeMeter Control Center]
	Press "Windows" key to open Start screen | Type "CodeMeter Control Center" | Press "Enter" key.
macOS	[Programs - CodeMeter - CodeMeter Control Center
Linux	[Applications - System - CodeMeter Control Center] or [Applications - Accessories - CodeMeter Control Center]

CodeMeter License Server starts only one-time on each PC!

Start and Stop *CodeMeter*-service or daemon

The following table shows you for different operating systems how start or stop the *CodeMeter* service or daemon.

Operating system	Description
Windows	**1.** Navigate via **"Windows | Control Panel | Administrative Tools | Services"** to *CodeMeter License Server*. **2.** Right mouse-click and **'Start'** or '**Stop**' the service. Alternatively, use the **"Action** 388 **"** menu of *CodeMeter Control Center*.

Operating system	Description
macOS	**1.** Navigate via **"System preferences | Other"** to the *CodeMeter* icon.  **2.** Click the *CodeMeter* Icon. The CodeMeter dialog displays  **3.** Click the **"Stop Service"** or **"Start Service"** button to stop or start the service.

Linux	**1.** Call the following script with `'sudo'` root privileges to stop the service:
	`/etc/init.d/codemeter stop.`
	2. Call the following script with `'sudo'` root privileges to start the service:
	`/etc/init.d/codemeter start` or alternatively the command `service codemeter start.`

CodeMeter License Server uses TCP/IP network protocol for communication and the default port `22350`. Make sure your firewall does not block this port. Please make sure that the used IP-Port `22350` is available for *CodeMeter* .

12.4.1 Structure and Navigation

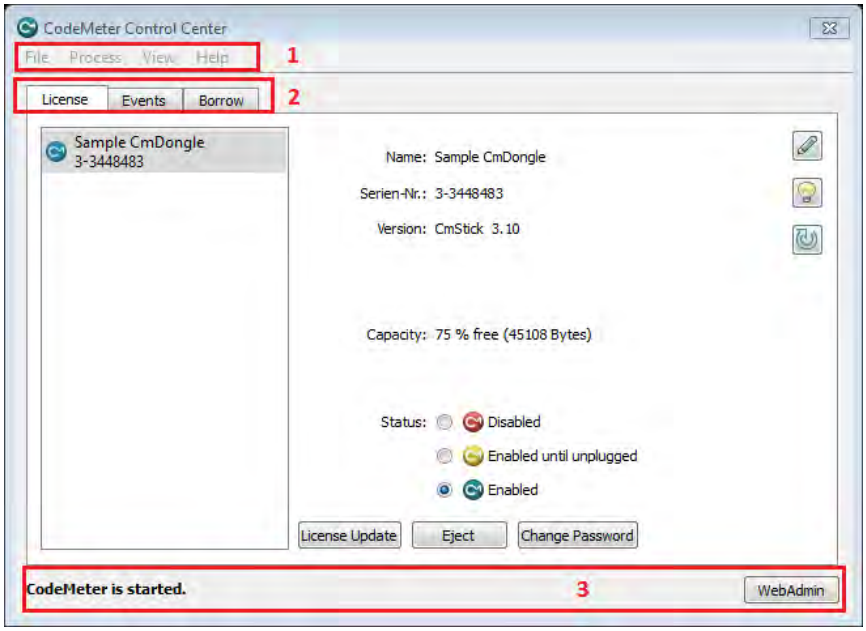


Figure 243: *CodeMeter Control Center* - Overview

The *CodeMeter Control Center* user interface is divided in three separate areas:

- menu bar[387] (**1**)
- Tab areas (**2**)
- *Status and Open CodeMeter WebAdmin*[393] (**3**).

Starting *CodeMeter Control Center*

You access and start *CodeMeter Control Center* in several ways:

Open
• Double-click on the *CodeMeter* or symbols in the info area of the Windows task bar
• Right mouse-click on the *CodeMeter* or symbol there, and subsequently select the **"Show"** menu item. The *CodeMeter Control Center* secondary menu (right mouse-click on the *CodeMeter* symbol) provides the additional menu items:

Item	Description
WebAdmin	Starts *CodeMeter WebAdmin* in the default Internet browser.
Eject all CmDongle(s)	Option to safely disconnect *CmDongles*.
Disable CmDongle	Prompt to insert the Cm*Dongle* Password.
Help	Opens the *CodeMeter* help.
About	Shows general information on *CodeMeter* components.
Quit	Exits but not shuts down the service *CodeMeter License Server*.

• Navigation by the **"Start | All Programs | CodeMeter Control Center"** start menu (Press "Windows" key to open Start screen | Type "CodeMeter Start Center" | Press "Enter" key).

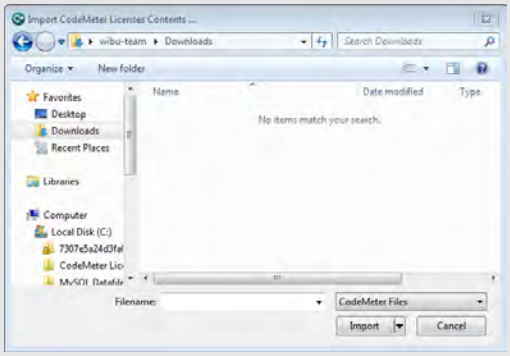
In the info area of the Windows task bar, different colors of the *CodeMeter*® symbols represent different status conditions of connected *CmContainer.*

Color	Status
Grey	No *CmContainer* is connected, or *CodeMeter License Server* is not started.
Green	An activated *CmContainer* is connected.
Blue double	Several *CmContainer* are connected and activated until disconnected.
Yellow	A *CmDongle* is connected and activated until it is disconnected.
Red	A deactivated *CmContainer* is connected.

Figure 244: *CodeMeter* Symbols Windows Task Bar

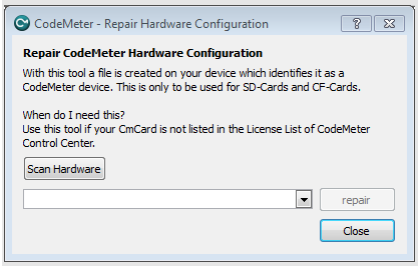
12.4.2 Menu Bar

File Menu

Element	Description
Import license	In order to import license contents using *CodeMeter Control Center*, proceed as follows: **1.** Select the **"File | Import License..."** item. **2.** Select in the following **"Import CodeMeter License Contents ..."** dialog the *CodeMeter* files of the types `*.WibuCmRaU`; `*.wbb`; `*.wbc` and read in license data by clicking on the **"Import** button.  Figure 245: *CodeMeter Control Center* - Import Licenses Alternatively, you can also directly import the license file using the Windows Explorer. Simply drag & drop the file in the **License** tab area of *CodeMeter Control Center*.
WebAdmin	Opens *CodeMeter WebAdmin* in the default Internet browser. Alternatively, press the key combination `<CTRL>+W`.
Logging	Saves all *CodeMeter* events to a log file. Alternatively, press the key combination `<CTRL>+L`. When you activate the logging, this also affects the logging display in *CodeMeter WebAdmin* on the **"Diagnosis** 416 **"** page. On Windows operating systems this log file is stored to the directory `%\Programm Files%\CodeMeter\Logs`. This log file is especially important for trouble shooting.

Element	Description
Preferences	Opens *CodeMeter WebAdmin* and is defaulted on the page where you are able to apply network settings[425].
Exit	Exits *CodeMeter Control Center*. Alternatively, press the key combination `<CTRL>+Q`. The service*CodeMeter License Server* however is not shut down.

Processes Menu

Element	Description
Eject all CmDongles	Ejects all connected *CmDongles* in one go. Alternatively, press the key combination `<CTRL>+ALT+Q`.
Defragment License Memories	Defragments the license memory of the selected *CmContainer*. Alternatively, press the key combination `<STRG>+ALT+D`.
Update Time Certificates	Updates the time certificates in the selected *CmContainer*. All time stamps are refreshed.
Start CodeMeter Service	Starts the service *CodeMeter License Server*. Use this menu item if *CodeMeter License Server* has been stopped before, for example, when you made changes on the network settings in *CodeMeter WebAdmin* which require the restart of the service. When you have administrator privileges under Windows you can also manage the *CodeMeter License Server* service by setting the desktop management (System Settings | Management | Services).
Repair Hardware Configuration	Repairs the hardware configuration of the *CmDongle* form factors SD Card and CF Cards. This tool is required if the *CmCard* hardware is not listed in the license list of *CodeMeter Control Center*. 
Stop CodeMeter Service	Stops the service *CodeMeter License Server*.
Restart CodeMeter Service	Restarts the service *CodeMeter License Server*.

View Menu

Element	Description
Hide Window	Minimizes and hides the *CodeMeter Control Center* window back into the info area of the Windows task bar. Alternatively, press the key combination `<CTRL+M>`.
Refresh	Refreshes the display of all connected *CmContainer*. Alternatively, press the key `<F5>`.
Zoom in	Enlarges the display in the **Events** tab area. Alternatively, press the key combination `<CTRL>++`.
Zoom out	Scales down the display in the **Events** tab area. Alternatively, press the key combination `<CTRL>+-`.
Copy Event Content	Copies the event actions in the **Events** tab area to the clipboard. Alternatively, press the key combination `<CTRL>+C`.
Clear Event Content	Deletes the event actions in the **Events** tab area. Alternatively, press the key combination `<ALT>+C`.
Show all connected CmContainer	Shows all connected *CmContainer* including details in the **Events** tab area. Alternatively, press the key combination `<ALT>+S`.
List all open Handles	Shows all open handles in the **Events** tab area. Handles work as references for the developer for further programming.
Show all available License Entries	Shows all *CmContainer* license entries in the **Events** tab area. Alternatively, press the key combination `<ALT>+E`.
Borrow visible	Toggles between a visible and not visible **Borrowing** tab area. By default, starting with *CodeMeter* Version 6.00a this tab area is not visible.

Help Menu

Element	Description
Help	Opens the *CodeMeter* online help. Here you access the help files on *CodeMeter License Server* and *CodeMeter Control Center*.
Register CmDongle	Opens the secure website https://my.codemeter.com to register *CmDongles*.

About	Informs on the started *CodeMeter Control Center* version.

12.4.3 License Tab

The **"License"** Tab shows you information on connected *CmContainer* and provides some options to configure connected *CmContainer*. Moreover, you are able to update licenses located in your *CmContainer* using the *CmFAS Assistant* 394.

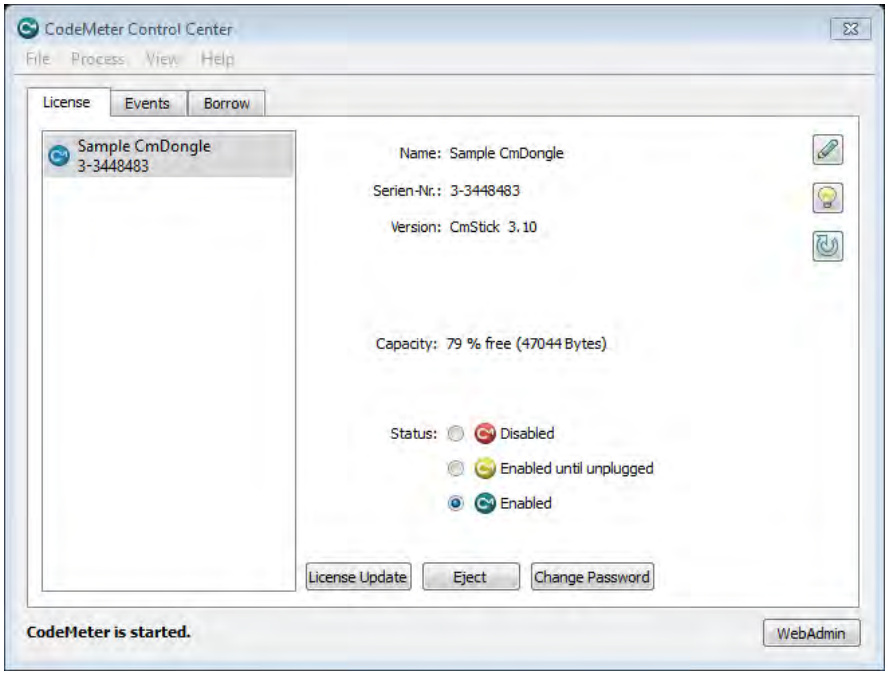
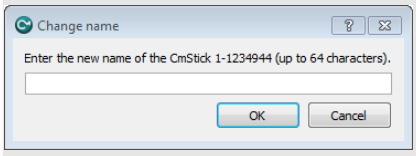
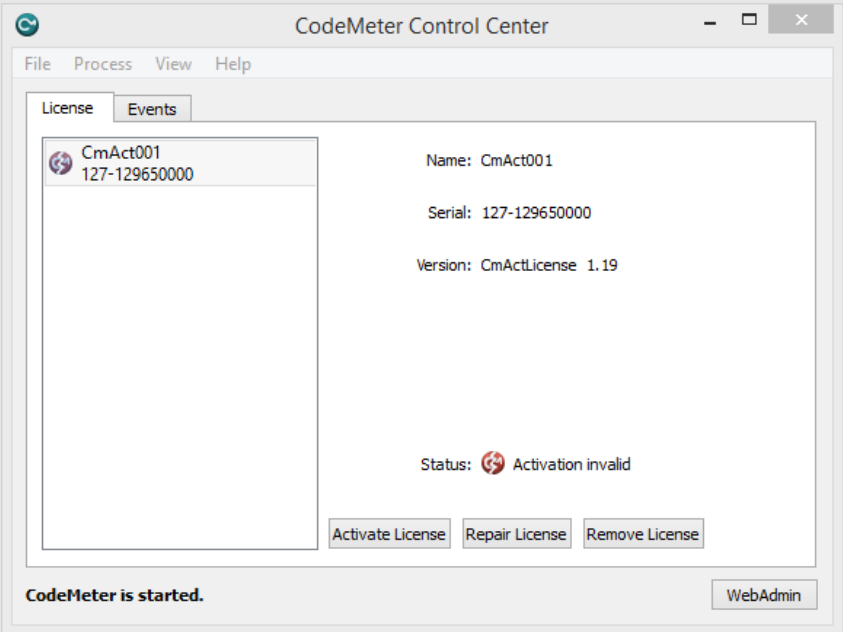
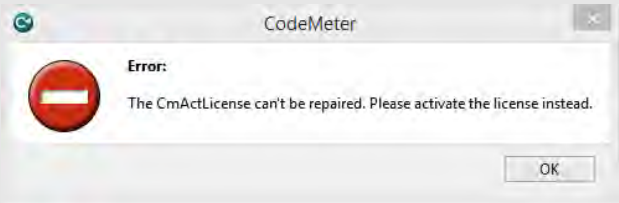


Figure 246: *CodeMeter Control Center* License Tab

Element	Description
Name	Changes and displays the name of the selected *CmContainer*. In the subsequent dialog you can edit the name.  Figure 247: *CodeMeter Control Center* - Change Name of *CmContainer*
	Flashes the LEDs of the selected *CmStick*. This eases the identification of a *CmStick,* if several *CmSticks* are connected.
	Updates the firmware of the selected *CmDongles*. This guarantees the correct execution of essential functions, and solves eventually occurring problems. When you execute a firmware update, you require an Internet connection. Then *CodeMeter Control Center* automatically connects to the Firmware Update Server of Wibu-Systems. You are prompted to enter your *CmDongle* Password in order to confirm this action. The update may take a couple of minutes. You must not remove the *CmDongle* before this process is finished. Otherwise, irreparable damage of the *CodeMeter* SmartCard Chip may occur.
Capacity	Informs on the capacity of the *CodeMeter* SmartCard Chip of a selected *CmDongle*. The capacity is displayed in percent format, and by number of absolute bytes. Please note that this value tells nothings about the memory allocation of an eventual flash memory of a *CmDongle*.
Status	The status group informs on the activation status of the selected *CmDongle*. Color: Status The connected *CmContainer* is disabled. No licensed application can use license information in the *CmContainer*. This is may be the case, if a *CmActLicense* license is 'broken'.

Element	Description	
	Color	Status
		 Then the **"Repair license"** button is available to eventually repair the license. If this is not successful, the following error message displays. In this case, only a reactivation is helpful. Please contact your software vendor.  For information on eventual reasons for a 'broken' license als a look at the log on the Events 391 tab may be helpful.
		The *CmDongle* is enabled as long as it is connected. If the *CmDongle* is removed from the PC, automatically the licensed access by applications is deactivated.
		The *CmContainer* is fully enabled. In the case of a *CmDongle,* the licensed access of applications is still featured even if the *CmDongle* is removed.
		Wibu-Systems recommends the activation status **"Enabled until plugged out"**. This ensures that even when a *CmDongle* is lost, unauthorized access to the licenses and personal data in the *CmDongle* is not possible.
		Changing Activation Status In order to change the activation status, please proceed as follows: **1.** Select the radio button of the desired status option. **2.** Enter the *CmDongle* Password in the following dialog.
		The default password for *CmDongle* is `"CodeMeter"`.
		 Figure 248: *CodeMeter Control Center* - Enter Password **3.** Click the **"OK"** button to confirm the status change.

Element	Description
License Update	Click this button to request new, or update existing licenses for selected *CmContainer*. The *CodeMeter Field Activation Service* (CmFAS) Assistant[393] opens.  Figure 249: *CodeMeter Control Center - CmFAS Assistant*
Eject	Click this button to disconnect the selected *CmDongle*. The *CmDongle* logs off from the operating system, and can be safely removed from the PC.
Change Password	Click this button to change the password of the selected *CmDongle*. In the following **"Change Password"** dialog please complete the respective fields.  Figure 250: *CodeMeter Control Center* - Change Password **1.** Enter in the **"Old Password"** field the currently used *CmDongle* password. The default password for *CmDongle* is `"CodeMeter"`. **2.** Enter in the **"New Password"** field the new desired *CmDongle* password. **3.** Re-enter in the **"Retype Password"** field the new desired *CmDongle* password. If you forgot the *CmDongle* password, you have the option to set a new *CmDongle* password by using the *CmDongle* Master Password. **4.** Click the **"OK"** button to confirm your input. **5.** Activate the **"Input Master Password"** option and specify your *CmDongle* Master Password in the **"Old Password"** field. A Master Password you have received when you registered at the website my.codemeter.com. In order to register, use the **"Help | Register CmDongle"** menu item. A registration bears several advantages and serves to provide security when using *CodeMeter*. Only when you are registered loosing the own password can be remedied by requesting a Master Password.

12.4.4 Events Tab

This tab displays information at start and at runtime of *CodeMeter License Server* and comprises the following items:

- number of connected *CmContainer*
- number of *CmContainer* entries
- number of found license container at the *Firm Item* level
- accesses to *CodeMeter License Server*

You configure the display of the event list using the **"View | ...**[388] **"** menu item.

You log the content for the event view using the **"File | Logfile**[387] **"** menu item.

12.4.5 Borrowing Tab

By default, starting with *CodeMeter* Version 6.10 this tab displays only, if borrowing clients entries exist programmed with the 'old' borrowing using prepared *CmContainer*.

This tab informs on borrowable licenses as a feature of *CodeMeter* license borrowing [43]. Then licenses can also be used when the access to license information does not require to be connected to the license server.

You can toggle the view of this tab using the "**View | Borrow visible**" menu item.

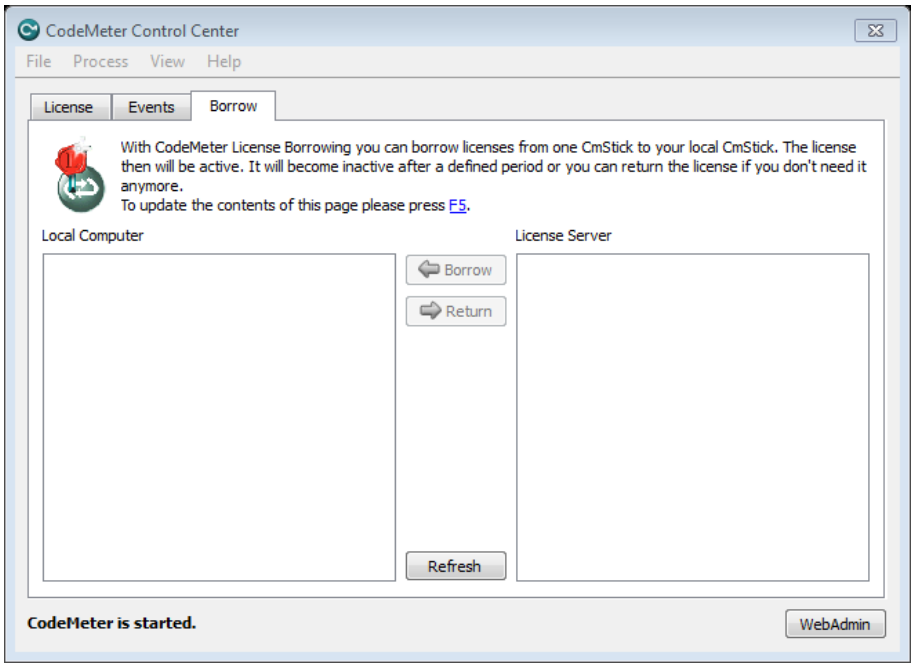


Figure 251: *CodeMeter Control Center* - Borrowing Tab

License Server

On the right, you see all licenses available for the 'License Borrowing' feature. The licenses are ordered by existing license server, *Firm Items*, and *Product Items*. The displayed licenses either are borrowable or inactive.

You can borrow only active licenses. You recognize active licenses by the colored symbol and the activated **"Borrow"** button.

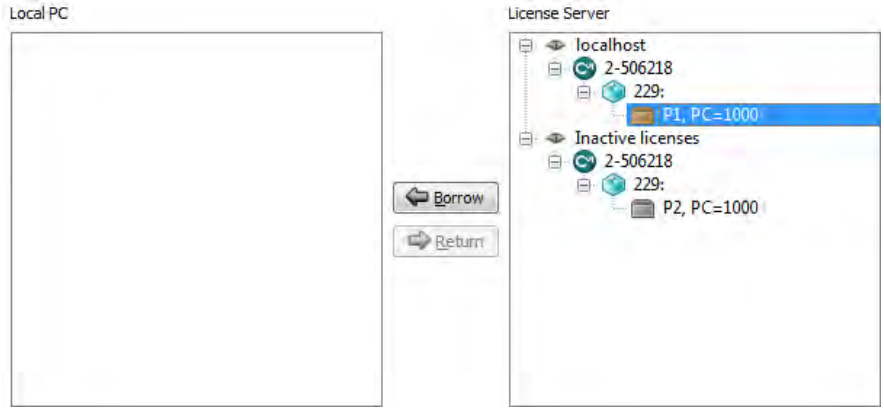


Figure 252: *CodeMeter Control Center* - Borrow Licenses

1. Click on the **"Borrow"** button to borrow licenses from the license server for the local PC.

Local PC

On the left, all licenses borrowed for the local use on a PC from a license server are displayed.

These licenses are deactivated according to the defined borrowing period. However, you also have the option to return borrowed licenses before the borrowing period expires.

1. Click on the **"Return"** button to return borrowed licenses, and make them available again for the license server.

Figure 253: *CodeMeter Control Center* - Return Licenses

For refreshing the display of the tab press the key <F5> or the **"Refresh"** button.

12.4.6 Status and Starting CodeMeter WebAdmin

Status

This area displays information on the *CodeMeter License Server* status, i.e. if this service is started or not. If you want to change the status, use the **"Process | Stop CodeMeter Service"** or **"Process | Start CodeMeter Service"** menu items.

WebAdmin

Click this button to open *CodeMeter WebAdmin*. Alternatively, you can use the **"File | WebAdmin"** menu item.

12.5Importing and Updating Licenses

The *CmFAS Assistant* [394] supports you in importing and updating license files for your *CmContainer*. Using various dialogs you manually create license requests, import license updates, and, optionally, create receipts for these operations the end-user then sends to the software vendor. Using license files also allows the activation of licenses on a PC which has no direct Internet access. The figure below illustrates this process.

Please note that importing license updates files (`*.WibuCmRaU`) is currently not supported for a *CmContainer* in operation. Before a license update, please save your work and close all other running *CodeMeter*© protected applications which access licenses on the target *CmContainer*.

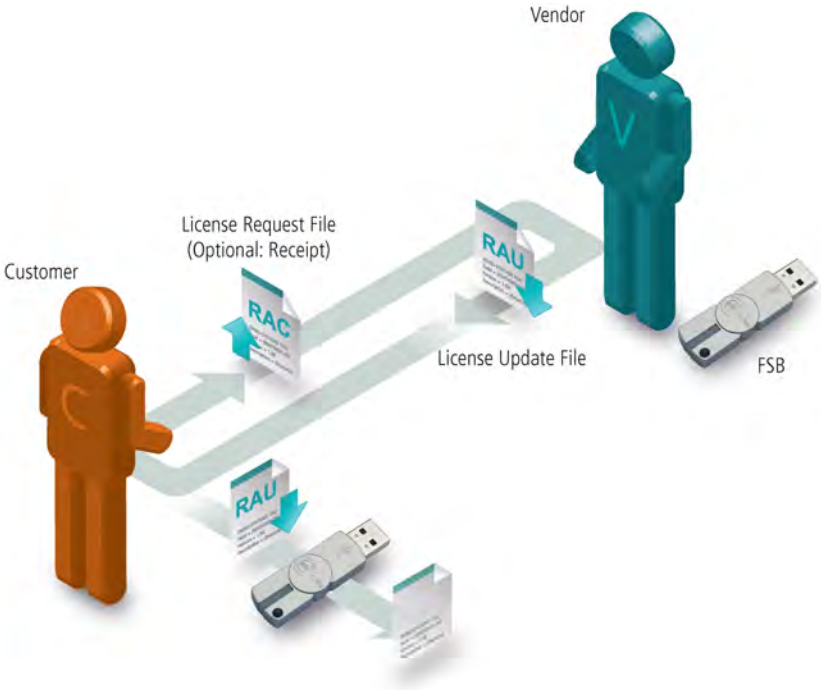


Figure 254: *CmFAS* - File-based Remote Update

12.5.1 The CmFAS Assistant in CodeMeter Control Center

> ⚠ Please note that importing license updates files (`*.WibuCmRaU`) is currently not supported for a *CmContainer* in operation.
> Before a license update, please save your work and close all other running *CodeMeter*© protected applications which access licenses on the target *CmContainer*.

1. Open *CodeMeter Control Center*. If several *CmContainer* are connected to the computer, select the desired *CmContainer*.
2. Click on the **"Update License"** button.

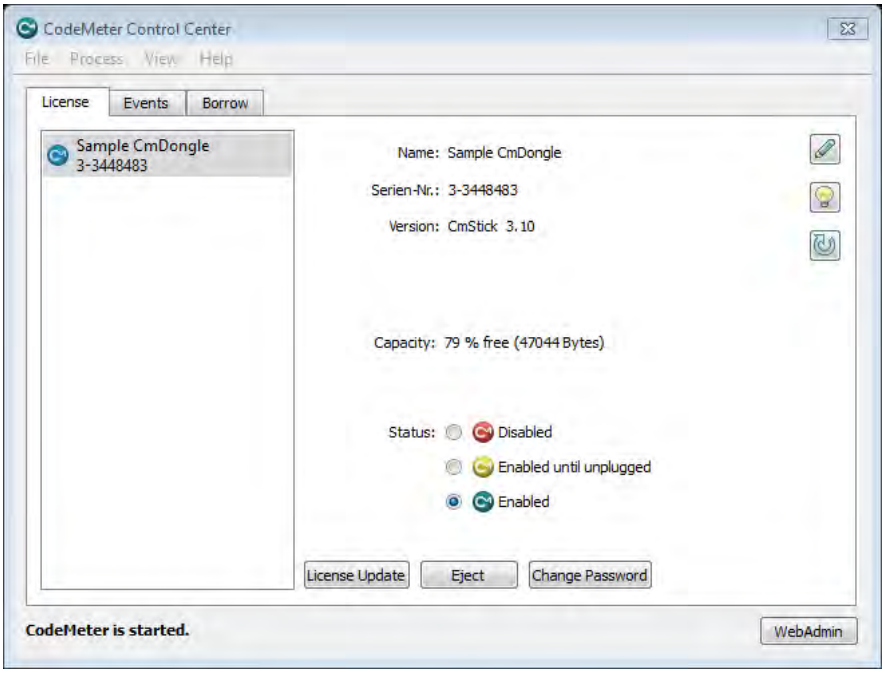


Figure 255: License Update - *CodeMeter Control Center*

The *CodeMeter Field Activation* (CmFAS) Assistant opens with a welcome dialog.

Figure 256: *CmFAS Assistant*

3. Click the **"Next"** button.

12.5.1.1 Create License Request File

The starting dialog prompts you to proceed. There you select from creating a license request, import a license update you received from the software vendor, or, optionally, create a receipt after an update to send it to the software vendor. After your selection click the **"Next"** button.

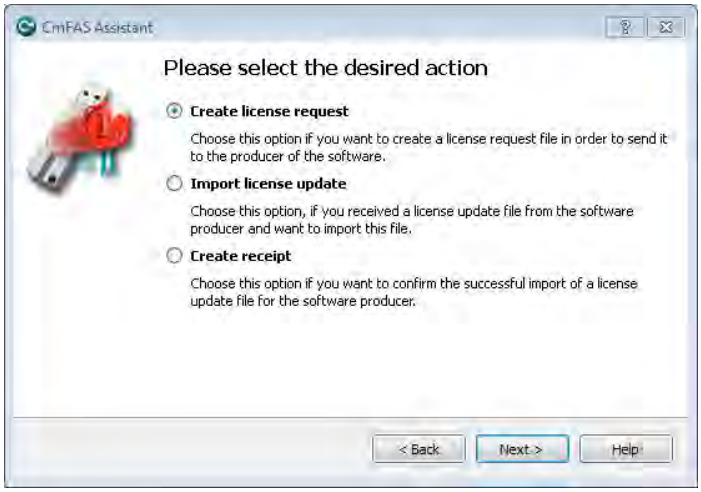


Figure 257: *CmFAS* - Create License Request

12.5.1.1.1 Extend Existing License

On creating a license request, you select whether you want to extend an existing license, or add a license of a new vendor. After your selection click the **"Next"** button.

Figure 258: *CmFAS* – Extend existing License

When you extend an existing license, select the software vendor(s) for which you want to create a license request. After your selection click the **"Next"** button.

Figure 259: *CmFAS* - License Extension - Select Vendor

The next dialog allows you to save the license request file to a desired location. Then click the **"Commit"** button to create the file. This file you then can send by e-mail to the software vendor.

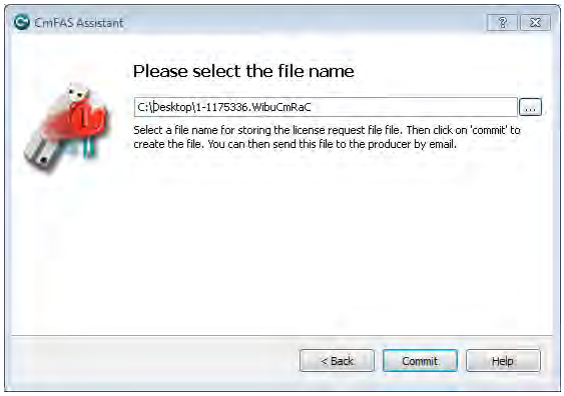


Figure 260: *CmFAS* – License Extension – Save File

Finally, a dialog displays which confirms the successful creation of the license request file. Click the **"Finish"** button to close the dialog.

12.5.1.1.2 Add a License of a new Producer

On creating a license request you can decide to extend an existing license, or to add a license of a new vendor. Select **"Add license of a new vendor"** and click the **"Next"** button.

Figure 261: *CmFAS* – New License

In the next dialog, specify the *Firm Code* you received by the software vendor, and click the **"Next"** button.

Figure 262: *CmFAS – Firm Code*

The next dialog allows you to save the license request file to a desired location. Then click the **"Commit"** button to create the file. This file you then can send by e-mail to the software vendor.

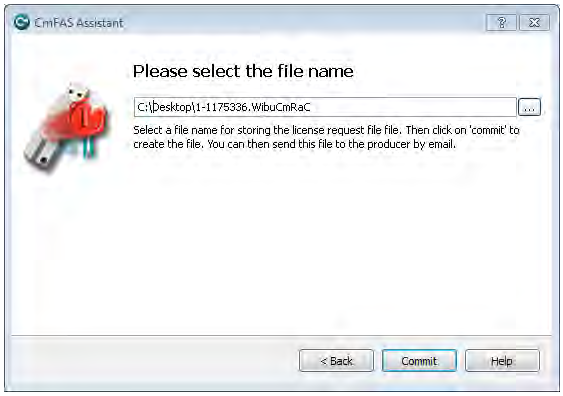


Figure 263: *CmFAS* - Save File

In both case, either when extending or adding a license you receive a confirmation the license request file has been successfully created. Click on the **"Finish"** button to complete this process.

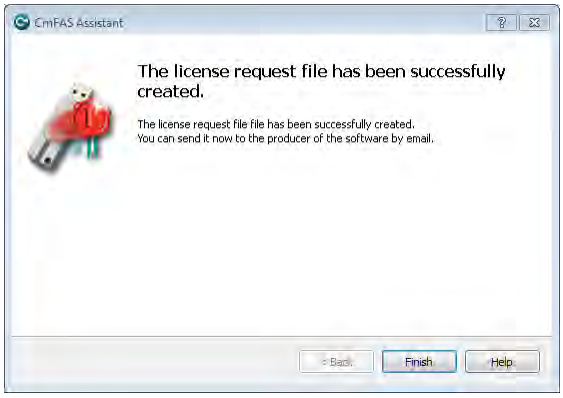


Figure 264: *CmFAS* - Receipt

12.5.1.2 Import License Update

> Please note that importing license updates files (`*.WibuCmRaU`) is currently not supported for a *CmContainer* in operation.
> Before a license update, please save your work and close all other running *CodeMeter*© protected applications which access licenses on the target *CmContainer*.

In order to import a license update, in the start dialog select the respective option, then click the **"Next"** button.

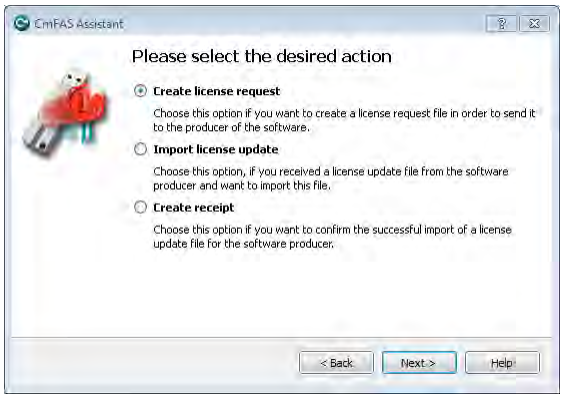


Figure 265: *CmFAS* - Import License Update

In the next dialog, select the file name you used when saving the license update file you received. Then click the **"Commit"** button to import the license update file.

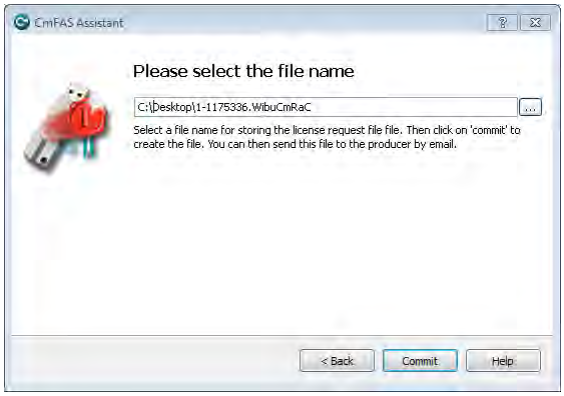


Figure 266: *CmFAS* - License Update - Save File

The following dialog confirms the successful import. Optionally, you can send a receipt to the software vendor. This option you also have in the start menu. Click the **"Finish"** button.

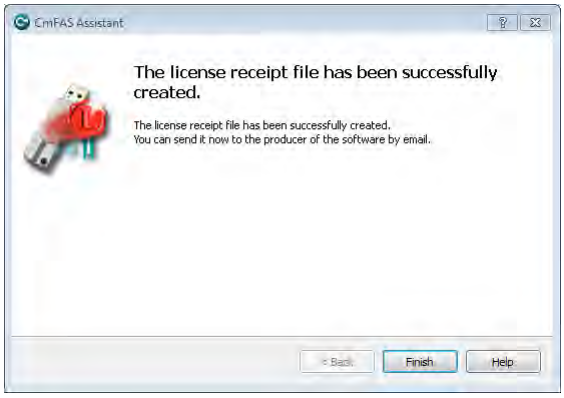


Figure 267: *CmFAS* - License Update - Receipt

12.5.1.3 Create Receipt

In the start menu, select the option **"Create Receipt"**, then click the **"Next"** button.

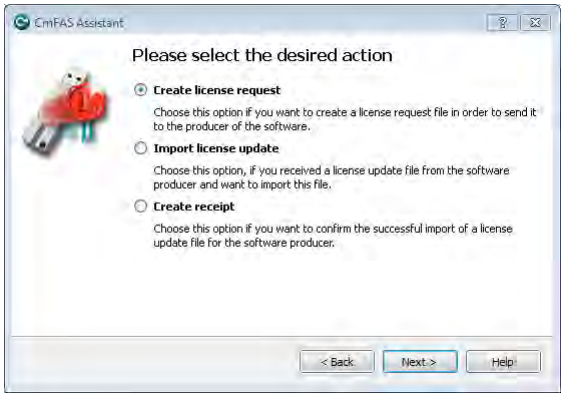


Figure 268: *CmFAS* - Create Receipt

In the next dialog, select the software vendor you want to send the receipt to, then click the **"Next"** button.

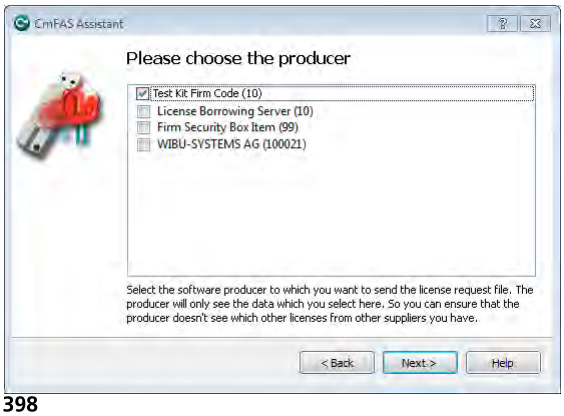


Figure 269: *CmFAS* - Create Receipt - Producer

Save the receipt file using the **"Commit"** button and send it to the software vendor.

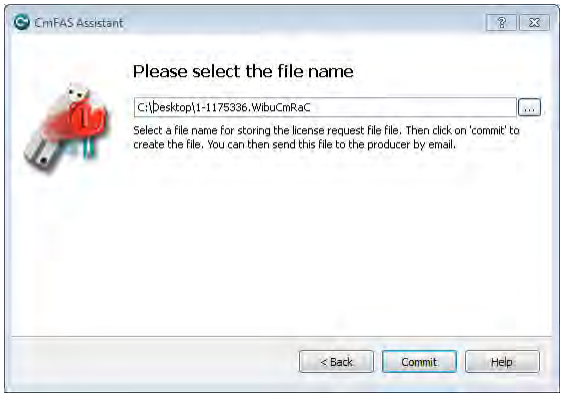


Figure 270: *CmFAS* - Create Receipt - Save File

The successful creation of the receipt file is confirmed in the next dialog. Click on the **"Finish"** button to complete this process.

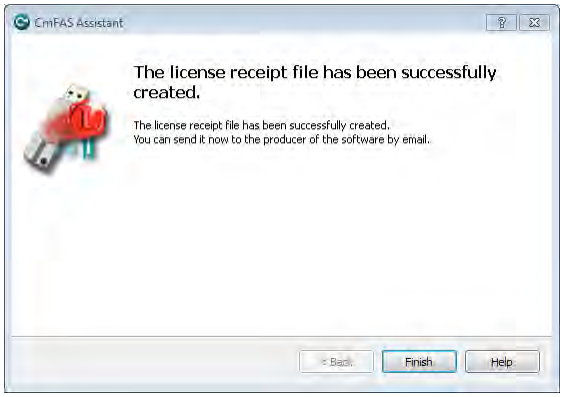


Figure 271: *CmFAS* - Create Receipt - Receipt

12.6 CodeMeter WebAdmin

With *CodeMeter WebAdmin* you obtain information on connected *CmContainer* and available licenses stored in them. In addition, you configure the service *CodeMeter License Server*. In detail, *CodeMeter WebAdmin* provides many configuration and analysis options in the following areas:

- **status information**: host[403], *CmContainer*[404]
- **configuration**[417]: use as network server, proxy settings, access protection, remote access, time server, backup
- **display**[411]: display of all existing licenses locally and on the network, view of license conditions, session information
- **management**[413]: management of network licenses by manual allocation of licenses
- **diagnosis**[416]: logging
- **backup**[425].

The following list briefly describes terms which recur on single pages in *CodeMeter WebAdmin*.

Term	Description
Access Mode	see: Status
Activation Time	Informs on the activation time of a license, i.e. the start time of a valid license.
Borrow Licenses	Informs on existing borrowed licenses, the borrowing period, and a unique security identifier (SID) when used on a network.
Currently Borrows Licenses	Number of the currently borrowed licenses.
Expiration Time	Informs on the expiration date of a license, i.e. when the license expires.
Extended Protected Data	Additional entry field for binary data for the licensor.
Feature Map	Informs on licenses which the licensor delivers with different functionalities and modules, or in different versions. These are mapped by *Feature Maps* describing a special functional scope. The value specified here informs on the valid functionality or the activated module/version.
Firm Code	Number which identifies the separate license container of a licensor.
Hidden Data	Additional entry field for binary data for the licensor.
Implicit Firm Item (IFI)	The license container holding licenses the user is able to use only with his/her *CmDongle* Password. This license container is identified by the number of "0".
License Quantity	Informs on the total number of licenses available for a license.

Term	Description
Linger Time	Informs on the time how long the license lingers after the license is re-allocated after the protected application is closed.
Maintenance Period	Informs on the period in which a protected version of the software has to be released to represent a licensed version. The start and the end of the period displays.
n/a	Informs that no related entry exists for this license (not available).
Product Code	Number which identifies the license entry, i.e. a product, of a licensor.
Protected Data	Additional entry field for binary data for the licensor.
Secret Data	Additional entry field for binary data for the licensor.
Status	Informs on how the number of started instances of a protected software relates to the allocation of licenses. User Limit: here each started instance allocates a license. Shared: here several started instances of the same application on the same PC allocate only a single license. Exclusive: here a protected application runs only once on a PC. No User Limit: here any number of started instances of the protected application can be started on the network without allocating additional licenses.
Unit Counter	Informs on licenses which are billed by use (pay-per-use, pay-per-print, etc.). This is implemented by counters which are decremented on use of a product. The value specified here informs on remaining units for the use of a license.
Usage Period	Informs on the usage period of a license. The value specified here informs on the use of a licenses in days. The value can also be bound to a starting time for the validity of a license.
User Data	Additional entry field for binary data for the licensee.

Table 9: *CodeMeter WebAdmin* - Terms in License Display

If *CodeMeter WebAdmin* should not start, please proceed as follows:

1. Check if the used Internet browser is not set to "offline mode".
2. Check the JavaScript support of your Internet browser.

   JavaScript must be activated for effective using *CodeMeter WebAdmin*.

3. Type in the URLs: `http://localhost:22350` or `http://127.0.0.1:22350` directly in the address field of your Internet browser.

The use of TCP/IP in *CodeMeter*

The communication between protected applications and *CodeMeter License Server* bases on the Transmission Control Protocol/Internet Protocol (TCP/IP). This is valid not only for locally existing licenses, but also for licenses which are provided via a network.

By default, *CodeMeter* uses the port 22350 registered by Wibu-Systems at IANA (Internet Assigned Numbers Authority) and uniquely assigned for the *CodeMeter* communication. The list of assigned ports can be viewed at www.iana.org/assignments/port-numbers.

In order to make available a client access to a *CodeMeter License Server* on the network, a communication using the *CodeMeter* port must be supported. If the server should locate in another network area, eventually the port must be made known and accessible as part of the infrastructure (router, firewall, etc.).

For the direct access to *CodeMeter License Server* on the network, the communication bases on TCP. For an automatic search of servers on the network, additionally a broadcast via UDP (User Datagram Protocol) is performed (it is listened only at server search time and only until the end of the UDP Waiting Time).

The access using the *CodeMeter* port is performed only for the access to *CodeMeter License Servers* and this only within the organization which runs the network server.

In particular, using this port **no** communication into the internet is performed.

In *CodeMeter* settings of *CodeMeter WebAdmin* an option exist to configure 425 the *CodeMeter* port to a value other than the default of 22350. However, such a change should have plausible reasons, e.g. in the case of parallel test environments on the same network. In addition, such a change requires the same configuration of all affected *CodeMeter License Servers*.

12.6.1 Basics

TCP/IP based

Communication between *CodeMeter WebAdmin* and connected *CmContainer* is browser-based and uses network components. Thus the installation of the network protocol TCP/IP is required, and access must be granted to the `localhost`.

However, an actual connection to the Internet is not established.

Network Server and Server Access

By default, *CodeMeter License Server* is only available locally (`localhost`).

A change of the remote access to CodeMeter WebAdmin 420 during operation requires a restart of the *CmWebAdmin* service. Please note the firewall settings.

Firewall Settings

Please also note that the settings of your firewall do not block communication.

> *CodeMeter License Server* uses a specific IP port (defaulted on `22350`) to communicate with your PC and the network. This network port is registered at IANA (Internet Assigned Numbers Authority) and uniquely assigned for *CodeMeter* communication.
>
> Make sure that your firewall is not blocking this port. Enable the used IP port `22350` and make sure it is accessible by *CodeMeter*, i.e. share the communication for this IP port.
>
> If the *CodeMeter WebAdmin* communication is required not only locally but also in server operation, you must also release port `22352` (in the case of HTTPS, port `22353`), since port `22350` is automatically forwarded.

Communication Mode

By editing registry or server entries you are also able to define which communication mode *CodeMeter License Server* uses.

The following table shows you where for which operating system you find the profiling to set the communication mode.

Operating system	Registry / Server Entry
Windows	`HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini`

The parameter **ApiCommunicationMode**. is available for setting the mode. The following properties are available:

CodeMeter-Version	Properties
smaller than 4.40	'1' TCP/IP (Default) '2' Shared Memory
starting with 4.40	'1' Platform-specific (Default) Platform-specific defaults: • Windows: IPv6, IPv4; Shared Memory • Linux/macOS:IPv6, IPv4 • WinCE: IPv4, Shared Memory '2' Shared Memory '4' IPv4 '8' IPv6 Single modi may be combined.

> Wibu-Systems recommends to use the relevant default settings, if no justified reasons suggest otherwise.

12.6.2 Open CodeMeter WebAdmin

CodeMeter WebAdmin is a web-based tool to be displayed with any standard internet browser.

The following table shows existing start options.

Operating System	Start
Windows	• via *CodeMeter* symbol in the task bar (right mouse-click) and selection of 'WebAdmin' item. • via the 'WebAdmin' option in *CodeMeter Control Center* • directly in your Internet browser when typing in the URLs: **http://localhost:22352** or **http://127.0.0.1:22352**.
macOS / Linux	• via *CodeMeter* in the task bar (right mouse-click) and selection of 'WebAdmin' item. • via the 'WebAdmin' option in *CodeMeter Control Center* • directly in your Internet browser when typing in the URLs: **http://localhost:22352** or **http://127.0.0.1:22352**.

If *CodeMeter WebAdmin* should not start, try the following:

1. Check if the used Internet browser is not set to "offline mode".
2. Check the JavaScript support of your Internet browser.

   > JavaScript must be activated for effective using *CodeMeter WebAdmin*.

3. Type in the URLs: `http://localhost:22352` or `http://127.0.0.1:22352` directly in the address field of your Internet browser.

On all pages you are able to select from a list of available server.

Right to the display **"Current Server"** follows the name of the actual PC on which the service *CodeMeter License Server* is started. A search request using the port `22352` is sent to the network. For changing the server, please proceed as follows:

1. Click the icon.
   A dialog displaying a list of all available server displays.

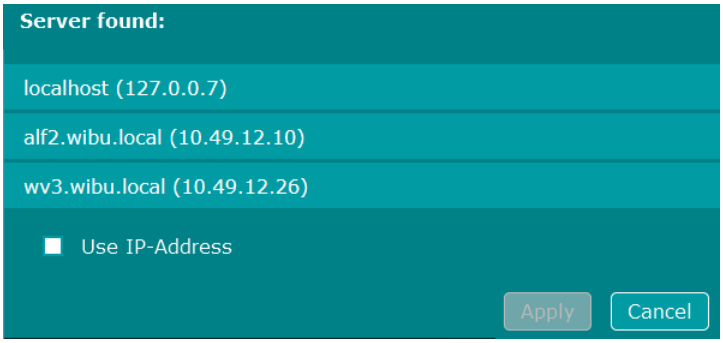


Figure 272: *CodeMeter WebAdmin* – Available Server

2. Select another PC on which *CodeMeter* is also started and the service *CodeMeter License Server* runs.
   The entry changes color to orange and the **"Apply"** button becomes operable.
3. Click the **"Apply"** button.
4. Check the **"Use IP Address"** check box, if you want to use the IP address.

12.6.3 Operating

CodeMeter WebAdmin uses the following elements covering the navigation through the pages, the display of information, and actions.

Navigation element	Description
License Monitoring All Licenses Sessions License Tracking	Classical tab menu including dropdown controls.
Server Configuration > Server Access	Breadcrumb trail navigation area.
or	Expand or collapse additional detailed information.
R W	The pictograms inform on **R**ead and **W**rite access. On mouse over tool tips display additional access mode details. A click on a pictogram opens - if required - the login on a separate page. Depending on the configured access modes a dropdown menu provides the following entries: • **Allow write access** • **revoke read access** • **Revoke read/write access** For access settings see WebAdmin configuration page 420. A click on a pictogram opens - if required - the login on a separate page.

Display element	Description
	Image to show that the *CmContainer* is a *CmDongle*.
	Image to show that the *CmContainer* is a *CmActLicense*.
	Image to show that the *CmContainer* is a *CmCloudContainer*.
	Image to show that the *CmContainer* is a virtual *vCmContainer*.
	Information icon changing its color from orange to red informs on important information to read and respond, e.g. restarting of *CodeMeter License Server*.
O	Icon displaying license transfer options. Hover over control changes color to orange and displays information. A click advances to the *Product Item* detail page.
H	Icon displaying license transfer history. Hover over control changes color to orange and displays information. A click advances to the *Product Item* detail page.

Action element	Description
	Opens a dialog to select available server.
	Defragmenting of the *CmDongle* chip memory.
	Updating the *Certified Time* of the *CmContainers* using a *CodeMeter Time Server*.
	Adds a new entry to a list.
	Deletes a list entry.
	Moves up a list entry a level higher.
	Moves down a list entry a level lower.

12.6.4 Dashboard

The dashboard displays basic server information on the *CodeMeter* runtime.

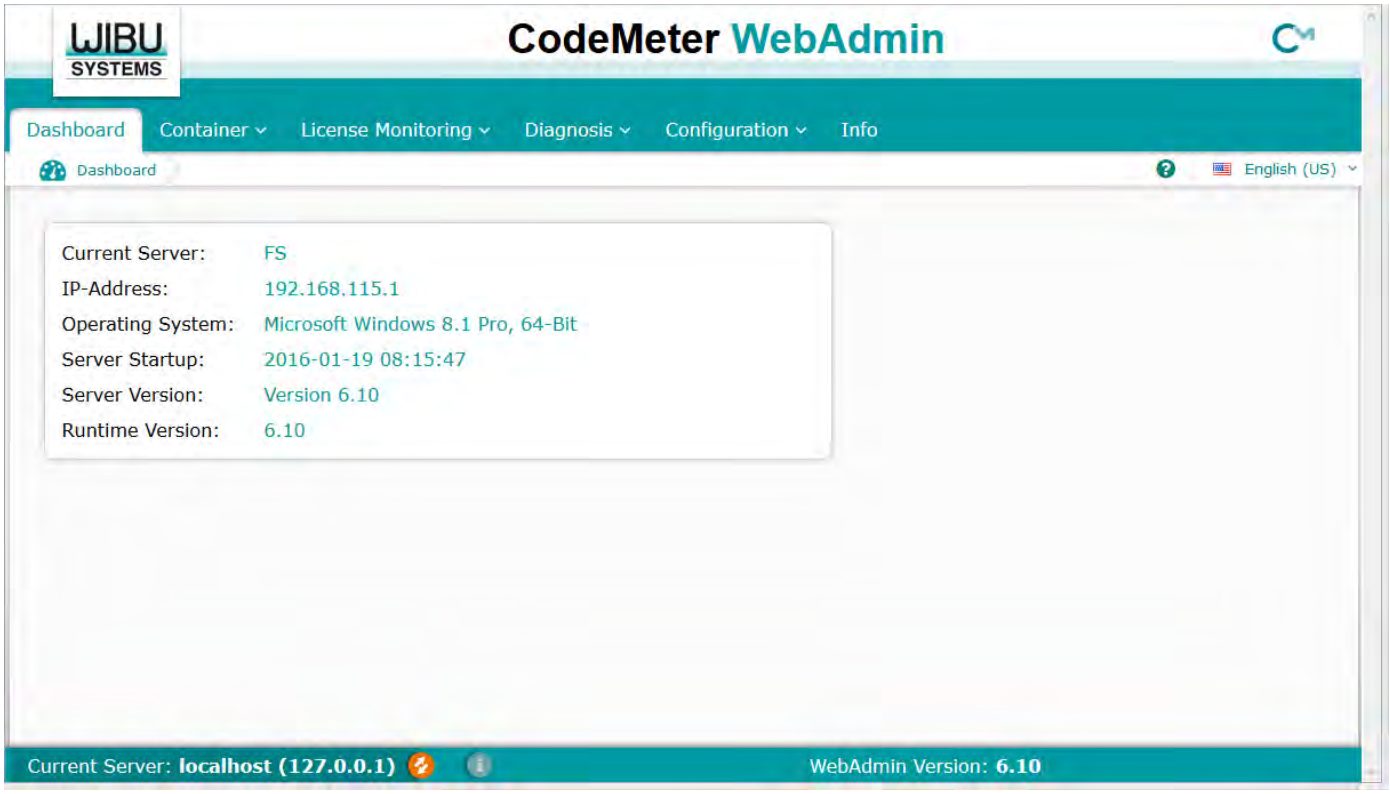


Figure 273: *CodeMeter WebAdmin* – Dashboard

Element	Description
Current Server	Name of the machine on which the service *CodeMeter License Server* has been started.
IP Address	Shows information on the network address in use.
Operating System	Shows information on the operating system in use.
Server Startup	Shows information on the start time of the server.
Server Version	Shows information on the *CodeMeter* version on the server.
Runtime Version	Shows information on the *CodeMeter* runtime in use.
Certificate expiration time	Optionally shows details on certificates expiration times, if HTTPS has been selected as protocol and self-signed certificates are used as defined on the WebAdmin configuration page 421.
	Certificate expiration time: Used self signed certificate expires on 11 Nov 17 12:00 UTC. The certificate will be automatically renewed at first server restart after 10 Jan 18 12:00 UTC. This certificate renewal will trigger a browser warning.

12.6.5 Container

Using the **Container** navigation item finds information on connected *CmContainer*. This covers:

- licenses[404] including detailed information on Firm Items[405] and Product Items[405].
- CmContainer Info[407]
- User data[408] including detailed information on Product Items[409]
- Backup and Restore[409]

> If you open a selected *CmContainer* using *CodeMeter Control Center*, only this single *CmContainer* displays. At the same time, the tab **Licenses** opens. This also is the case, if you select a single Container from the list of **All Containers**.
>
> Click **All Containers** to open a clear overview of all connected *CmContainer*.

12.6.5.1 Licenses

The expanding area **"Licenses"** displays all licenses the selected *CmContainer* holds.

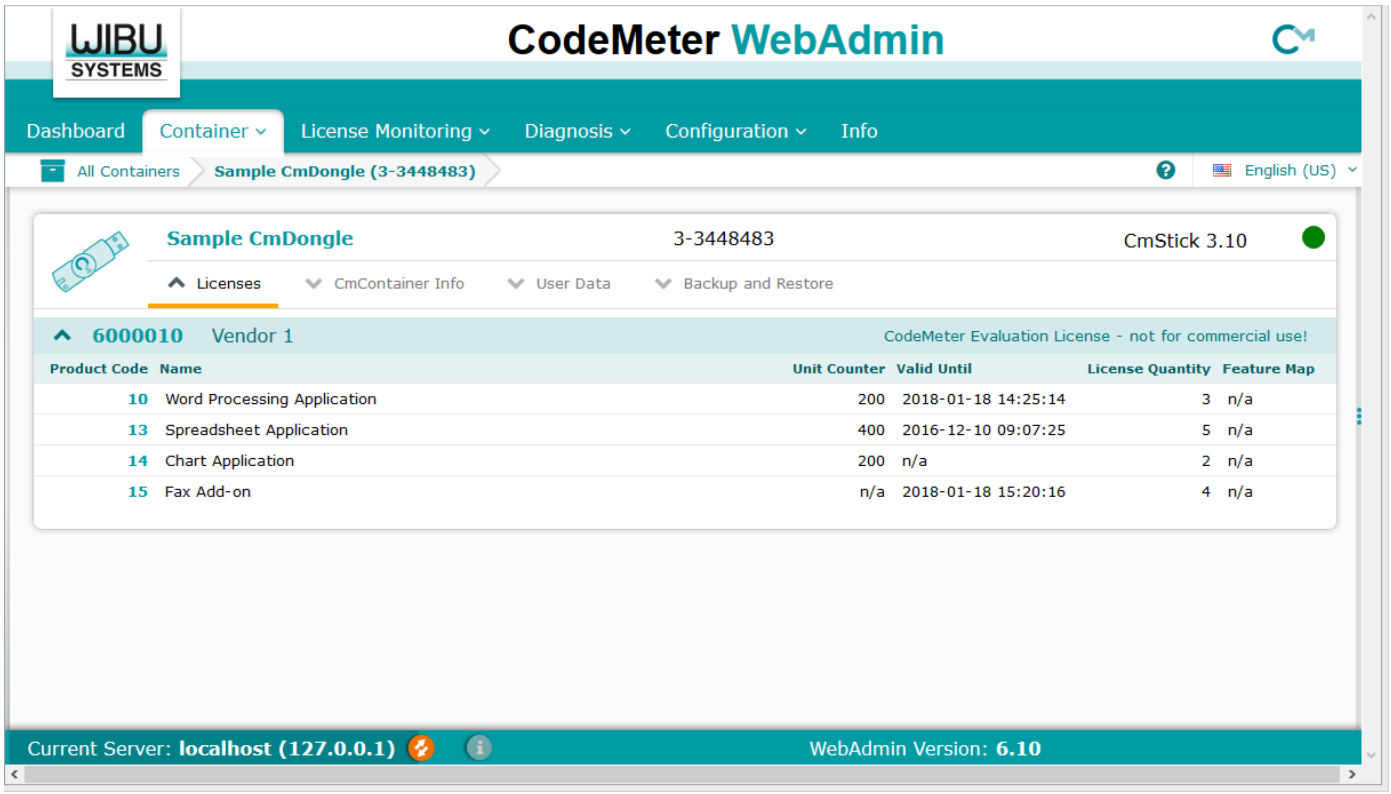


Figure 274: *CodeMeter WebAdmin* – CmContainer Details - Licenses

The display of licenses is ordered by different vendors. A vendor is uniquely identified by number value, the *Firm Code*, and a name. For example, in the figure above this is the *Firm Code* "600010" of "Vendor 1".

All related products, i.e. the licenses, are listed below the single licensor holding the respective *Product Code*, defined by a unique number value.

> - If a license is expired or an *Unit Counter* has reached a value of 0, the license entry is in red writing.
> - If you see the O H pictograms, then a license transfer[437] is involved with the respective *Firm Codes* or *Product Codes*.
>
> O inform on License Transfer Options
>
> H informs on the License Transfer History
>
> On hovering the pictograms a separate popup informs, if clicking the pictogram, the **Firm Item Detail**[405] - or **Product Item Detail**[405] page opens.
>
> - Arrow symbols attached to sub-Product Code items indicate that *Module Items*[448] exist.

In the figure above, this is, for example, the product "Spreadsheet Application" with the *Product Code* of 13 or the product "Chart Application" with a *Product Code* of 14. In addition, you obtain further information[399] on existing **Unit Counter**, **Valid Until** (Usage Period, Expiration Time, Activation Time), **License Quantity** and **Feature Map**.

Click on the highlighted *Firm Code*[405] entry for the display of more detailed information on the license conditions of products by a specific vendor.

Click on the highlighted *Product Code*[405],entry for the display of more detailed information on the license conditions of products by a specific vendor.

12.6.5.2 Firm Item Details

This page displays detailed information on:

- *Firm Item* options of the selected *CmContainer*

Element	Description
CmContainer	Name of the *CmContainer* including mask and serial number.
Firm Access Counter	Displays the *Firm Access Counter* reading. The *Firm Access Counter* (FAC) locates at the *Firm Item* level of a *CmContainer*. This counter allows to control whether a *Firm Item* can be used for encryption purposes. By default, the FAC is deactivated and has a value of `65535 (0xFFFF)`. It can be programmed by the software vendor to any other value.
Firm Update Counter	Displays the *Firm Update Counter* (FUC) reading. This counter is automatically incremented on programming.
Firm Precise Time	Displays the time of the *Firm Item*-programming.

- the licensing of products of a specific software vendor.

  In the following figure, for example, you see all licenses of "Vendor 1" (*Firm Code 600010*). Additional information[399] covers **Product Code**, **CmContainer** serial number, **Name**, **Unit Counter**, **Valid Until** (Usage Period and/or. Expiration Time), **License Quantity** and **Feature Map**.

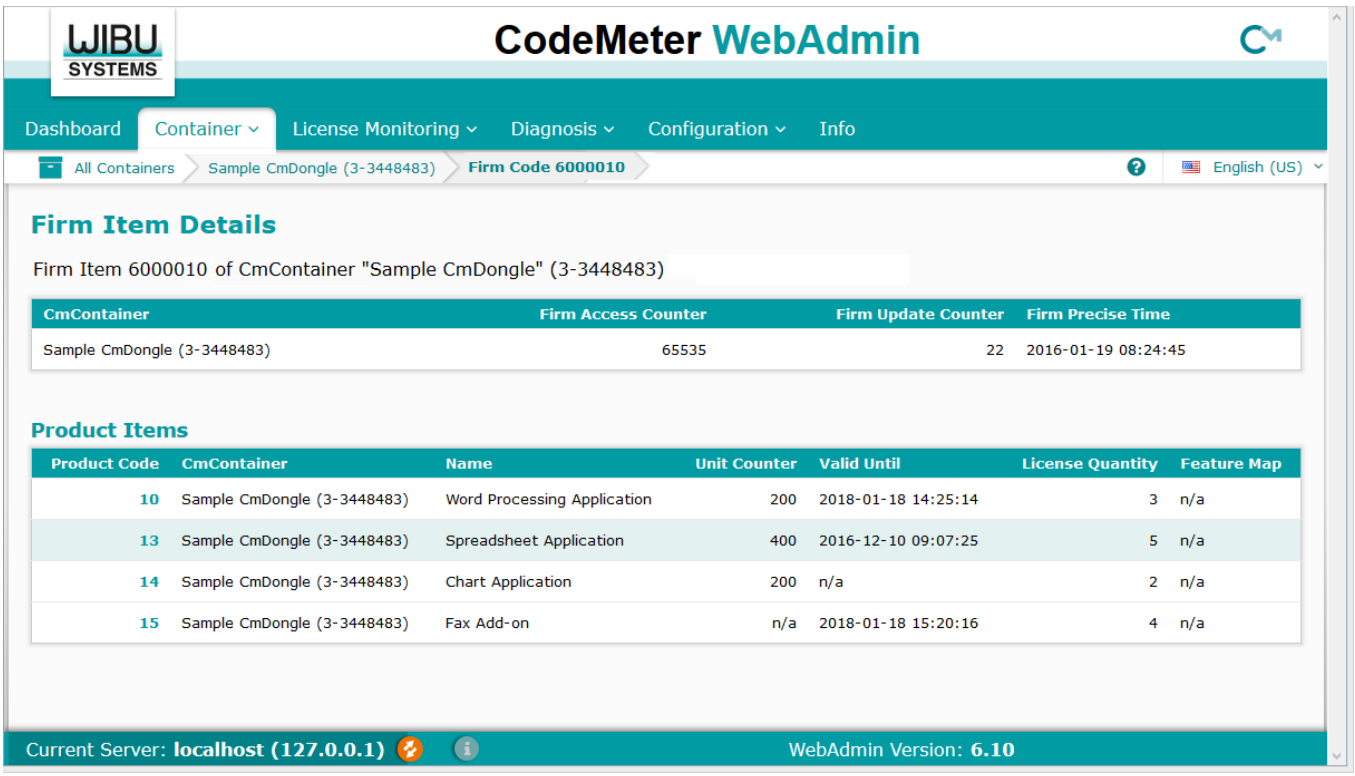


Figure 275: *CodeMeter WebAdmin* – Firm Item Details

12.6.5.3 Product Item Details

This page displays detailed licensing information of products of a specific vendor.

The following figure shows all available information[399] on the product "Word Processing Application" with a *Product Code* "10" of the software vendor at the *Firm Item* level with a *Firm Code* of "10".

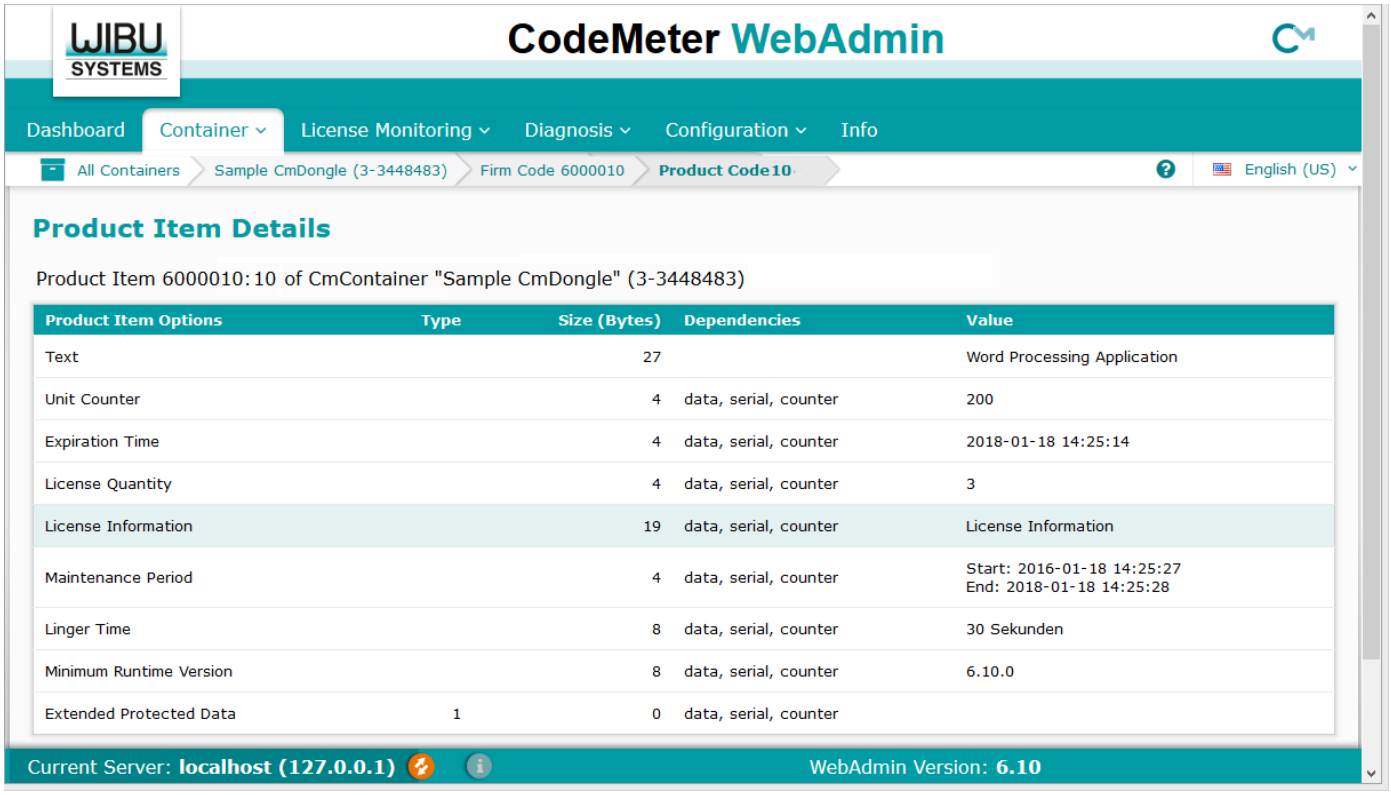


Figure 276:*CodeMeter WebAdmin* – Product Item Details

Element	Description
Product Item Options	In the first column you see the *Product Item Options*.These are license properties set by the licensor. For illustrative reason the figure lists all options. When listed in other cases, not all of these options 399 are always displayed. In the figure above you see that the license has been borrowed for the local use.
Type	If the license properties represent data fields, the column informs in which area of the *CmContainer* these fields are located.
Size (Bytes)	The column the number of bytes a listed license property allocates.
Dependencies	The column informs whether a licensor has set dependencies for the programming sequence of the *CmContainer*.
Values	The final column displays the stored value of the single license property.

The license properties as displayed in the figure above are not always set. The display of your license may differ.

12.6.5.4 CmContainer Info

The expanding area **"CmContainer Info"** displays information on the selected *CmContainer*.

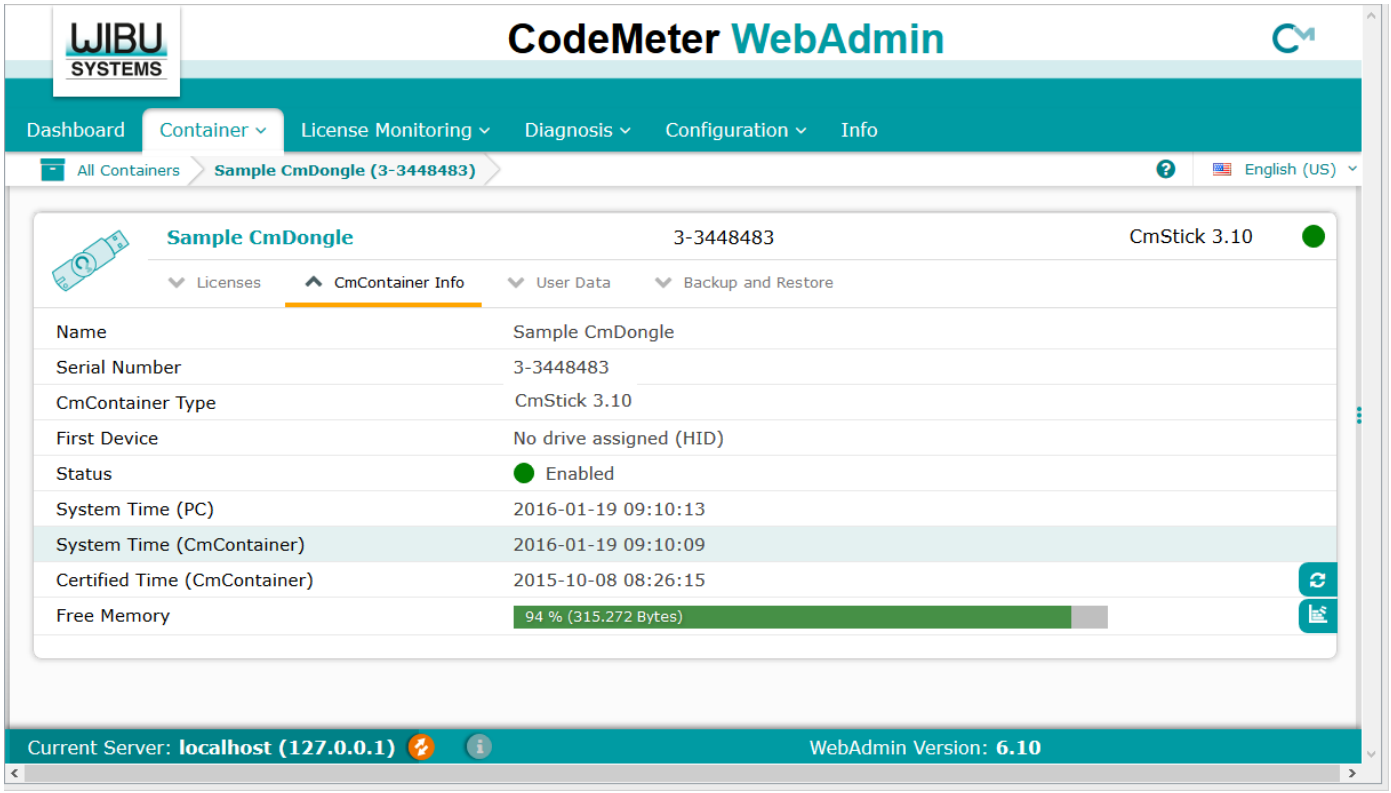
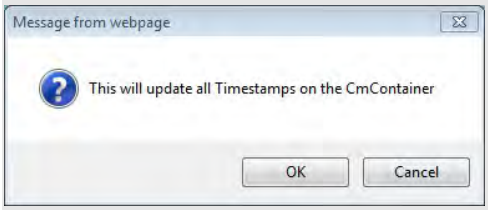


Figure 277: *CodeMeter WebAdmin – CmContainer* Info

The following information and elements are available.

Element	Description
Name	Shows the **Name** of the selected *CmContainer*. If you want to change the name of your *CmContainer*, use *CodeMeter Control Center*.
Serial Number	Shows the **Serial Number** of the selected *CmContainer*.
CmContainer Type	Shows the **Type** of the selected *CmContainer*.
First Device	Shows the drive information of the selected *CmDongle* if configured as Mass Storage Device. Alternatively to the Mass Storage Device status, the *CmDongle* can display as HID without a drive status. Each status can be configured 461. The drive size is only displayed in the case of *CmDongles* with Flash memory.
Status	Shows the current activation status of the selected *CmContainer*. The following status settings are displayed: • **Disabled**: The connected *CmContainer* is deactivated and not usable by any application. • **Enabled until Unplugged**: The *CmDongle* is activated as long as it is connected and supplied by electrical energy. After removed from the PC the *CmDongle* is automatically deactivated. • **Enabled**: The *CmContainer* is fully activated. If a *CmDongles* is removed, the license access is still possible after plug out. • **Empty**: The *CmContainer* is empty and must yet be activated (*CmActLicense* - LIF). You change the activation status of a *CmContainer* using *CodeMeter Control Center* 387. Wibu-Systems recommends the activation status **"Enabled until Unplugged"** when using *CmDongles*. This ensures that even when a ***CmDongles*** is lost, unauthorized access to the licenses and personal data in the ***CmDongle*** is lost, unauthorized access to the licenses and personal data is not possible.
System Time (PC)	Shows the **System Time** (local time on the PC) when the service *CodeMeter License Server* has started.
System Time (CmContainer)	Shows the saved **System Time** (internal time) of the *CmContainer*. These two system times may differ due to the pending synchronization process.
Certified Time (CmContainer)	Shows the *Certified Time* saved in the *CmContainer*. In order to update the *Certified Time* of your *CmContainer* using a *CodeMeter* Time Server, click the button. This action is confirmed by a dialog.

Element	Description
	 Figure 278: *CodeMeter WebAdmin* - Update Certified Time
Free Memory	Shows the **Free Memory** of the SmartCard chips of a *CmDongle*, i.e. how much space is available for the programming of additional license entries.
Defragment	Click the button to defragment the memory of the *CmDongle* chip.

12.6.5.5 User data

The expanding area **"User Data"** displays detailed information[399] on products (licenses) the user is able to use only explicitly with the personal *CmDongle* Password. The *Firm Code* in this case has a value of "0".

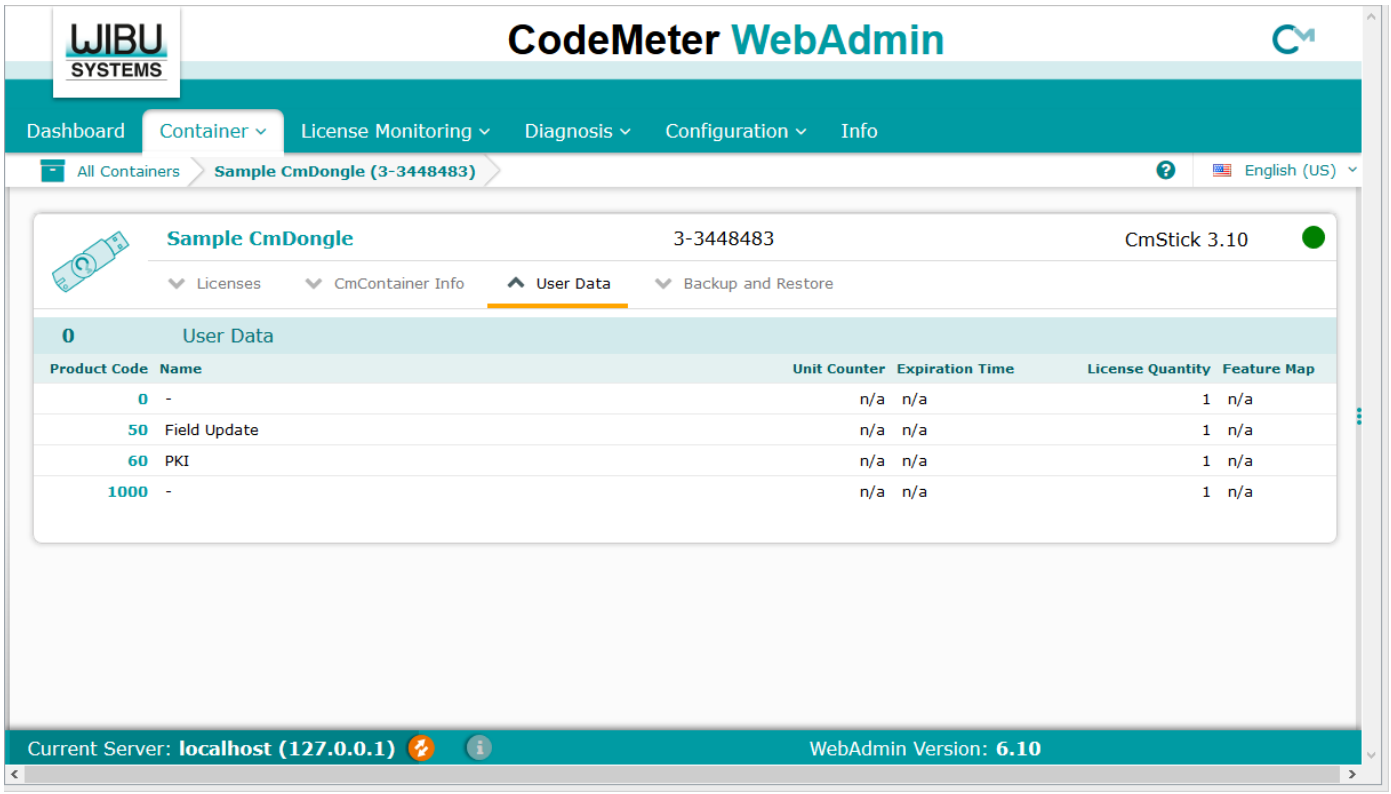
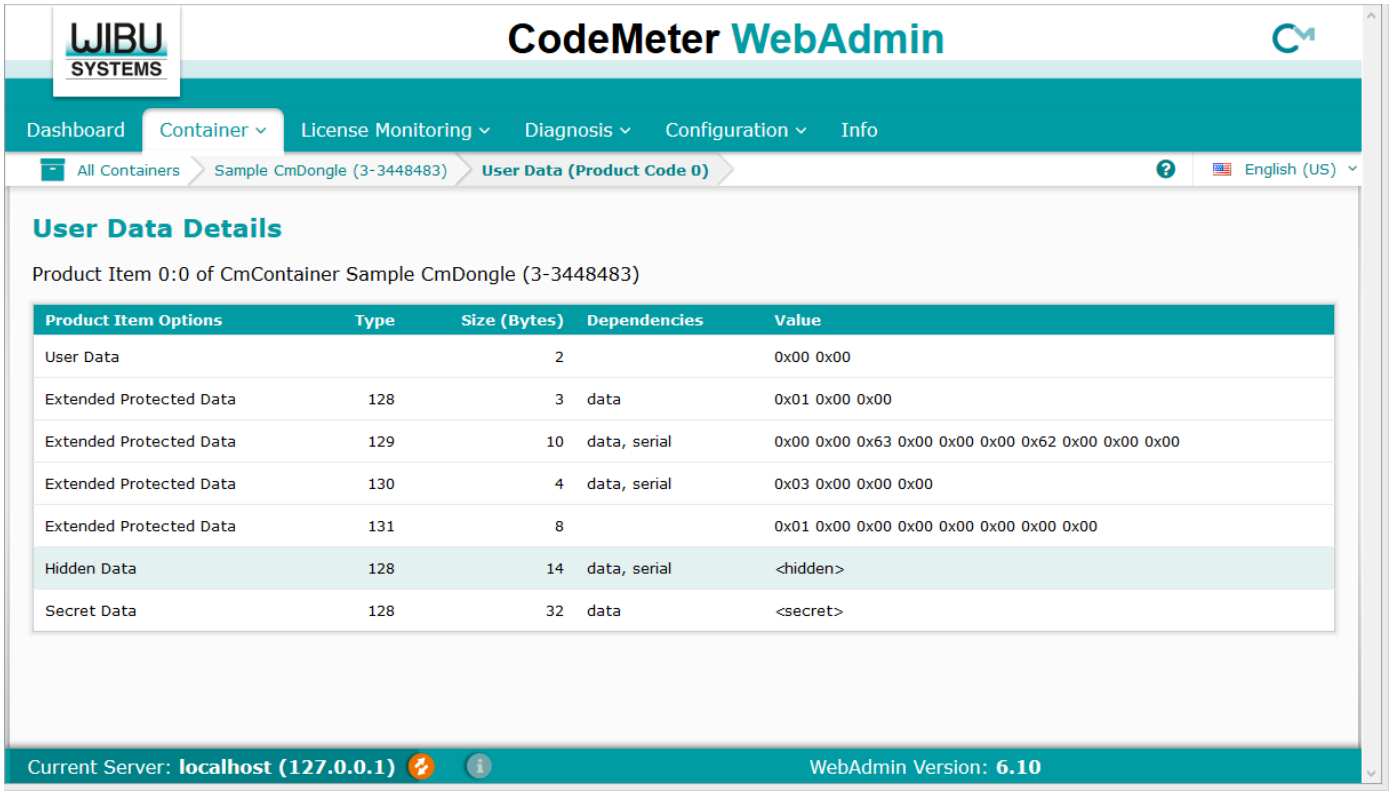


Figure 279:*CodeMeter WebAdmin* – User Data

12.6.5.6 User Data Details

The **"User Data Details"** page displays detailed Product Item information[399] on products (licenses) the user is able to use only explicitly with the personal *CmDongle* Password.



Product Item Options	Type	Size (Bytes)	Dependencies	Value
User Data		2		0x00 0x00
Extended Protected Data	128	3	data	0x01 0x00 0x00
Extended Protected Data	129	10	data, serial	0x00 0x00 0x63 0x00 0x00 0x00 0x62 0x00 0x00 0x00
Extended Protected Data	130	4	data, serial	0x03 0x00 0x00 0x00
Extended Protected Data	131	8		0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00
Hidden Data	128	14	data, serial	<hidden>
Secret Data	128	32	data	<secret>

Figure 280: *CodeMeter WebAdmin* – User Data Details

12.6.5.7 Backup and Restore

The expanding area **"Backup and Restore"** allows you to save personal data located in your *CmDongle*, and restore them in the *CmDongle*.

This area does not display for the Container type *CmActLicense*.

Note, that the backup and restore mechanism only comprise the user data in the *CmDongle* but no license information of other licensors. Backup and restore exclusively relates to the license container with the *Firm Code* "0".

In order to restore licenses which do not locate in the personal area (*Firm Item* levels unequal to *Firm Code* "0"), please contact Wibu Support.

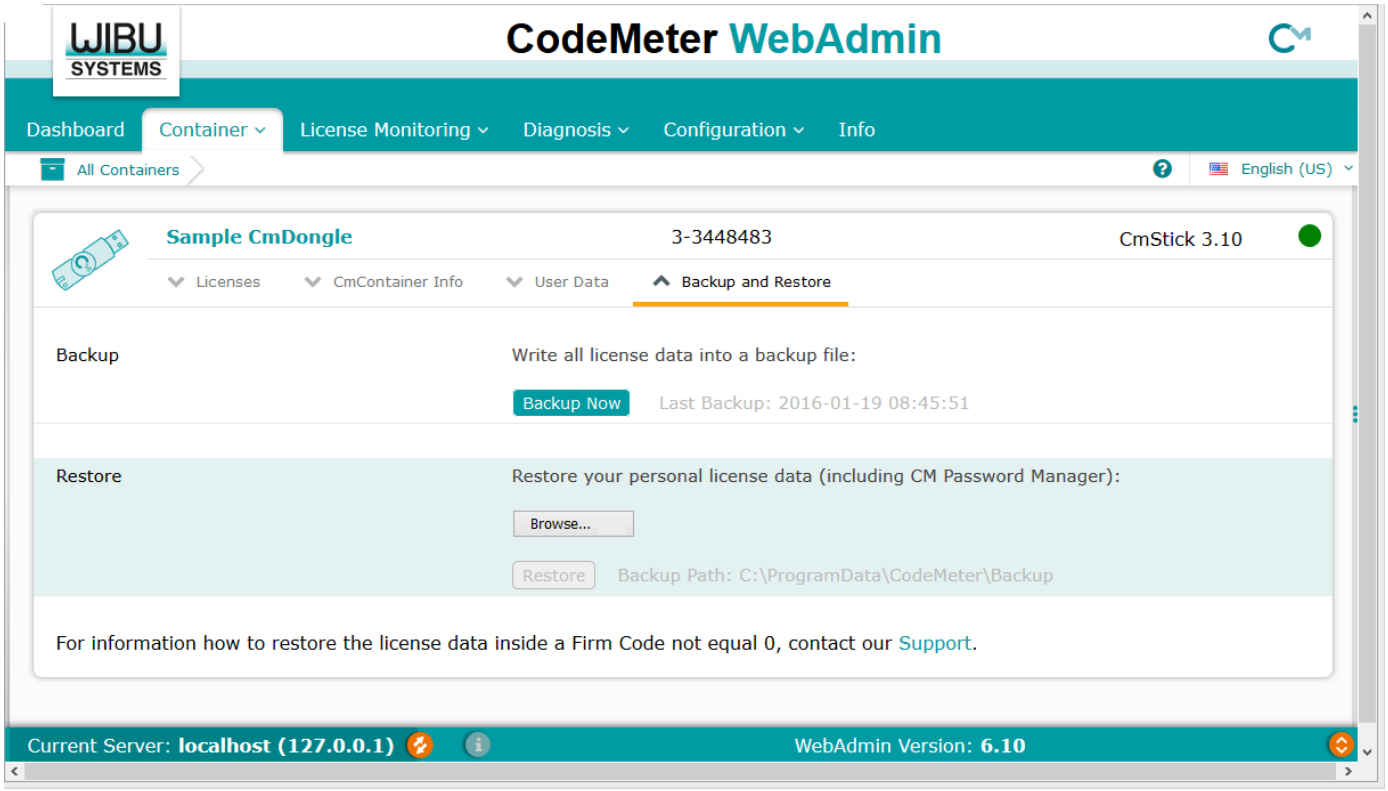
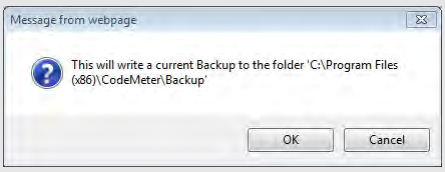
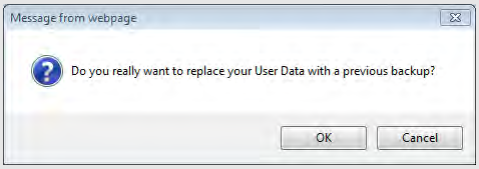


Figure 281: *CodeMeter WebAdmin* - "Backup and Restore"

Area	Description
Backup	**1.** Click the **"Backup now"** button to apply an instant backup of your personal *CmDongle* data (user data). In addition, the time of the **Last Backup** is displayed. **2.** Confirm the following dialog to create the backup file. 
Restore	**1.** Click the **"Browse"** button to select the backup copy which is to be restored. The location of the backup file displays. **2.** Click the **"Restore"** button to start the restoring process. **3.** Confirm the following dialog and click the **"OK"** button.  If you import a backup into the *CmDongle*, all changes after the backup was created are lost. **4.** Enter the Password of the *CmDongle* in which the backup file is to be imported. 

> You are also able to import the saved data into another *CmDongle*. Please note, however, that the second *CmDongle* must have the same password !

12.6.6 License Monitoring

The **"License Monitoring"** page displays all existing licenses and their allocation ordered by licensors and related licenses.

Next to describing information on **Product Code**, **Name**, and **Feature Map**, the column **License Quantity** shows the respective total number of **available** licenses..

Shared and Available Licenses

In addition, an expandable area structures the licenses according to access modes (**User Limit (Borrowed)**, **No User Limit**, **Exclusive**, **Shared**) and shows **available** licenses.

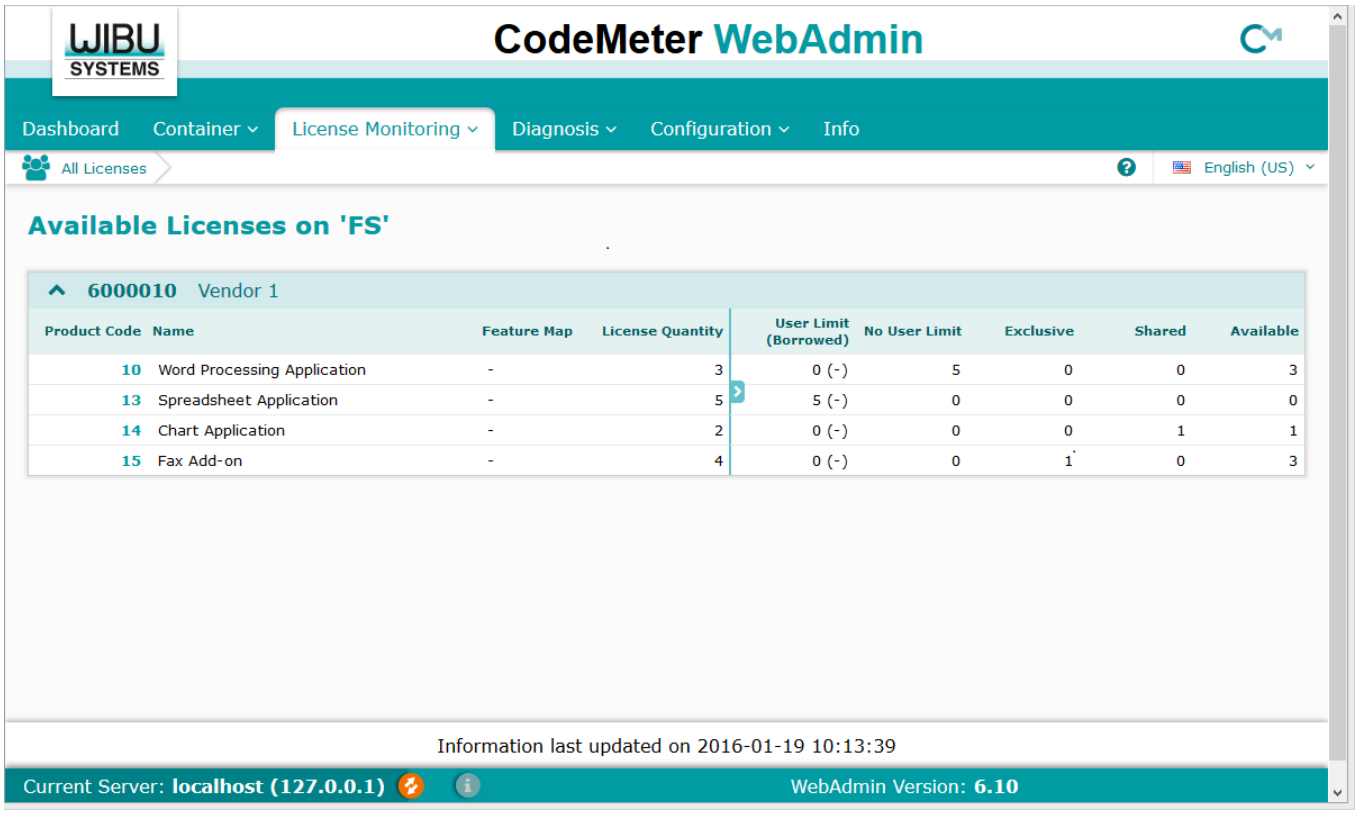


Figure 282: *CodeMeter WebAdmin* – "License Monitoring | Available"

Element	Description
User Limit	Here each started instance allocates a single license. It does not make a difference if the *CmContainer* was found locally, or on a network. In brackets the number of borrowed licenses display, if existent.
Station Share	Here multiple instances can be started on a single PC but allocate only a single license. You use this setting, for example, when you want to provide the end-user with the option of starting the application several times. On a terminal server each session allocates a license. In virtual machines each machine allocates a license.
Exclusive	Here a protected application can be started only once on a PC.
No User Limit	Here any number of instances of the protected application can be started locally or in a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used.

12.6.6.1 License Monitoring Details

The **"License Monitoring Details"** page displays detailed information on the license allocation.

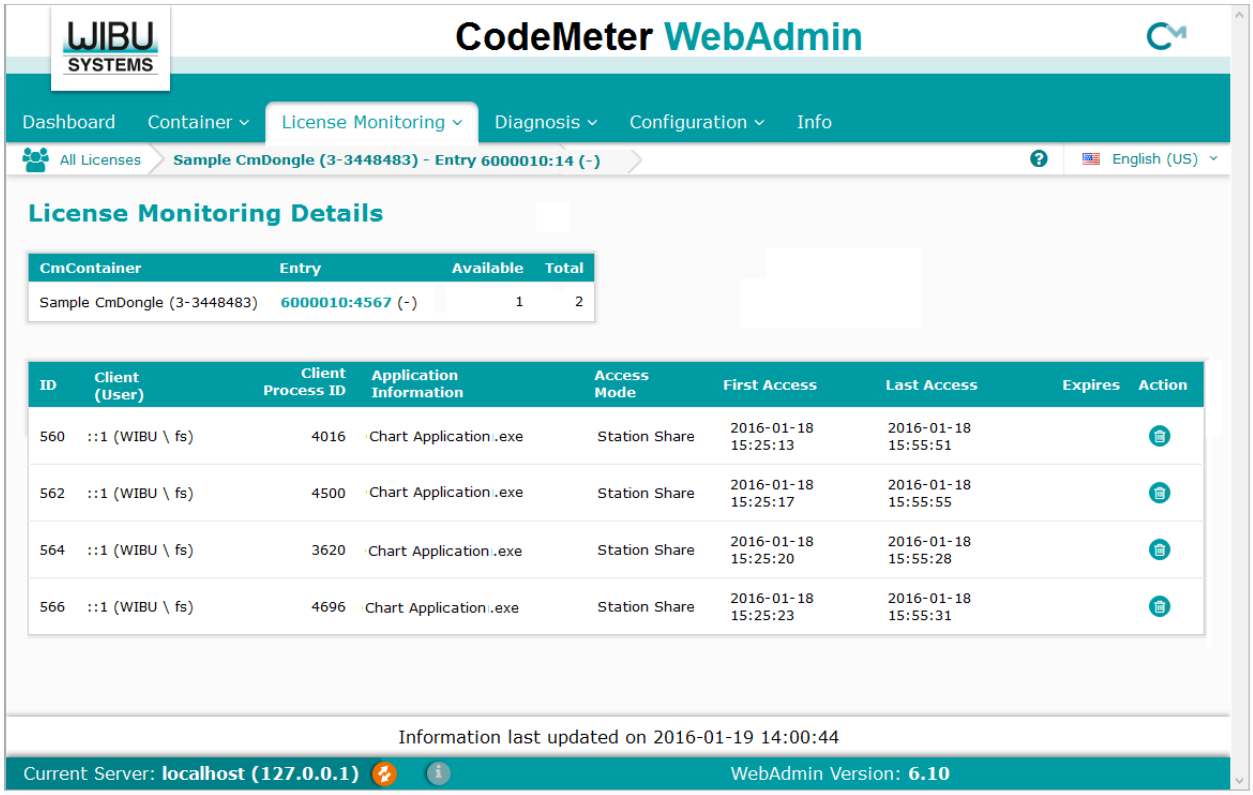


Figure 283: *CodeMeter WebAdmin* – "License Monitoring | License Monitoring Details"

For example, in the figure above you see:

- the licenses for the application derive from the licensor with the *Firm Code* 600010 and describe the product with the *Product Code* 14.
- the licenses are stored in the *CmContainer* with the mask and serial number `3-3448482`.
- in total 1 client, identified by **ID**, **Client** (`::1(WIBU \ fs)`) and **Client Process ID** columns, 4 times accessed the application "Chart Processing" using Station Share. Here multiple instances can be started on a single PC but allocate only a single license. An access from another PC would be possible as the upper table (Total 2, Available 1) shows.
- There is no expiration date.
- Client `::1(WIBU \ fs` for the first time accessed the application (**First** and **Last Access** columns are of same date).
- Using the pictogram of the **Action** column allows you to deallocate single accessed licenses..

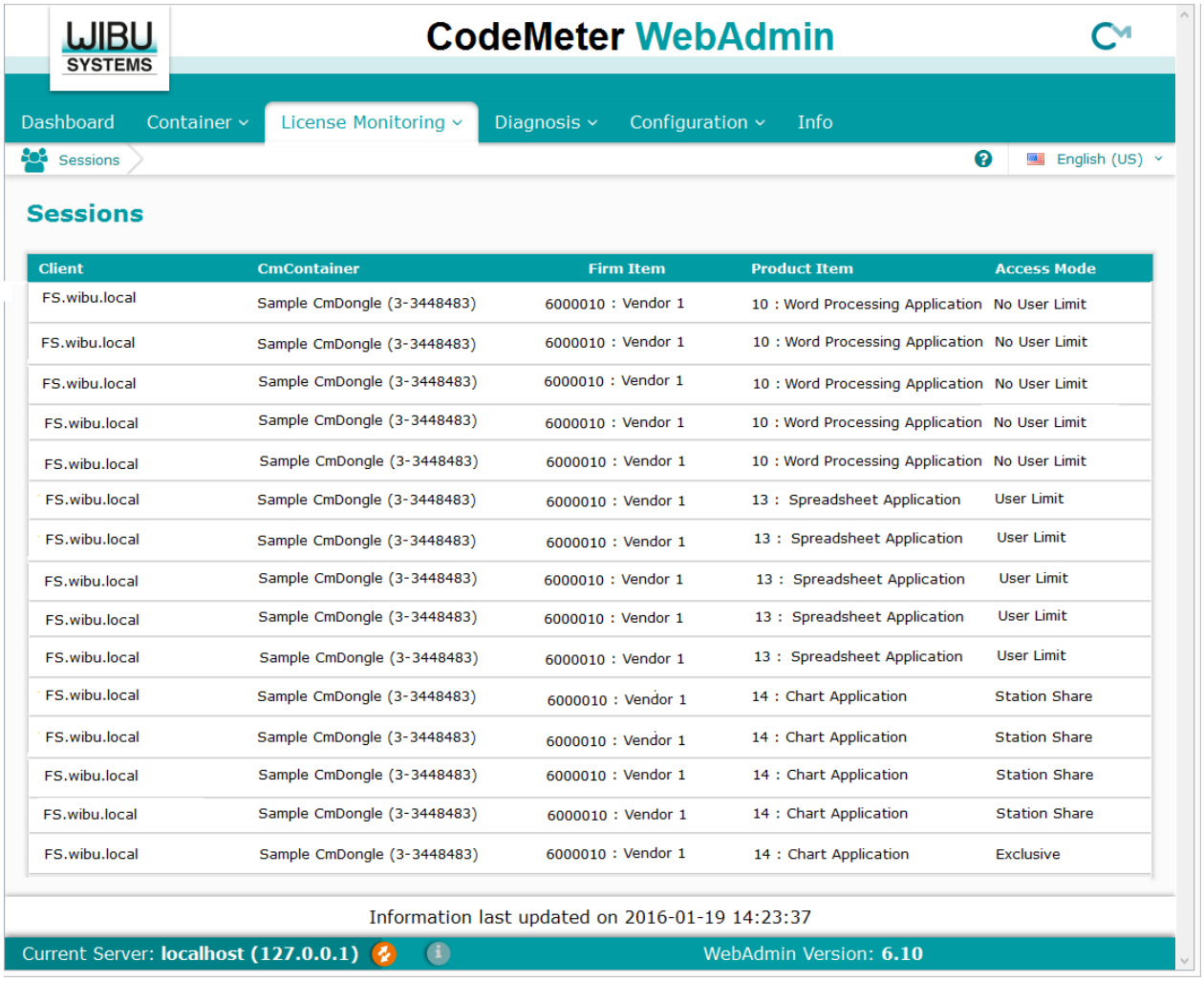
> You cannot deallocate and reallocate borrowed licenses before they have been returned.

For example, this is necessary when all licenses are allocated but an additional instance of the application needs to be started.

> After deleting of an access the license is deallocated and available again. The client of the application receives a respective error message.

12.6.6.2 Sessions

The **"License Monitoring | Sessions"** page displays all existing licenses ordered by users actually logged on (**Clients**).



Sessions

Client	CmContainer	Firm Item	Product Item	Access Mode
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	10 : Word Processing Application	No User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	10 : Word Processing Application	No User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	10 : Word Processing Application	No User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	10 : Word Processing Application	No User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	10 : Word Processing Application	No User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	13 : Spreadsheet Application	User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	13 : Spreadsheet Application	User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	13 : Spreadsheet Application	User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	13 : Spreadsheet Application	User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	13 : Spreadsheet Application	User Limit
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	14 : Chart Application	Station Share
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	14 : Chart Application	Station Share
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	14 : Chart Application	Station Share
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	14 : Chart Application	Station Share
FS.wibu.local	Sample CmDongle (3-3448483)	6000010 : Vendor 1	14 : Chart Application	Exclusive

Information last updated on 2016-01-19 14:23:37

Current Server: **localhost (127.0.0.1)** WebAdmin Version: **6.10**

Figure 284: *CodeMeter WebAdmin* – "License Monitoring | Sessions"

Here you obtain all describing information 399 on the **CmContainer**, licensor (**Firm Item**), license (**Product Item**), and **Access Mode**.

12.6.6.3 License Tracking

The **"License Monitor | License Tracking"** page allows you to track who, when, from where, how often uses server licenses of *CodeMeter*-protected applications.

For Windows operating systems you find the profiling entries stored in the registry, for other operating systems entries are set in the file `server.ini`. The following table shows you the respective locations.

Operating system	Registry / Server.ini Entry
Windows	`HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini`

There exist two relevant profiling entries for *License Tracking*.

Entry	Property	Value
`LogLicenseTracking`	`[DWord]`	`[0;1]` Default value is is `0` and Logging for License Tracking is disabled.
`LogLicenseTrackingPath`	`[SZ]`	`<path>` Default path on Windows operating systems is `%ProgramData%\CodeMeter\LicenseTracking`. For other operating systems the default path has the same value of the general profiling entry `LogPath`.

> Please note that you must stop the *CodeMeter License Server* service, make the change, and then restart the *CodeMeter License Server* service before the change can take effect.

On the basis of selectable *Firm Codes*, log files and licenses, accesses are displayed graphically and in detail. The created report may serve to use information on license requests and denials for saving license costs and create forecasts or prognoses.

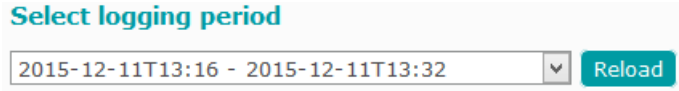
Using a separate navigation the number and origin of allocated, rejected or released licenses can be tracked according to specified view modes (month, day, hour). Clicking on the displayed bars shows more details on the use of licenses.

For using license tracking, please proceed as follows:

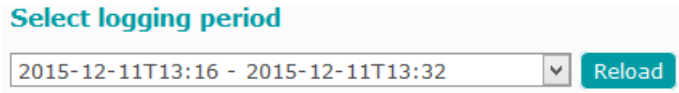
1. Select the desired *Firm Code* using the field "**Select Firm Code**".

   Click the "**Reload**" button to update the displayed *Firm Codes*.
2. Select the log file using the field "**Select logging period**".

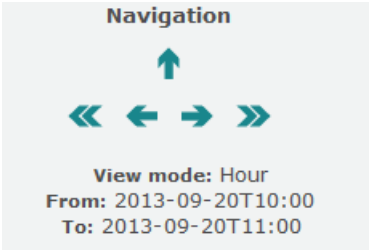
   

   Click the "**Reload**" button to update the logging period entries.
3. Select the license to be tracked using the field "**Select license**".

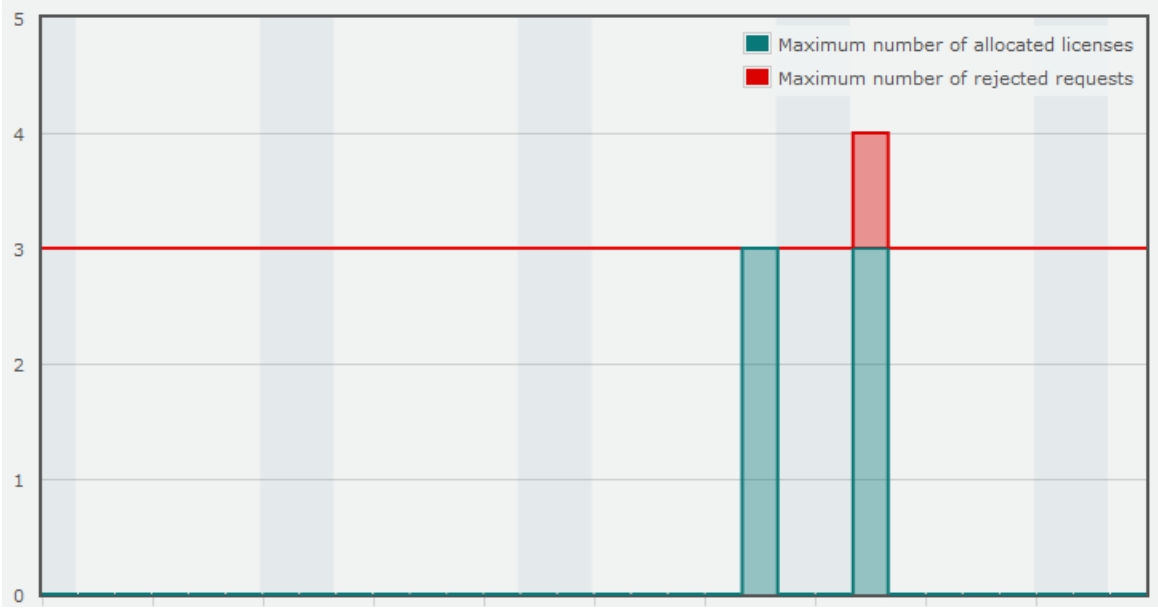
4. Click the button "**Create report**".

   The separate area **Navigation**:

   - informs on the view mode (Month, Day, Hour),
   - shows the tracked period (From - To),
   - allows to browse back and forward in time periods and switch back to the previous view mode.

   Below the selection area a **bar chart** displays showing the maximum number of allocated licenses and rejected requests over time.

   The default is set to the view mode month.

5. Move over a colored bar to open an over-layered dialog for information display.

6. Click left to change to view mode **Day**.
   For switching back to the view mode **Month** you may use the arrow symbol in the **Navigation** area.
7. Move over a bar again tom switch to the view mode **Hour**.

8. Move over a bar again and left click to open the separate **Details** area.

Detailed information and separate tables for single bars list details on **Active Users**, **Rejected Requests** and **All Events**.

Details

Period: **2013-10-10T15:09:00 - 2013-10-10T15:09:59**
Maximum number of allocated licenses: **3**
Maximum number of rejected requests from different users: **1**

Active Users (ID, Client, User)

Active Users

ID	Client	User
57	10.49.12.17	wv
58	10.49.12.17	wv
59	10.49.12.17	wv
60	10.49.12.17	wv
61	10.49.12.17	wv
62	10.49.12.17	wv
63	10.49.12.17	wv
64	10.49.12.17	wv
65	10.49.12.17	wv
66	10.49.12.17	wv
67	10.49.12.17	wv
68	10.49.12.17	wv

Rejected Requests (Second, Event Type, Client, User)

Rejected Requests

Second	Event Type	Client	User
26	Denial	10.49.12.17	wv
28	Denial	10.49.12.17	wv
30	Denial	10.49.12.17	wv
34	Denial	10.49.12.17	wv
36	Denial	10.49.12.17	wv

All Events (Second, Event Type, ID, Client, User)

All Events

Second	Event Type	ID	Client	User
3	Denial		10.49.12.17	wv
5	Access	60	10.49.12.17	wv
5	Release	58		
5	Release	59		
7	Access	61	10.49.12.17	wv
7	Access	62	10.49.12.17	wv
11	Release	60		
13	Access	63	10.49.12.17	wv
13	Denial		10.49.12.17	wv
13	Release	62		
16	Access	64	10.49.12.17	wv
16	Access	65	10.49.12.17	wv
20	Release	63		
22	Access	66	10.49.12.17	wv
22	Release	64		
22	Release	65		
24	Access	67	10.49.12.17	wv
24	Access	68	10.49.12.17	wv
24	EOF			

The detail view uses the following elements:

Element	Description
ID	uniquely discerns requesting / accessing processes.
Client	identifies the IP address of the requesting / accessing machine.
User	identifies the user requesting / accessing the license.
Second	informs on the second time value.
Event Type	**Denial** describes that a user requested a license but did not get one because no more licenses could be allocated. It will not show license requests of licenses that do not exist on this server.
	Access describes that a license on a server is allocated to a user.
	Release describes that a user has released a formerly accessed license on a server.

12.6.7 Diagnosis

The **"Diagnosis"** page allows you to log all events related to the *CodeMeter License Server* service .

12.6.7.1 Events

The **"Diagnosis | Events"** page provides information which supports you in detecting eventually occurring errors by viewing events.

CodeMeter WebAdmin displays a protocol only if you previously activated 387 this function in *CodeMeter Control Center*.
There you find further information on how to save the log file.

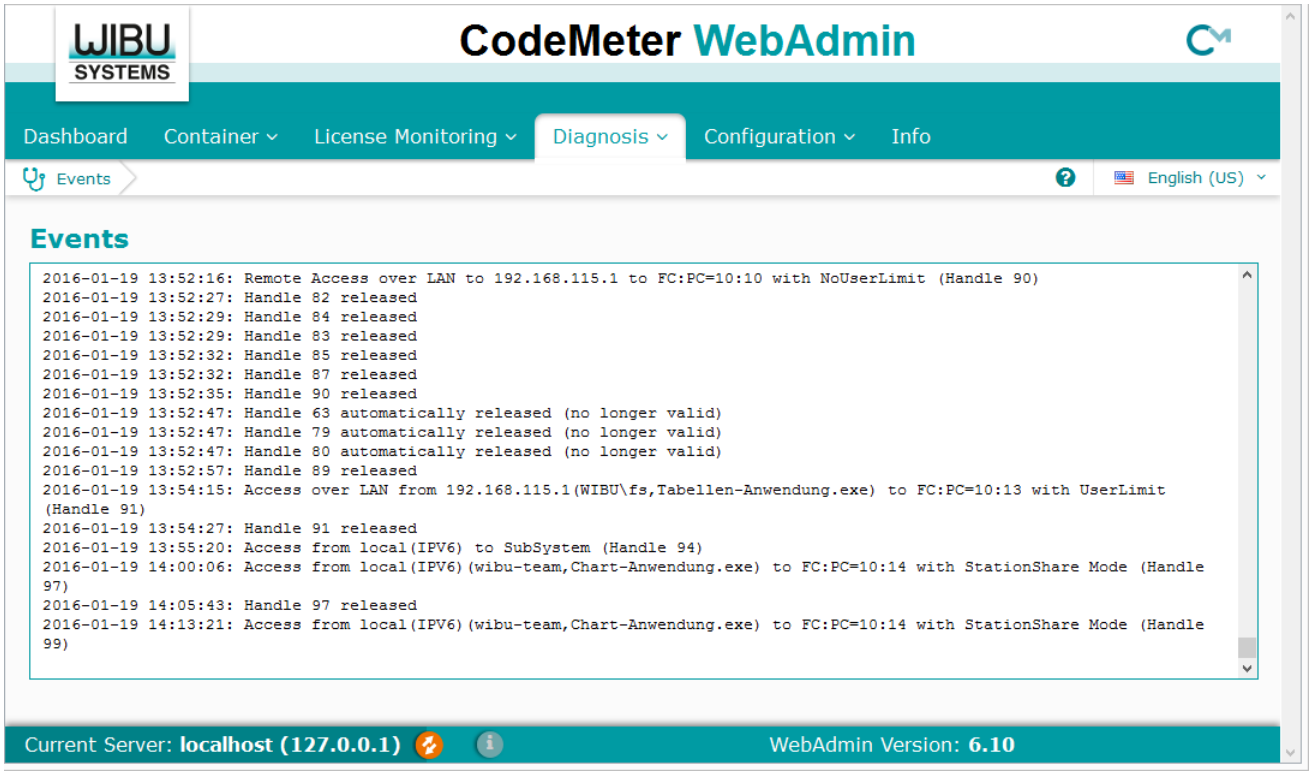


Figure 285: *CodeMeter WebAdmin* – Diagnosis | Events

12.6.8 Configuration

Using the **Configuration** navigation item allows you to configure the following settings:

- Basic[417]
- Server[425]
- Advanced

12.6.8.1 Server Search List

The **"Configuration | Basic | Server Search List"** page allows to define access to and order of installed[425] *CodeMeter* network LAN and WAN (Wide Area Network) server.

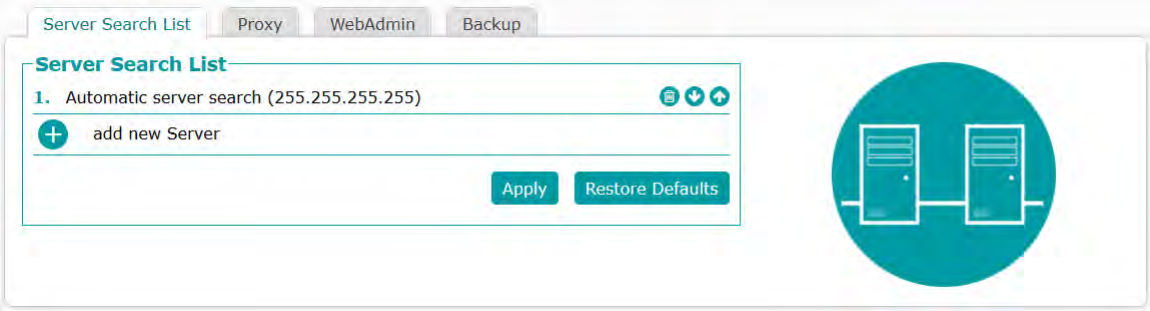
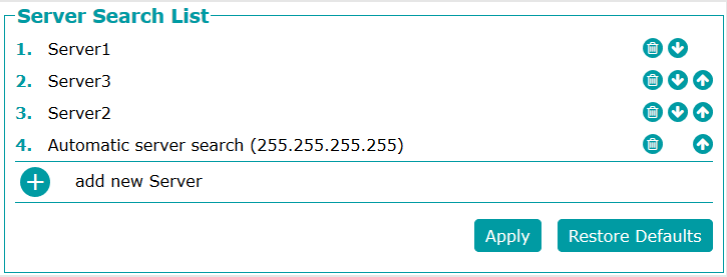


Figure 286: *CodeMeter WebAdmin* – "Configuration | Network"

The **Server Search List** is used to define the access to and order of *CodeMeter* network LAN and WAN (Wide Area Network) servers.

By default, the **Server Search List** holds the entry "**Automatic server search (255.255.255.255)**". Then licenses are sought first locally and then in the network (subnet). Using the entry "**add new Server**" allows you to decisively add single target server addresses.

Element	Description
Server Search List	You edit the **Server Search List** by using the respective **add new Server**, **remove"** buttons. You can also change the order by using the **up** and **down** buttons. You save the changes you made by using the **"Apply"** button.



You set back the settings of the server search list using the **"Restore Defaults"** button.

Alternatively, you are also able to set the Server Search List using the configuration files `CodeMeter.ini` or `Server.ini`. The table below shows you where to find the respective files.

Operating System	Configuration File
Windows	`%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini`
macOS	`\Library\Preferences\com.wibu.CodeMeter.Server.ini`
Linux	`\etc\wibu\CodeMeter/Server.ini`

In the separate section `[ServerSearchList]` define the server as the exampe below shows:

```
[ServerSearchList]

[ServerSearchList\Server1]
Address=184.45.89.5

[ServerSearchList\Server2]
Address=185.55.78.6
```

When you define network settings, in some cases, this requires the restart of the *CodeMeter* service. However, you do not have to eject or deactivate the *CmContainer*. After you specified the settings you are able to stop and then restart the *CodeMeter* service in *CodeMeter Control Center* 388 . For non-Windows operating systems see here 384
.

In order to check for a successful connection, on the **"Home"** page click the **"Host Name"** button and look for the successful appending of the PC as server. The check works also by opening *CodeMeter Control Center* on the clients and the server and looking for the communication status in the respective "Events" tabs.

If a connection is still not established specify on the client PCs the server IP address.

Using in a local area network (LAN):

By specifying the PC names or IP addresses you define that the client requests exactly address the defined *CodeMeter* network server. This increases the performance on the network.

If the *CodeMeter* network server is located in another subnet, you should always specify the IP address in the server search list in order to preclude UDP broadcast problems.
By default, *CodeMeter License Server* binds to the first network adapter found.

Using in a wide area network (WAN):

Please note, that a WAN connection has to be provided by the Software Vendor.

Specify the IP address(es) for client requests to the defined *CodeMeter License Server* in the WAN.

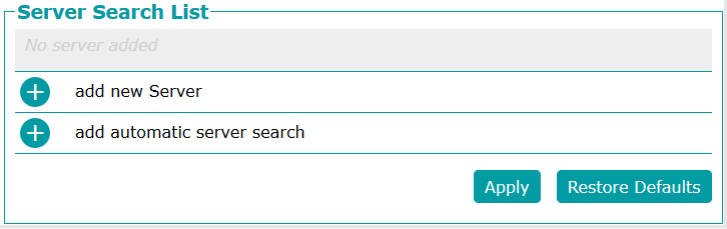
When specifying the IP address(es) please note that you are required to prefix a "`https:\\`"needed for the secured communication with a reverse proxy in the WAN.

add automatic server search

This entry allows to automatically search licenses on servers, first locally and then in the network (subnet). A so-called broadcast is performed.

If *CodeMeter WebAdmin* finds the entry 255.255.255.255, it is displayed as "**Automatic server search (255.255.255.255)**" on the list.

The entry is added at the end of the list.
If the list holds the entry as last entry, using "**add new Server**" the new Server is added as second last entry, i.e. without changes the entry "**Automatic server search (255.255.255.255)**" always remains at the end of the list.
If the list holds the entry other than the last, using "**add new Server**" the new Server is added as last entry.



12.6.8.2 Proxy

On the **"Configuration | Basic | Proxy"** page you define settings when using a proxy server. You require a proxy server, for example, coupled with:

- Firmware update
- update of the *certified time* using a Wibu-Systems time server
- access via CmWAN on a remote *CodeMeter* server.

By default, the **system proxy settings** apply.

Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

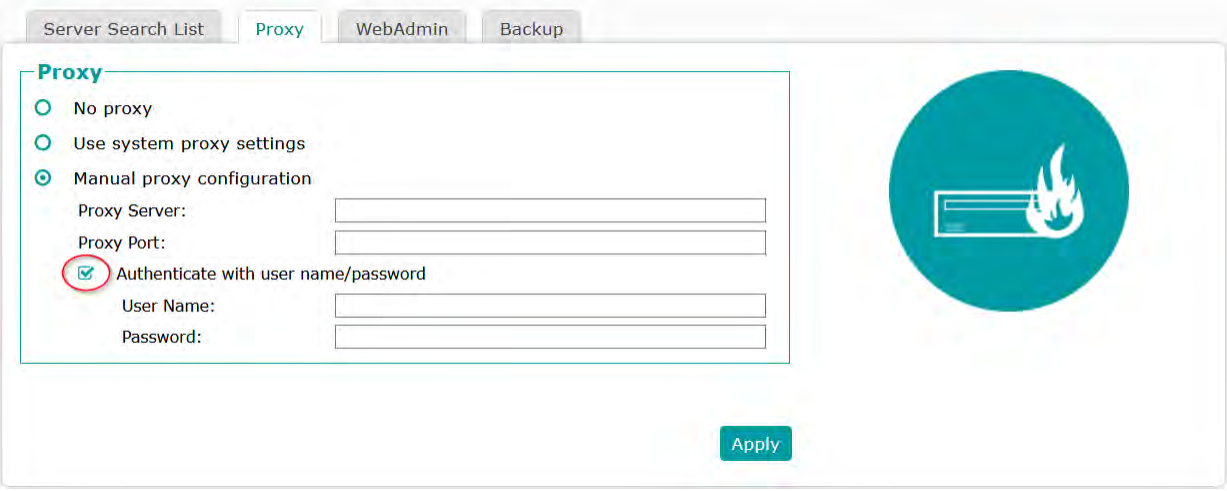


Figure 287: *CodeMeter WebAdmin* - "Configuration | Proxy"

You have several configuration options:

a) No proxy
The client server network communication is direct.

If you use a proxy, the following alternatives exist:

b) using the **proxy system settings**
Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

c) Manual proxy configuration
Please proceed as follows:

1. Specify the **Proxy Server** as IP address or DNS name.
2. Specify the **Proxy Server** as IP address or DNS name.

Authentication is automatically handled.

For explicitly using credentials for authenticate access to the proxy server, please proceed as follows:

1. Check the **"Authenticate with user name /password"** box.
2. Specify the **User Name**.
3. Specify the **Password**.

You may also access these settings using cmu 453 and the profiling 375.

Click the **"Apply"** button to save changes.

12.6.8.3 WebAmin

On the **"Configuration | WebAdmin"** page you defining settings to manage the access (local and remote), select the desired protocol (HTTP or HTTPS), and specify the eventually required authentication credentials for accessing *CodeMeter WebAdmin*.

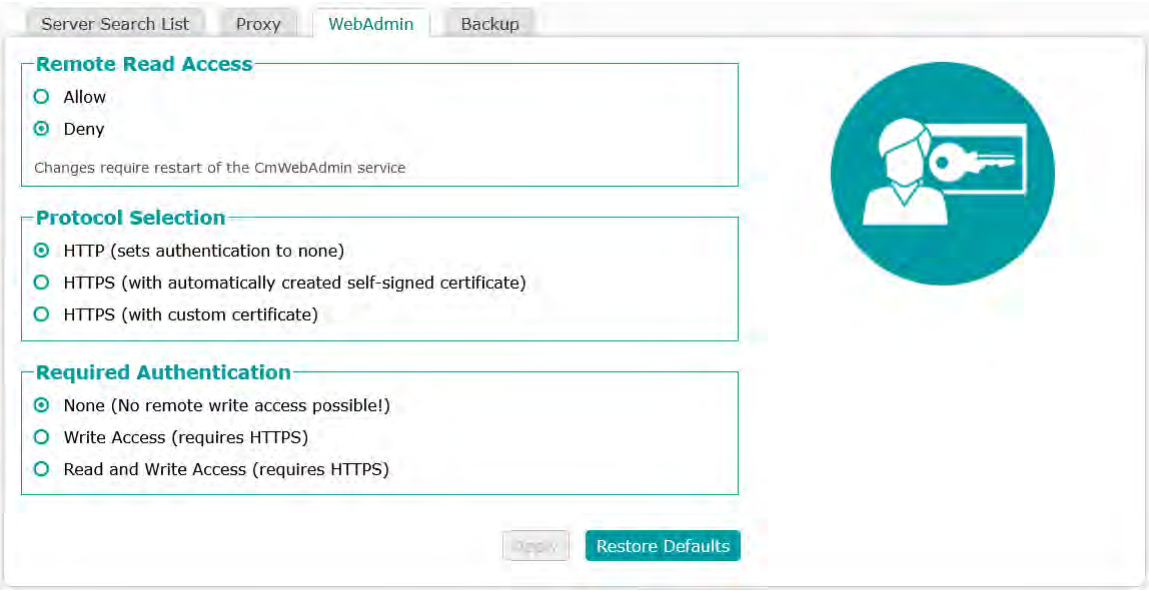


Figure 288: *CodeMeter WebAdmin* - "Configuration | WebAdmin"

> The pictograms R and W inform on **R**ead and **W**rite access. On mouse over tool tips display additional access mode details. A click on a pictogram opens - if required - the login on a separate page.
>
> Depending on the configured access modes a dropdown menu provides the following entries:
>
> - **Allow write access**
> - **revoke read access**
> - **Revoke read/write access**

Remote Read Access

By default, the remote read access is denied.

Element	Description
Allow	Activate this option to allow remote read access to *CodeMeter WebAdmin* (default).
Deny	Activate this option to deny remote read access to *CodeMeter WebAdmin*.

> A change of this during operation requires a restart of the *CmWebAdmin* service. Please note that you also have to set the firewall accordingly.
>
> Please proceed as follows to restart the *CmWebAdmin* service on Windows:
>
> 1. Enter "`Services`" in the Windows search field.
> 2. Find `CmWebAdmin` in open Services window.
> 3. Click right mouse button and select **Restart**.
>
> On macOS first stop the service and than start it using the following commandlines:
>
> ```
> sudo launchctl stop com.wibu.CodeMeter.WebAdmin
> sudo launchctl start com.wibu.CodeMeter.WebAdmin
> ```
>
> On Linux for a restart of the service use the following commandline:
>
> ```
> sudo service codemeter-webadmin restart
> ```

Protocol Selection

This group allows you selecting the protocol used for accesses. Optionally, you can select between HTTP and HTTPS.

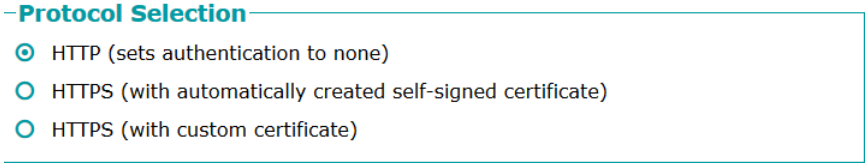
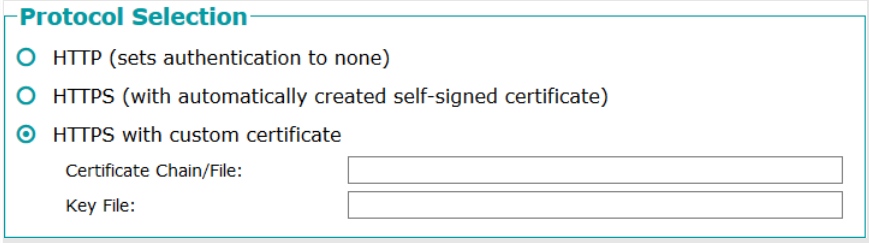


Figure 289: *CodeMeter WebAdmin* - "Configuration | Protocol Selection"

Element	Description
HTTP (sets authentication to none)	If you use HTTP (default), you do not require any authentication. The related "Required Authentication" radio button is automatically set to 'None'.
	If you use HTTPS, you can additionally decide whether using an automatically created self-signed certificate or a custom certificate you obtained from a separate certification authority (CA) provider.
HTTPS (with automatically created self-signed certificate)	 Automatically self-signed certificate files, i.e. the Certificate Chain/File and the Key File will be created, if not already existing, at the displays location,
HTTPS with custom certificate	 Here you enter path and file names of the certificate files you obtained from your certification authority (CA) provider.

HTTPS Protocol and Web browser

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the web browser issue warnings for not secure connections.

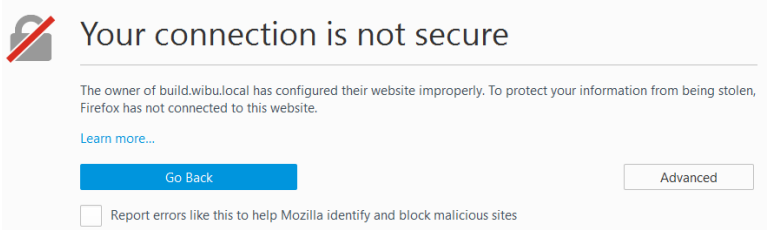
> If you use automatically generated, self-signed certificate files, you must nevertheless allow the access despite unsafe access information, and eventually define exceptions. The unsafe information refer only to the fact that certificates are self-signed and not issued by a certification authority (CA).
>
> If you have received certificate files from a certification authority (CA) provider, please follow the respective import instructions.

> If self-signed certificates expire and become invalid, a separate entry on the Dashboard 403 page will inform you displaying details on the certificate expiration time.

Please proceed for Firefox, Chrome, Internet Explorer, Microsoft Edge and Safari as follows.

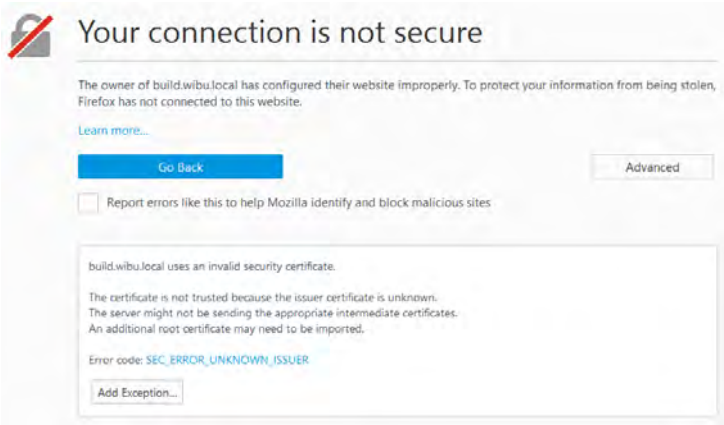
- Firefox

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the following page displays.

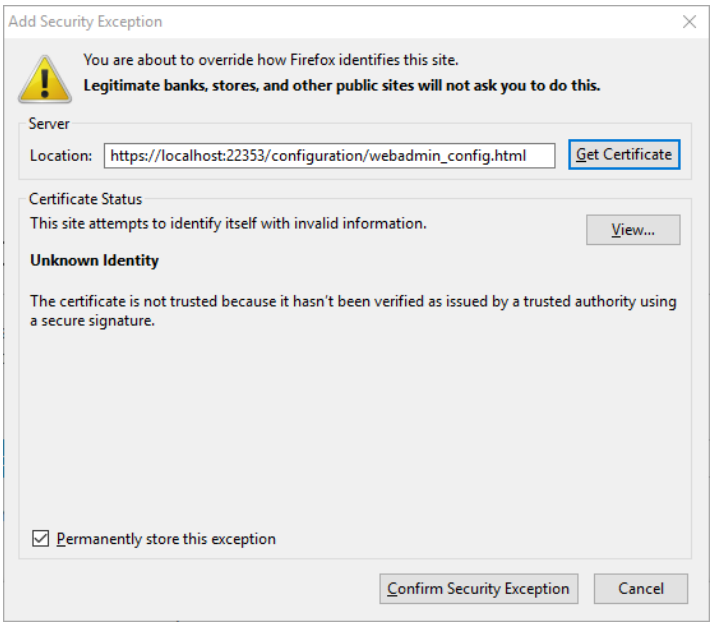


Then please proceed as follows:

1. Click the button "**Advanced**".
   A page displays asking you to import the certificate.



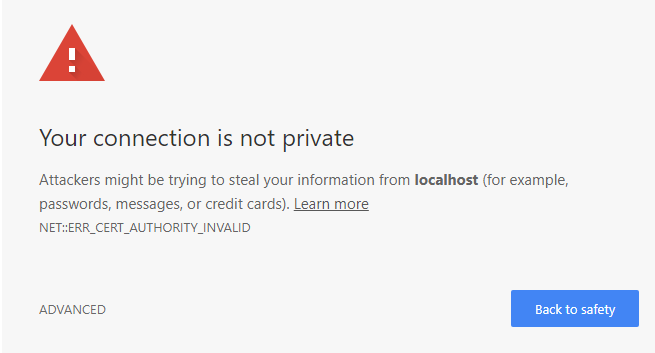
2. Click the button "**Add Exception**".
   A dialog to confirm the exceptions displays.



3. Click the button "**Confirm Security Exception**".
   CodeMeter WebAdmin opens.

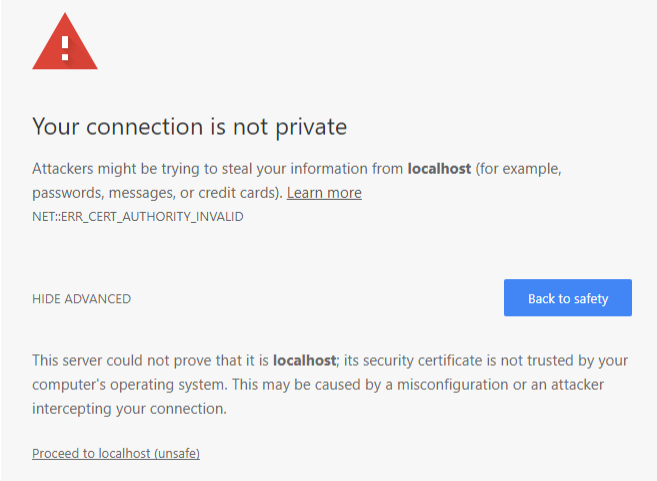
Chrome

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the following page displays.



Then please proceed as follows:

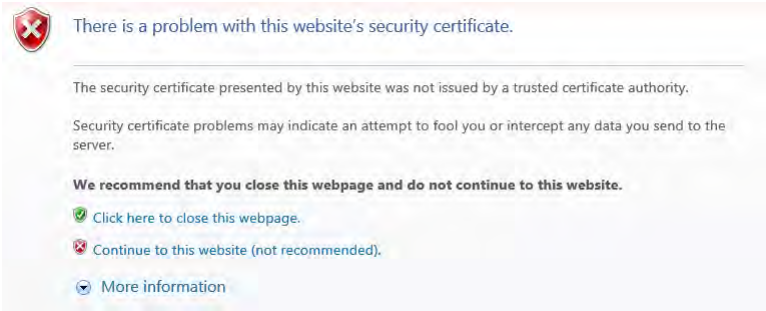
1. Click the button "**Advanced**".



2. Click the button "**Proceed to localhost (unsafe)**".
   The address line displays Not secure | https: .

- Internet Explorer

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the following page displays.

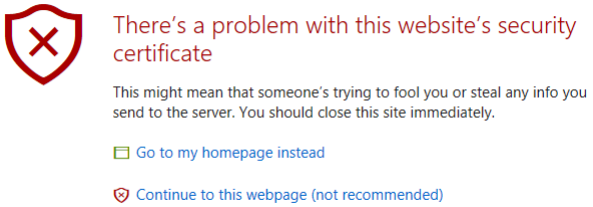


Then please proceed as follows:

1. Click the button "**Continue to this website (not recommended)**".
   The address line displays Certificate error.

- Microsoft Edge

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the following page displays.

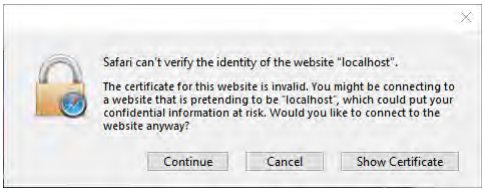


Then please proceed as follows:

1. Click the button "**Continue to this web page (not recommended)**".
   The address line displays Certificate error .

- Safari

After selecting the HTTPS protocol and opening *CodeMeter WebAdmin* the first time, the following page displays.



Then please proceed as follows:

1. Click the button "**Continue**".

Required Authentication

This group allows you specifying the required authentication credentials for accessing *CodeMeter WebAdmin* with Write and/or Read privileges.

Element	Description
None (No remote write access possible)	No authentication is required but also no remote write access to *CodeMeter WebAdmin* is possible.
Write Access (requires HTTPS)	Activate this option to allow authenticated write access to *CodeMeter WebAdmin*. Write Access Enter Password: Verify Password: Please complete the necessary data in the fields **Password** and **Verify Password** for Write Access, and click the button "**Apply**". Then write access will require a login on a separate login page. Please click the button "**Login for write permission**". Login for write permissions The login page displays. Authenticate for Write Access Enter Password: Login
Read and Write Access (requires HTTPS)	In order to organize read and write access even more fine-grained, you are able to use different credentials for different groups allowed for authenticated remote read and/or write access to *CodeMeter WebAdmin*. Read and Write Access Enter Read Password: Verify Read Password: Enter Write Password: Verify Write Password: Please complete the necessary data in the fields **Password** and **Verify Password** for Read and Write Access. Then read access will require a login on a separate login page. Authenticate for Read Access Enter Password: Login Write access will additionally require a login on a separate login page. Please click the button "**Login for write permission**". Login for write permissions The login page displays. Authenticate for Write Access Enter Password: Login

Click the "**Apply**" button to save all the changes you have made. By a previous click on the "**Restore Defaults**" button you save the default settings.

12.6.8.4 Backup

On the **"Configuration | Basic | Backup"** page you define settings for the location and intervals of *CmDongle* data backups.

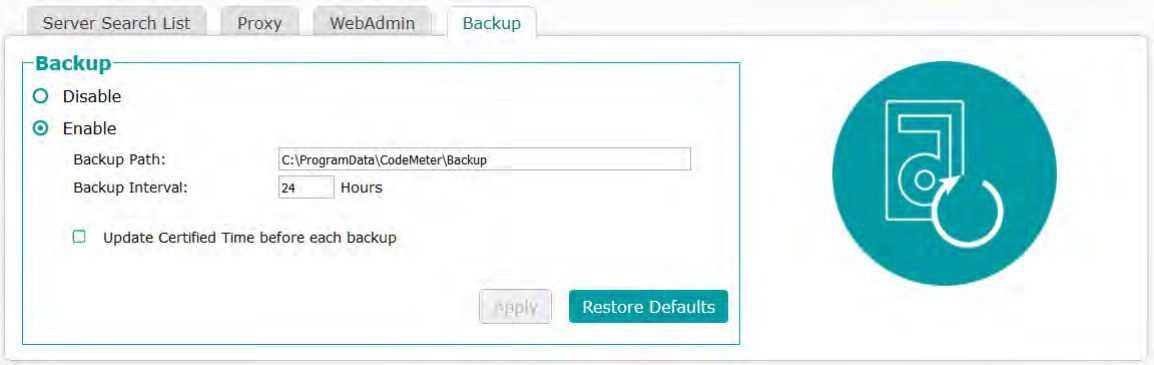


Figure 290: *CodeMeter WebAdmin* - "Configuration | Backup"

Element	Description
Backup Path	Specify in the **Backup Path** field the location where the backup file of the *CmDongle* is to be saved. The default location for backup files depends on the operating system in use.
Backup Interval	Specify in the **Backup Interval** field the recurring time period for automatic backups. By default, automatically a data backup is executed every 24 hours. However, you are also able to create a backup for the *CmDongle* at any time.
Certified Time	Activate this option when a **Certified Time** update has to take place before a backup is executed.

Click the "**Apply**" button to save the changes you have made.

By a previous click on the "**Restore Defaults**" button you save the default settings.

12.6.8.5 Server Access

On the **"Configuration | Server | Server Access"** page you set up *CodeMeter*® in a network and/or a wide area network (WAN).

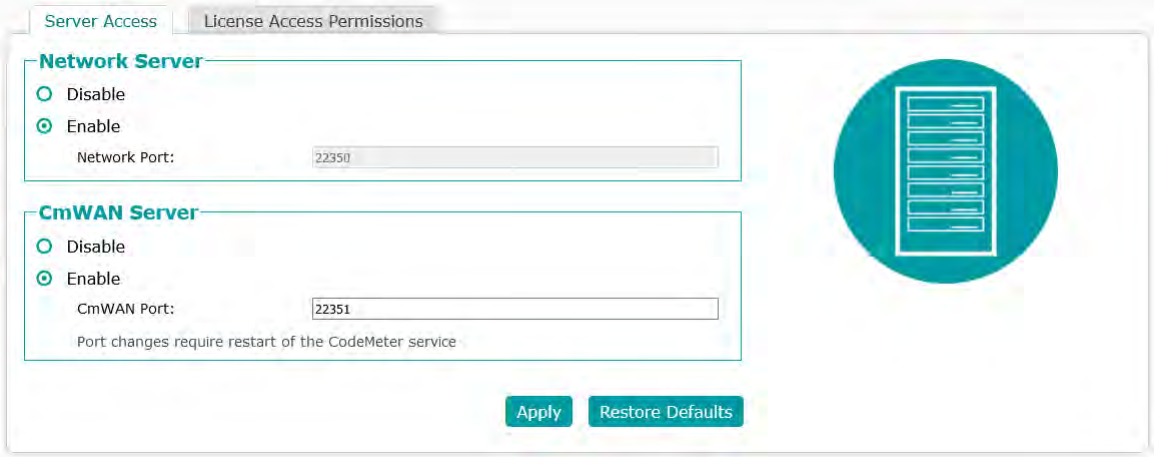


Figure 291: *CodeMeter WebAdmin* – "Configuration | Server | Server Access"

For activating the **Network Server** option, please proceed as follows:

1. Click the **"Enable"** radio button to use the PC as *CodeMeter* network server.
   Then this PC provides its *CodeMeter* licenses on the network using the service *CodeMeter License Server*.
2. Specify a **Network Port**. By default, the port 22350 is used for the *CodeMeter* communication.
   This network port is registered at IANA (Internet Assigned Numbers Authority) and uniquely assigned for the *CodeMeter* communication.

   You are able to customize this port value. However, make sure that all *CodeMeter License Server* use this port when a *CodeMeter* protected application is to be used on the network.

3. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings.

For activating the **CmWAN Server** option, please proceed as follows:

1. Click the **"Enable"**radio button to use the PC in a wide area network (WAN) and allow license accesses.

   Please note, that a WAN connection has to be provided by the Software Vendor.

2. Specify a **CmWAN Port**. By default, the port 22351 is used for the *CodeMeter* communication via WAN.

   You are able to customize this value. In this case, make sure that:
   - all *CodeMeter License Servers* use this port, if *CodeMeter* protected applications access licenses via WAN.
   - the configured reverse proxy has the same port setting.
     Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.

3. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings.

   Please note that changes made to the port settings require a restart[384] of *CodeMeter License Server*.

12.6.8.6 License Access Permissions

On the **"Configuration | Server | License Access Permissions"** page you define settings managing all client accesses to *CodeMeter License Server*.

Please note, that you previously must have configured *CodeMeter License Server* (`CodeMeter.exe`) as a network or a CmWAN server ("Server Access[425] ").
At the same time, also the license access permissions for `localhost` are managed.

On access configuration you can select among a basic and an advanced mode.

The basic[426] mode allows adding client computer and IP addresses for accessing *CodeMeter License Server*.

The advanced[428] mode allows, for example, specifying global and specific access rules for accessing licenses and reserving license access for single staff member or complete Active Directory groups.

Please select the desired License Access Permissions mode.

Basic Mode

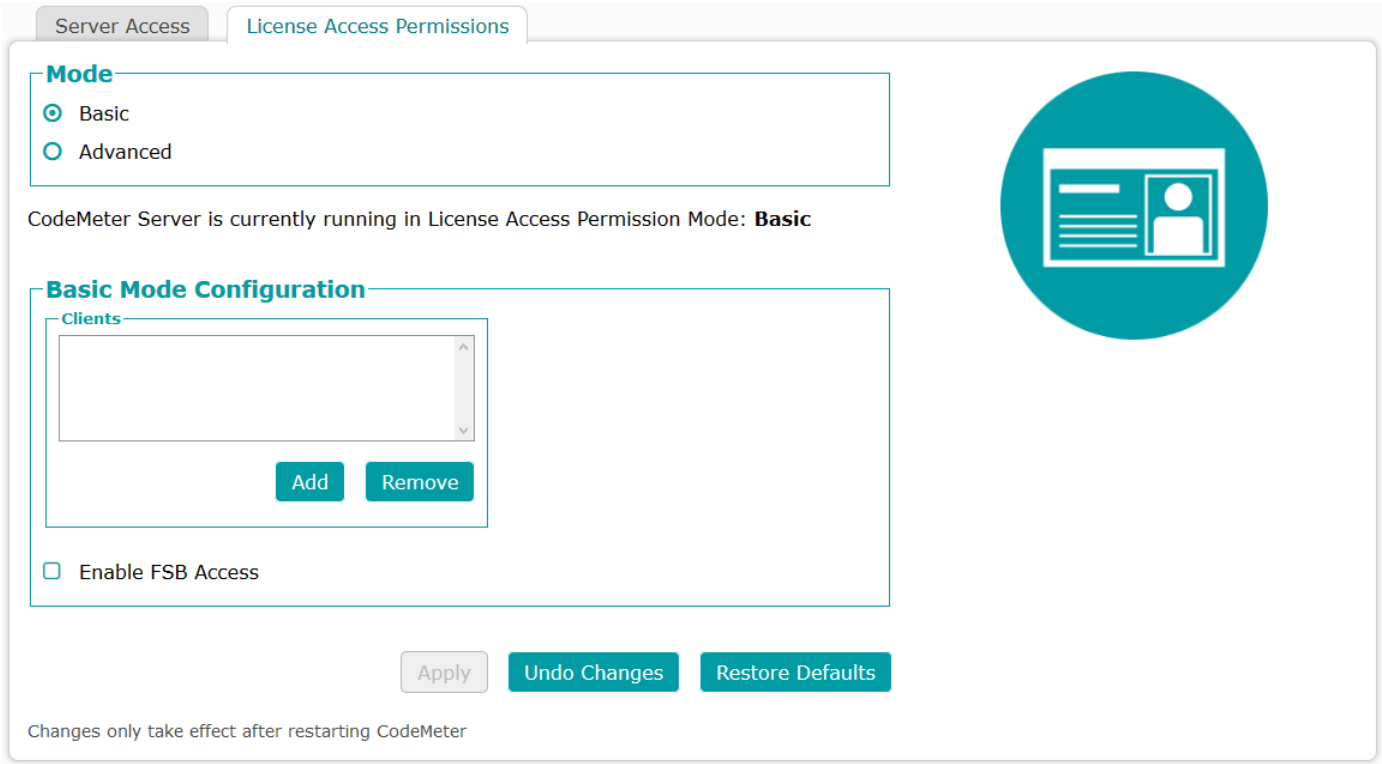
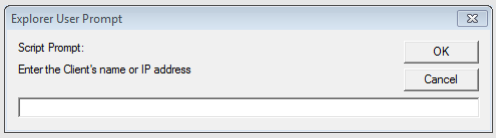


Figure 292: *CodeMeter WebAdmin* - "Configuration | Access Control"

Element	Description
Clients	Shows a list of all client PCs which have the privilege to use *CodeMeter License Server*, i.e. to allocate a license.

Element	Description
	When this list is empty, each *CodeMeter* client on the network is able to use *CodeMeter License Server*. This is the default setting. To add a new client to the client list, please proceed as follows: **1.** Click the **"Add"** button. A prompt dialog displays.  **2.** Specify the PC name or the IP address of the client in the dialog. **3.** Click the **"OK"** button. The PC is now added to the client list. To remove a client from the list, please proceed as follows: **1.** Click the **"Remove"** button. The PC is now removed from the client list
Enable FSB Access	If you own a *CodeMeter Firm Security Box* (FSB), this option activates the sharing of the FSB on the network. Then the FSB is able to be used by several users, for example, to program *CmContainer* or automatically protect applications. This option makes sense only for *CodeMeter* licensee with an individual *CodeMeter Firm Code.* Click the **"Apply"** button to save the changes you have made. By a previous click on the **"Default"** button you save the default settings. Then the client list is empty, and the FSB is not available on the network. When you define access settings, in some cases, this requires the restart of the *CodeMeter* service. However, you do not have to eject or deactivate the *CmContainer*. After you specified the settings you are able to stop and then restart the *CodeMeter* service in *CodeMeter Control Center* 388 . For non-Windows operating systems see here 384 .

Additional access control of client list via whitelist and Blacklist

Alternatively, you also have the option to create a white or blacklist for the access of clients.

Please note that on specifying subnet masks only input dividable by 8 is accepted.

This so-called profiling you conduct for different operating systems at the following locations:

Operating System	Profile Creation
Windows	Registry entry in `HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini.`

The generation of the profile for *CodeMeter License Server* comprises the following versions (`CodeMeter.exe`, `CodeMeterMacX`, `CodeMeterLin`, `CodeMeterSun`),

When you edit the `*.ini` files in the case of macOS and Linux, you must stop the service *CodeMeter License Server* before. Otherwise, changes you have been made do not apply.

Parameter	Description
Client<index>=<Subnetz>[,<serial>[,FC[,PC]]]] (Whitelist)	**Whitelist**: These parameters hold the IP addresses of client PCs on the network which have the privilege to access the local *CodeMeter License Server*. When the IP address of a client is not on this list, the access is denied. If no whitelist exists, no other restrictions apply. The specification of subnets is possible. The syntax is as follows: `Client<index>=<Subnetz>[,<serial>[,FC[,PC]]]` The serial number has to follow the pattern `MaskByte-Serial Number` (e.g. `1-1179681`). Example: Client1=192.168.0.0/24,1-123456,10,13 this addresses all computer ranging from 192.168.0.0 to192.168.0.255 (Class C). Usually are also /8 (Class A) and /16 (Class B). The serial number, FC, and PC are optional. This whitelist corresponds to the client list in *CodeMeter WebAdmin*.

Parameter	Description
Blacklist<index>=<Subnetz> [,<serial>[,FC[,PC]]] [SZ, optional]	**Blacklist**: These parameters hold the IP addresses of client PCs on the network which have no privilege to access the local *CodeMeter License Server*. When an IP address of a client is on this list, the access is denied. If no blacklist exists, no other restrictions apply. The syntax is as follows: `Blacklist<index>=<Subnetz>[,<serial>[,FC[,PC]]]` The serial number has to follow the pattern `MaskByte-Serial Number` (e.g. `1-1179681`). Example: Blacklist1=192.168.0.0/24,1-123456,10,13 this addresses all computer ranging from 192.168.0.0 to192.168.0.255 (Class C). Usually are also /8 (Class A) and /16 (Class B). The serial number, FC, and PC are optional.

Advanced Mode

The advanced access control mode allows the controlling of license access using access rules. The license access by single staff members but also of complete Active Directory groups can be organized. The detection of single staff members (user name) and groups happens automatically without any integration efforts.

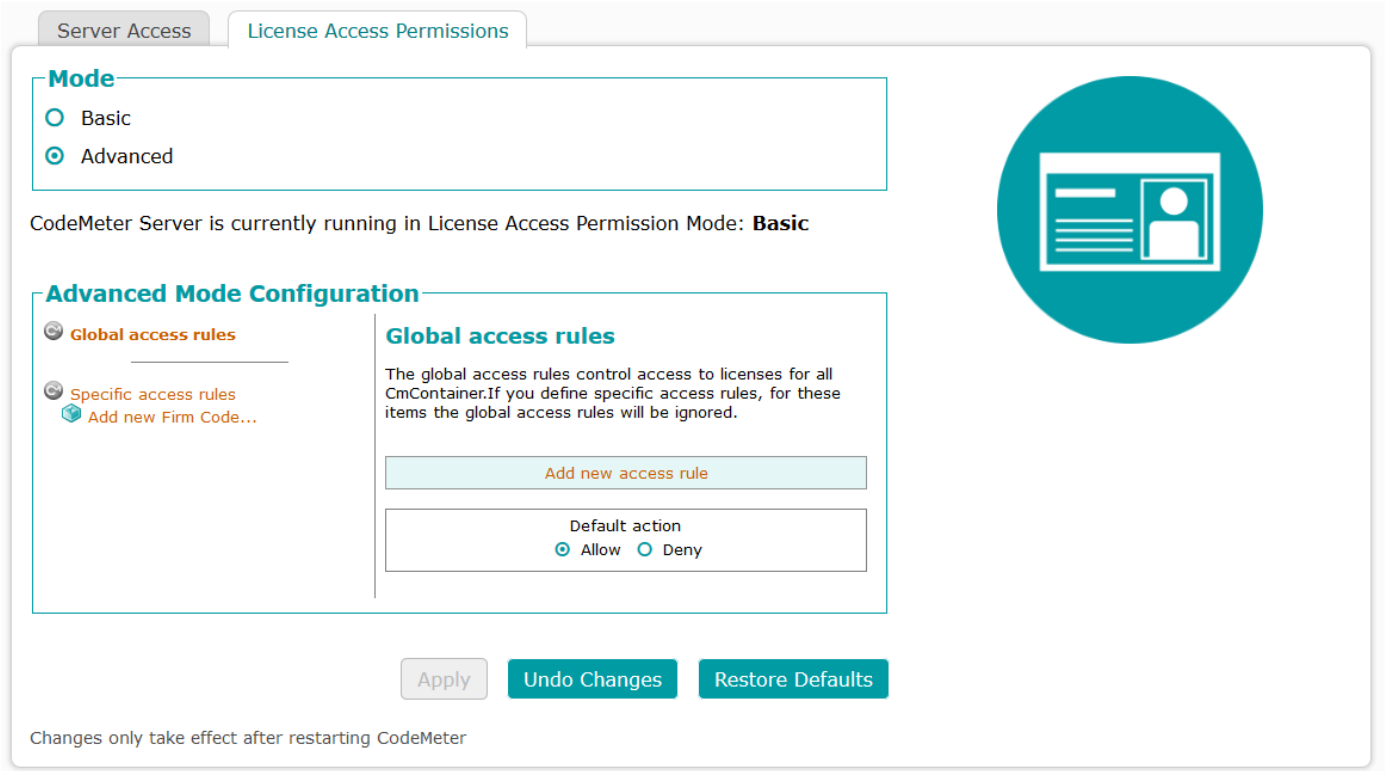


Figure 293:: *CodeMeter WebAdmin* - "Configuration | Access Control" - Advanced Access Control Mode

The advanced access control mode allows the controlling of license access using access rules. The license access by single staff members but also of complete Active Directory groups can be organized. The detection of single staff members (user name) and groups happens automatically without any integration efforts.

Two types of access rules exist:

- global access rules 429
- specific access rules 430

The global access rules control the license access to **all** *CmContainer*. If specific access rules have been defined, then these are exempted from the global access rules.

The specific access rules control license access to separately specified license entries for ISV or applications (*Firm Code*, *Product Code*). Then the specific access rules are valid and for matching license entries the global access rules are ignored.
Coupled with a limitation of user here also the number of available licenses can be set for each rule, and licenses can be reserved. This can set specific limits e.g. for some departments, while keeping one license reserved for exclusive use.

In some cases, different *Product Items* with the same *Product Code* are present, for instance, if the same software was bought with different license options. In such instances, the *Product Items* can be identified using associated *Product Items Text*, and rules can be defined for each *Product Item*.

The following conditions hold true for access rules:

- access rules are created, edited and deleted in a separate area or dialogs.

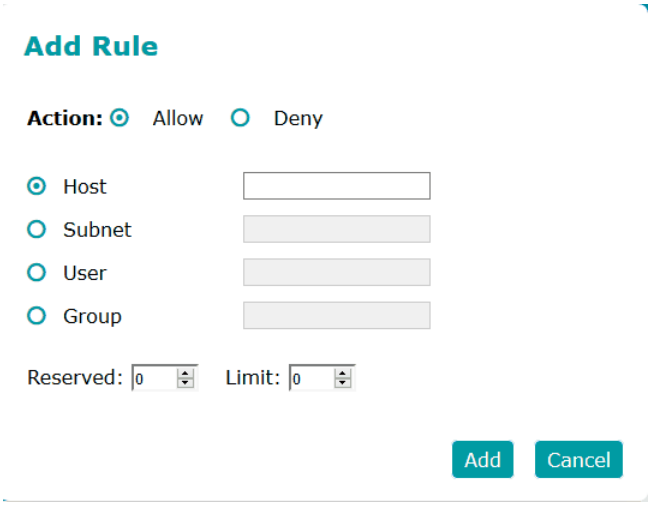
- access rules may cover several rules. Rules are processed top-down, which means that the order of the rules is decisive for the result.
- access rules conclude with an area defining the default setting for all license accesses which are not covered by rules.
- If specific access rules are defined, the most specified available access rules apply. If no rules are configured for the *Product Code*, the rules for the *Firm Code* apply. If no *Firm Code* rules exist, the global rules apply.

Please note that reading complete Active Directory (AD) groups depending on the complexity of their structures may take some time.

Creating the global access rules

In order to create the global access rules to control license access globally for all *CmContainer*, please proceed as follows:

1. Select the **"Global access rules"** item in the left tree view.
2. Click the **"Add new access rule"** button.
   A dialog for defining a new rule displays.



3. Click the **"Allow"** or **"Deny"** radio button in the area **Action** to decide, whether the following license access by client is to be allowed or denied.

   A client access can be defined by one of the following parameter: **Host** name, **Subnet** address **User** or **Group** name.
4. Specify the desired parameter in the respective field. If an active directory (AD) is integrated, the list of fields **User** and **Group** is auto-completed.
   Setting a profiling parameter allows to defne the update interval 381 to display this list.
5. Click the **"Add"** button to add the new rule.
   A click on the **"Cancel"** button cancels the process.

   The new rule displays in the right rule view.

   If you defined several rules, you may change the rule sequence by using the arrow symbols ↑ ↓. Rules are processed top-down, which means that the order of the rules is decisive for the result.

   Using the **"Edit"** or **"Delete"** link allows you to modify a completely delete a rule.
6. Define the default setting for all license accesses which are not covered by rules.
   You have the option to set the **Default action** to allow or deny license access.

   Click the **"Allow"** or **"Deny"** button.
7. Click the **"Apply"** button in the lower part to save the changes made to the global access rules. Using the **"Undo Changes"** button reverts the global access rules prior to the modification, and the **"Restore Defaults"** button applies the default settings.
   If you apply the changes made, please restart 384 *CodeMeter License Server*.

   e.g. The figure below shows an example of global access rules. It allows the global license access additionally by a guest user. All other licenses accesses are also allowed, if no specific access rules specify otherwise.

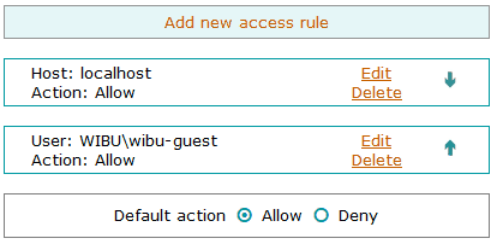


Figure 294:: *CodeMeter WebAdmin* - "Configuration | Access Control" - advanced Mode- global access rules

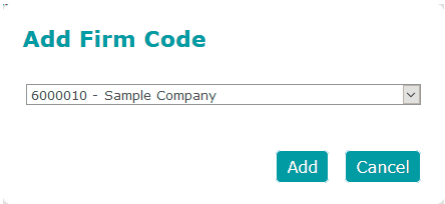
Creating specific access rules

In addition to defining the global access rules, you have also the option to control the license access to specific separate license entries. Here you define specific access rules for separate defined *Firm Codes* and/or *Product Codes*.

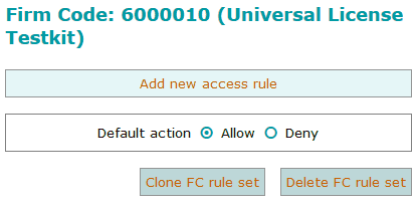
Firm Code-specific access rules

In order to create specific access rules to control license access to separate *Firm Codes*, please proceed as follows:

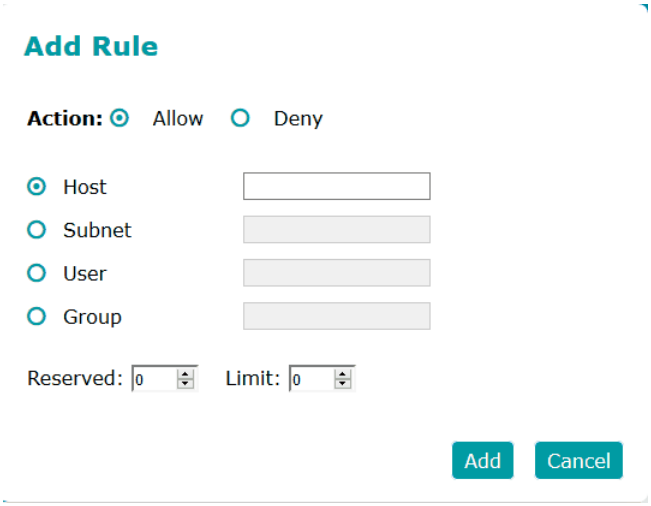
1. Select the **"Specific access rules"** item in the left tree-view.
2. Click the **"Add Firm Code"** button.
   The dialog for selecting a *Firm Codes* displays.



3. Select the *Firm Code* and click the **"Add"** button.
   New specific access rules valid for this *Firm Code* display in the right rule view.



4. Click the **"Add rule"** button.
   A dialog for defining a new rule displays.



5. Click the **"Allow"** or **"Deny"** radio button in the area **Action** to decide whether the following license access by client is to be allowed or denied.

   A client access can be defined by one of the following parameter: **Host** name, **Subnet** address **User** or **Group** name.
6. Specify the desired parameter in the respective field. If an active directory is integrated, the list of fields **User** and **Group** is auto-completed.
7. Click the **"Add"** button to add the new rule.
   A click on the **"Cancel"** button cancels the process.

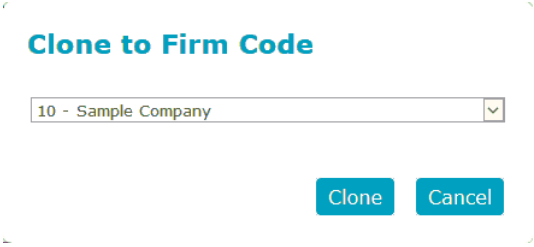
   The new rule displays in the right rule view.

   If you defined several rules, you may change the rule sequence by using the arrow symbols ↑ ↓. Rules are processed top-down, which means that the order of the rules is decisive for the result.

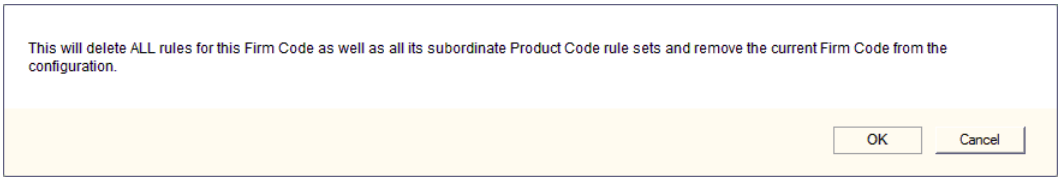
   Using the **"Edit"** or **"Delete"** link allows you to modify a completely delete a rule.
   If you delete *Firm Code*-specific access rules using the **"Delete rule list"** button, then also all *Product Code*-specific access rules - if existing - are deleted.
8. Define the default setting for all license accesses which are not covered by rules.
   You have the option to set the **Default action** to allow or deny license access.

   Click the **"Allow"** or **"Deny"** button.
9. Optionally, you are able to clone an existing rule set for a *Firm Code* (FC) 1:1 to another *Firm Code*.

Click the button **"Clone FC rule set"** to open a dialog which allows to select a target *Firm Code*. Clicking the button **"Clone"** starts the process.



Clicking the button **"Delete FC rule set"** deletes **all** rules.



Please note that not only the *Firm Code* rule set is removed but ALL rules also for any subordinate *Product Codes*.

10. Click the **"Apply"** button in the lower part to save the changes made to the global access rules. Using the **"Undo Changes"** button reverts the global access rules prior to the modification, and the **"Restore Default"** button applies the default settings.
If you apply the changes made, please restart 384 *CodeMeter License Server*.

e.g. The figure below shows an example *Firm Code*-specific access rules. It allows the license access to the complete *Firm Code* 10 by a guest user and the complete support department. All other licenses accesses are also allowed, if no specific access rules specify otherwise.

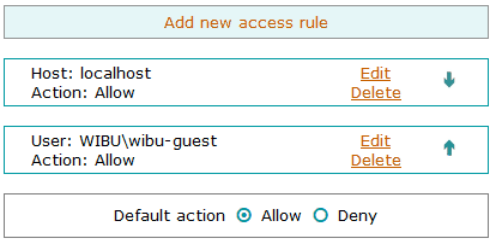


Figure 295:: *CodeMeter WebAdmin* - "Configuration | Access Control" - Advanced Mode - Specific access rules - *Firm Code*

Product Code-specific access rules

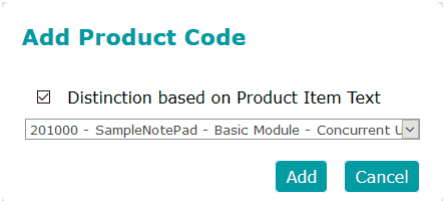
Specific access rules which refer to *Product Codes* also offer the option to reserve license accesses to defined clients. This, for example, allows to organize license access for separate departments while at the same time reserving exclusive license access for the heads of departments.

In order to create specific access rules to control license access to separate *Product Codes*, please proceed as follows:

Creating a *Product Code*-specific access rules requires a previously created *Firm Code*-specific access rules.

1. Select the **"Specific access rules"** item in the left tree-view.

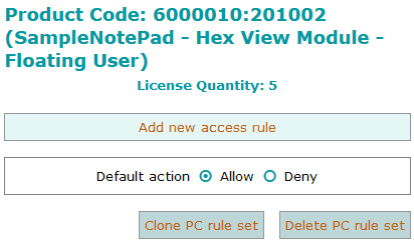
2. Click the **"Add Product Code"** button.
The dialog for selecting a *Product Code* displays.



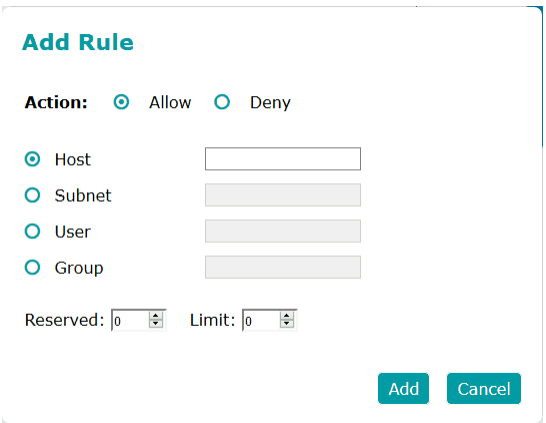
Checkbox **"Distinction based on Product Item Text"** .

3. Select the *Product Codes* and click the **"Add"** button.
A new specific access rules valid for this *Product Code* displays in the right rule view.

At the same time, the entry displays information on the **License Quantity**, i.e. the number of concurrent licenses on a network. This number is not to be exceeded, if later defining limits to the number of accesses.



4. Click the **"Add new access rule"** button.
   A dialog for defining a new rule displays.



5. Click the **"Allow"** or **"Deny"** radio button in the area **Action** to decide whether the following license access by client is to be allowed or denied.

   A client access can be defined by one of the following parameter: **Computer** name, **Subnet** address **User** or **Group** name.
6. Specify the desired parameter in the respective field. If an active directory is integrated, the list of fields **User** and **Group** is auto-completed.
7. Specify the number of license accesses which can be optionally reserved for a defined client in the field **Reserved**. The field **Limit** states the allowed maximum of allocated license accesses by this client. The setting for a reserved license access always available for the client is: Reserved: 1; Limit: 1.

   Please note that in the case of further specific access rules reservations and limits are added. The value of the License Quantity must not be exceeded.

   If a rule applies but according to the limit set no licenses can be allocated by this rule, then it is assumed that the rule does not apply and the next rule is applied.
8. Click the **"Add"** button to add the new rule.
   A click on the **"Cancel"** button cancels the process.

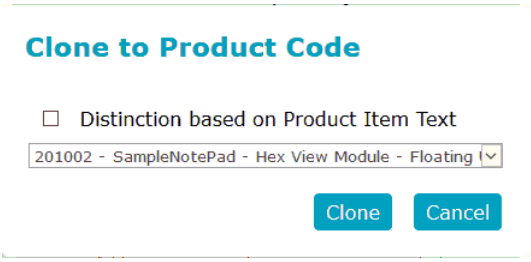
   The new rule displays in the right rule view.

   If you defined several rules, you may change the rule sequence by using the arrow symbols ↑ ↓. Rules are processed top-down, which means that the order of the rules is decisive for the result.

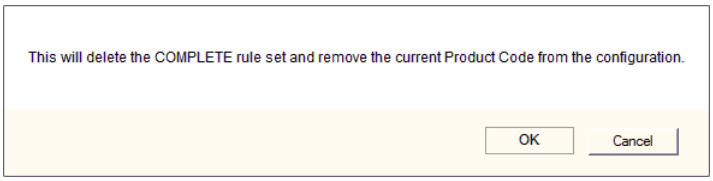
   Using the **"Edit"** or **"Delete"** link allows you to modify a completely delete a rule.
   In order to delete a complete *Product Code*-specific access rules use the **"Delete rule list"** button.
9. Define the default setting for all license accesses which are not covered by rules.
   You have the option to set the **Default action** to allow or deny license access.

   Click the **"Allow"** or **"Deny"** button.
10. Optionally, you are able to clone an existing rule set for a *Product Code* (PC) 1:1 to another *Product Code*.

    Click the button **"Clone PC rule set"** to open a dialog which allows to select a target *Product Code*. Clicking the button **"Clone"** starts the process.



Clicking the button **"Delete PC rule set"** deletes the **complete** rule set for this *Product Code*.



11. Click the **"Apply"** button in the lower part to save the changes made to the global access rules. Using the **"Undo Changes"** button reverts the global access rules prior to the modification, and the **"Restore Defaults"** button applies the default settings.
If you apply the changes made, please restart[384] *CodeMeter License Server*.

e.g. The figure below shows an example of a specific access rules with exclusive access rights (Reserved: 1; Limit: 1) to the *Product Code* 201000 of *Firm Code* 10 for a guest user, the complete support department and a Supervisor. 2 license accesses of a total of 5 license accesses (license quantity) remains available and the default license access is defined as allowed.

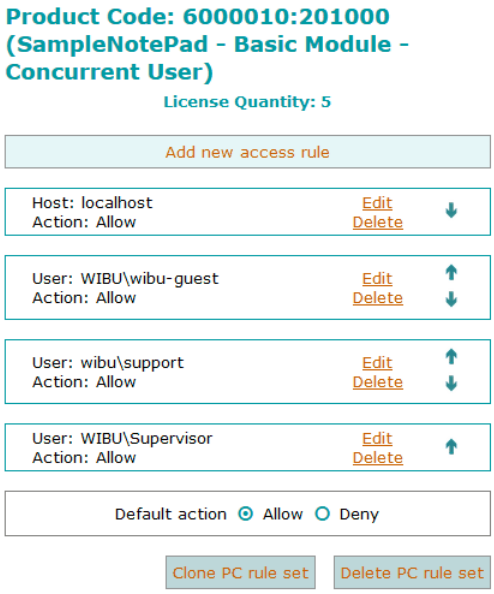


Figure 296:: *CodeMeter WebAdmin* - "Configuration | Access Control" - Advanced Mode - Specific access rules - *Product Code*

12.6.8.7 Prepared License Borrowing

On the page **"Configuration | Server | Prepared License Borrowing"** you are able, if you require, to modify entry-specific settings of borrowed licenses, in order to change the number of borrowed licenses or the borrowing period to a value other than programmed.

These settings display only, if activated at **"Configuration | Advanced Configuration | Extra"** in the group **Additional WebAdmin Configuration**[436].

Please note, that Prepared License Borrowing supports only *Firm Codes* smaller than `6.000.000`.

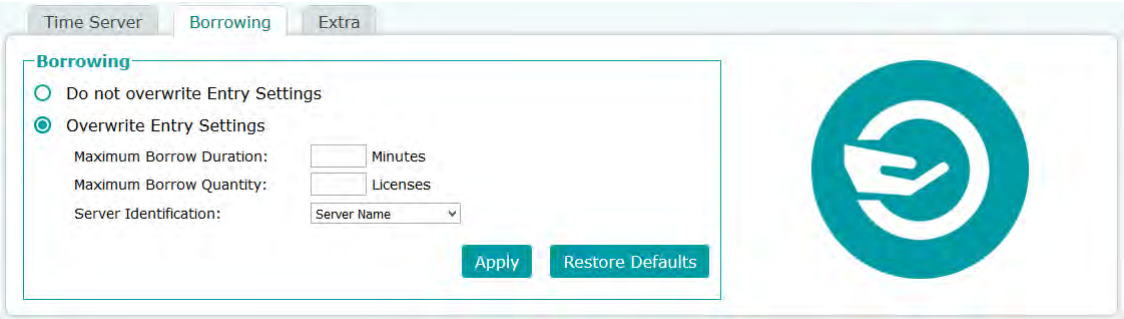


Figure 297: *CodeMeter WebAdmin* – "Configuration | Server | Prepared License Borrowing"

For the individual setting of License Borrowing paramete, please proceed as follows:

1. Activate the option **"Overwrite Entry Settings"** in order to allow modifying license options of the borrowed license.
2. Enter in the field **"Maximum Borrow Duration"** the maximum borrowing period in minutes, the license is to be borrowable.
3. Enter in the field **"Maximum Borrow Quantity"** the maximum number of borrowed license to be borrowable.
4. Select in field **"Server Identification"** how the server is identified: either by Server Name or IP address.
5. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings.

12.6.8.8 Time Server

On the **"Configuration | Advanced | Time Server"** page you define settings for the *CodeMeter* Time Server.

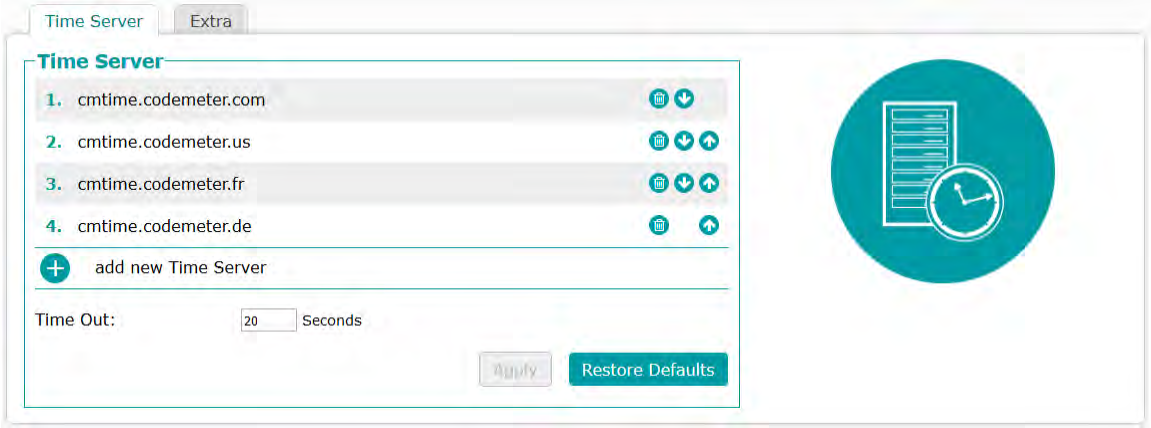


Figure 298: *CodeMeter WebAdmin* - "Configuration | Time Server"

Element	Description
Time Server	Shows a list of Wibu-Systems *CodeMeter* Time Server allowing for an update of the *Certified Time*. Time Server are specified either as Internet address or IP address. You edit the Time Server list by using the **"add"** or **"remove"** buttons. You change the order of the list by using the **"up"** and **"down"** buttons.
Time Out	Defines the maximum response period for the *CodeMeter* Time Server. The default value is 20 seconds. Click the **"Apply"** button to save the changes you have made. By a previous click on the **"Restore Defaults"** button you save the default settings.

Please note that only Wibu-Systems *CodeMeter* time servers may be specified here.

You cannot specify your own NTP (Network Time Protocol) time servers here, as this time synchronization does not guarantee a Certified Time, which plays an important role, for example, when retrieving and subsequently validating time-based licenses.

12.6.8.9 Extra

On the **"Configuration | Advanced | Extra"** page you configure some additional advanced settings.

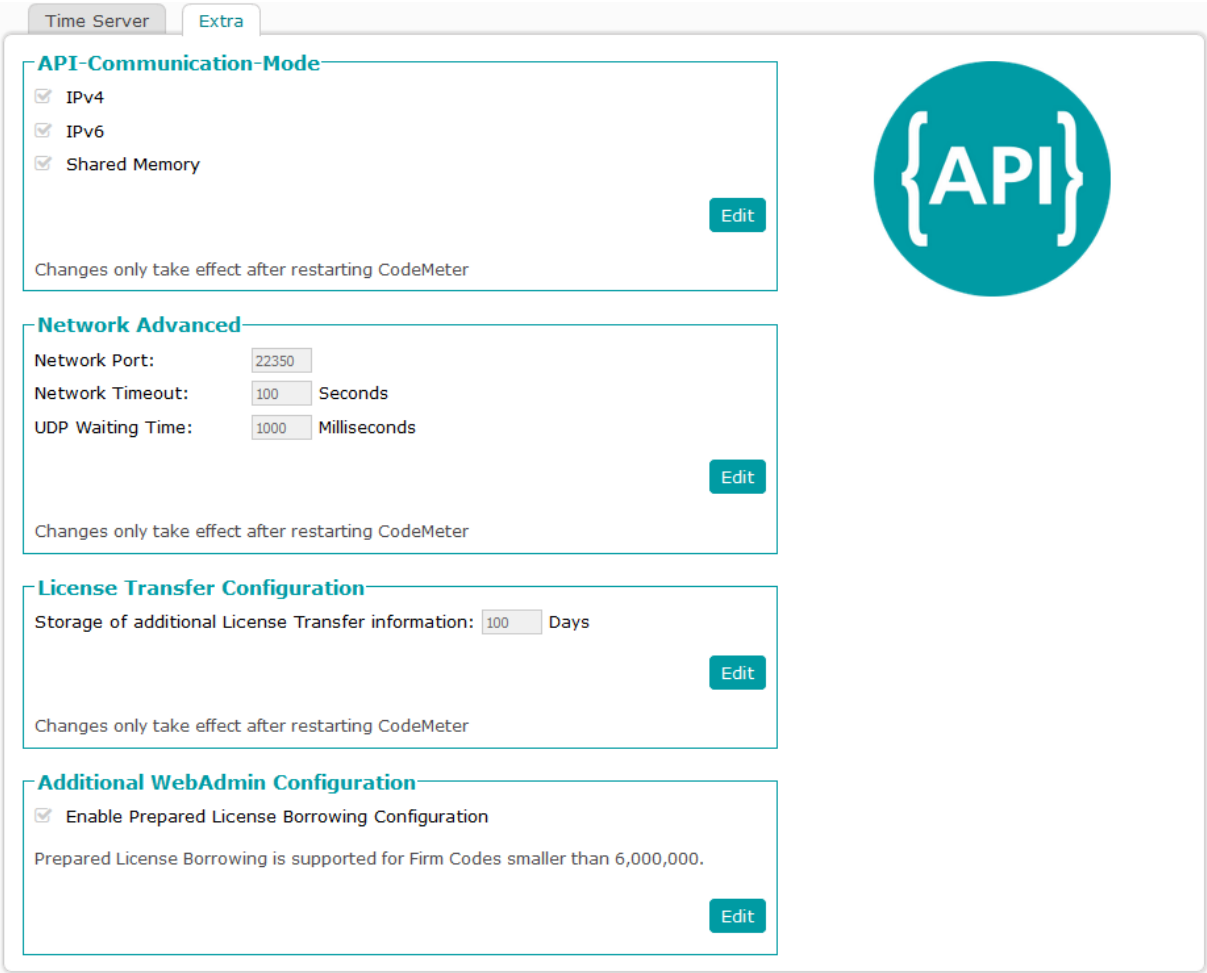


Figure 299: *CodeMeter WebAdmin* – "Configuration | Advanced | Extra"

API Communication Mode

Here you define which communication mode *CodeMeter License Server* uses. Please proceed as follows:

1. Click the **Edit** button.
2. Check the box to select the mode.

   The following parameter are available:

CodeMeter-Version	Properties
smaller than 4.40	'1' TCP/IP (Default) '2' Shared Memory
starting with 4.40	'1' Platform-specific (Default) Platform-specific defaults: • Windows: IPv6, IPv4; Shared Memory • Linux/macOS:IPv6, IPv4 • WinCE: IPv4, Shared Memory '2' Shared Memory '4' IPv4 '8' IPv6 Single modi may be combined.

Wibu-Systems recommends to use the relevant default settings, if no justified reasons suggest otherwise.

Alternatively, by editing registry or server entries you are also able to define which communication mode *CodeMeter License Server* uses. The following table shows you where for which operating system you find the profiling to set the communication mode. The parameter **ApiCommunicationMode**. is available for setting the mode.

Operating system	Registry / Server Entry
Windows	`HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini`

3. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings. Clicking the button **"Cancel"** cancels the configuration.

> Make sure you restart *CodeMeter* after changing the settings.

Network Advanced

Here you define some advanced network settings. Please proceed as follows:

1. Click the **Edit** button.
2. Enter the desired values in the respective fields.

   The following parameter are available:

> Please make sure you activated *CodeMeter License Server* for running as network server 425 .

Element	Description
Network Port	Define a network port other than the default port, if desired.
Network Timeout	Defines the maximum response period for the *CodeMeter* License Server if running on a network. By default, this value is 100 seconds.
UDP Waiting Time	Specify the **UDP Waiting Time** in order to define the period in which a UDP request for existing *CodeMeter License Server* on the network has to reply. By default, this value is 1000 milliseconds. Changing this time allows to customize the performance of the service. However, when no urgent need exists, you should keep that default.

3. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings. Clicking the button **"Cancel"** cancels the configuration.

> Make sure you restart *CodeMeter* after changing the settings.

License Transfer Configuration

Here you define the period of how additional license transfer activities are stored allowing you to obtain information on the PC (host) to which the last transfer activity referred to.

Please refer as follows:

1. Click the **Edit** button.
2. Enter the desired value in the respective field.

   The following parameter are available:

Element	Description
Storage of additional License Transfer information	Specifies the number of days license transfer information is saved. The default setting is 100 days.

4. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings. Clicking the button **"Cancel"** cancels the configuration.

> Make sure you restart *CodeMeter* after changing the settings.

Additional WebAdmin Configuration

If you require, you are able to allow and activate modifying entry-specific settings of borrowed licenses.

> Please note, that Prepared License Borrowing supports only *Firm Codes* smaller than `6.000.000`.

Please proceed as follows:

1. Click the **"Edit"** button

   The checkbox **Enable Prepared License Borrowing Configuration** becomes editable.
2. Check the box **Enable Prepared License Borrowing Configuration** to trigger displaying the page **"Configuration | Server | Prepared License Borrowing** 433 **"**.

5. Click the **"Apply"** button to save the changes or **"Restore Defaults"** restoring the default settings. Clicking the button **"Cancel"** cancels the configuration.

12.6.9 Info

The **"Info"** page displays an overview of products and important Wibu-Systems addresses.

Figure 300: *CodeMeter WebAdmin* – Info

12.6.1(License Transfer

> The License Transfer is supported for only, if using ***Universal Firm Codes*** with a number range bigger than 6.000.000.
>
> For *CodeMeter* Version 6.10 the following is valid:
>
> - License Transfer is supported only for the ***CmContainer Type*** *CmActLicense*.
> - Of the Transfer Types only the types ***Licenses*** (moving 'n' from 'm' licenses) and ***Borrow local license***.

CodeMeter WebAdmin on various pages displays License Transfer Options and a History. Following tables list potentials items.

For a brief example of a license transfer of type "Licenses" ('n' from 'm' licenses) see here 439 and for the type "Borrow local license" here 443.

License Transfer Options

In the case of a license transfer activity, this table displays the actual values of the license transfer option.

Option	Description
License received from	Serial number of the *CmContainers* from which a license has been received.
Pulling allowed	<table><tr><th>Value</th><th>Description</th></tr><tr><td>yes</td><td>A license can be actively requested by a receiving *CmContainer* from a sending *CmContainer*. Is valid only for all not time-limited license transfer types (move complete, move units, move licenses).</td></tr><tr><td>no</td><td>A license cannot be actively requested by a receiving *CmContainer* from a sending *CmContainer*.</td></tr></table>
Returning allowed	<table><tr><th>Value</th><th>Description</th></tr><tr><td>yes</td><td>A previously transferred license can be returned from the receiving *CmContainer* to the sending *CmContainer*.</td></tr><tr><td>no</td><td>A previously transferred license cannot be returned from the receiving *CmContainer* to the sending *CmContainer*.</td></tr></table>
Firm Item at target required	<table><tr><th>Value</th><th>Description</th></tr><tr><td>yes</td><td>On the target *CmContainer* a *Firm Item* must exist before the transfer takes place (*CmActLicense*).</td></tr></table>

Option	Description
	Value: no Description: On the target *CmContainer* a *Firm Item* must not exist before the transfer takes place (*CmActLicense*).
Transfer Type	Display of Transfer Type Value: Licenses Description: An existing *Product Item* with *License Quantity* is duplicated in two *Product Items* while the *License Quantity* is split over the two *CmContainer*. Value: Borrow Local License Description: A license is borrowed for local use only (without connection to a license server) for a definable period from one *CmContainer* to another. After the period has expired, the licenses automatically reallocates to the server's license pool. A locally borrowed license can not be further transferred
Transfer Depth	This option specifies the number of license transfer levels involved
Borrow Expiration Time	Date at which the borrowing expires.
Borrow Source	Serial number of the Source-*CmContainers* from which a license has been borrowed.
Licensor Transfer Key	Key to create the required certificates.

License Transfer History

In the case of a license transfer activity, this table displays the license transfer history.

Item	Description
#	Sequential number of the license transfer history item.
Status	Status of the transfer --- created, in transit of the transfer mode (push, return, pull), completed after obtaining receipt Value: 0 Description: 0 Value: Transferred Description: The transfer has been performed. Value: In Transit (return) Description: The return transfer is currently in transit. Value: In Transit (pull) Description: The pull transfer is currently in transit. Value: Transferred & Receipted Description: The transfer has been completed after receipt was obtained.
Type	Display of Transfer Type Value: Licenses Description: An existing *Product Item* with *License Quantity* is duplicated in two *Product Items* while the *License Quantity* is split over the two *CmContainer*. Value: Borrow Local License Description: A license is borrowed for local use only (without connection to a license server) for a definable period from one *CmContainer* to another. After the period has expired, the licenses automatically reallocates to the server's license pool. A locally borrowed license can not be further transferred
Time	Date stamp on creating the license history item.
Unit Counter	Depending on transfer type: Value: 0 Description: transfer types complete, licenses, borrow complete, borrow local license Value: Number Description: transfer type units: number of units to be transferred.
License Quantity	Depending on transfer type: Value: 0 Description: transfer types complete, units, borrow complete, borrow local license Value: Number Description: transfer type licenses: number of network license quantities to be transferred.
Borrow Expiration Time	Depending on transfer type: Value: 0 Description: transfer types complete, licenses, units Value: date Description: transfer types borrow complete, borrow local license: The BorrowExirationTime displays.
Transfer ID	Display of a unique Transfer ID
CmContainer	Serial number of the *CmContainer*
Update Counter	Infomation on the Update Counter metering at the sender when the license transfer has started.

12.6.10.1 Licenses

A brief example for performing a license transfer of type "Licenses" ('n' from 'm' licenses) covers moving[439] and returning[441] of licenses.

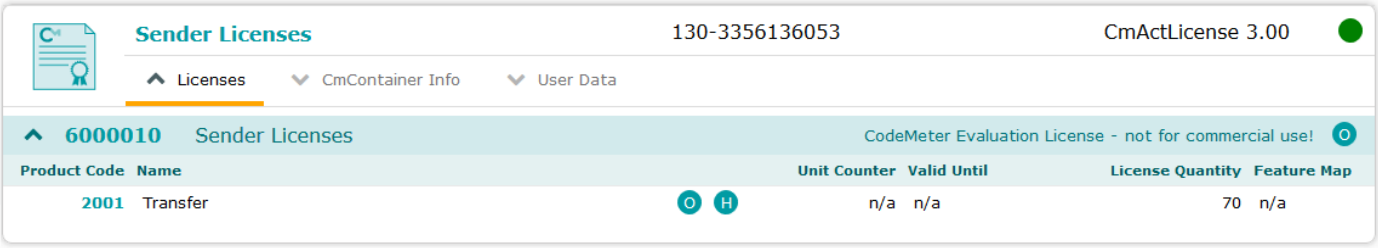
For a sender and a receiver the various *CodeMeter WebAdmin* views will display (*CmContainer*, *Firm Item*, *Product Item* Details, License Monitoring).

30 Licenses are moved and returned.

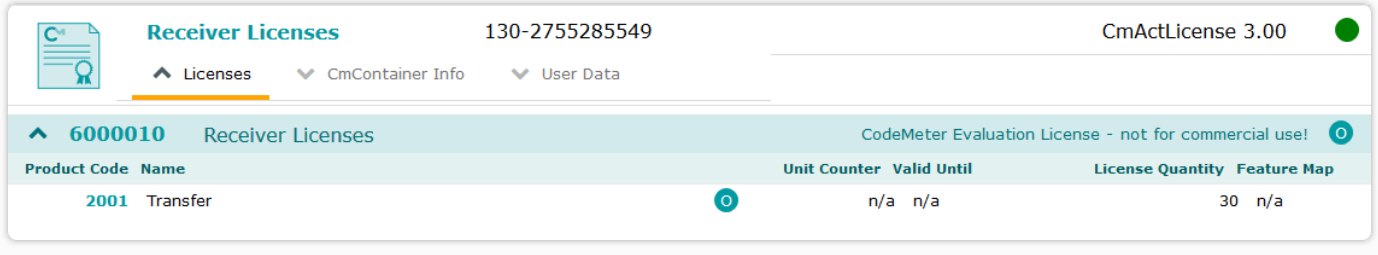
12.6.10.1.1 Move 'n' from 'n' licenses

CmContainer Details

Sender

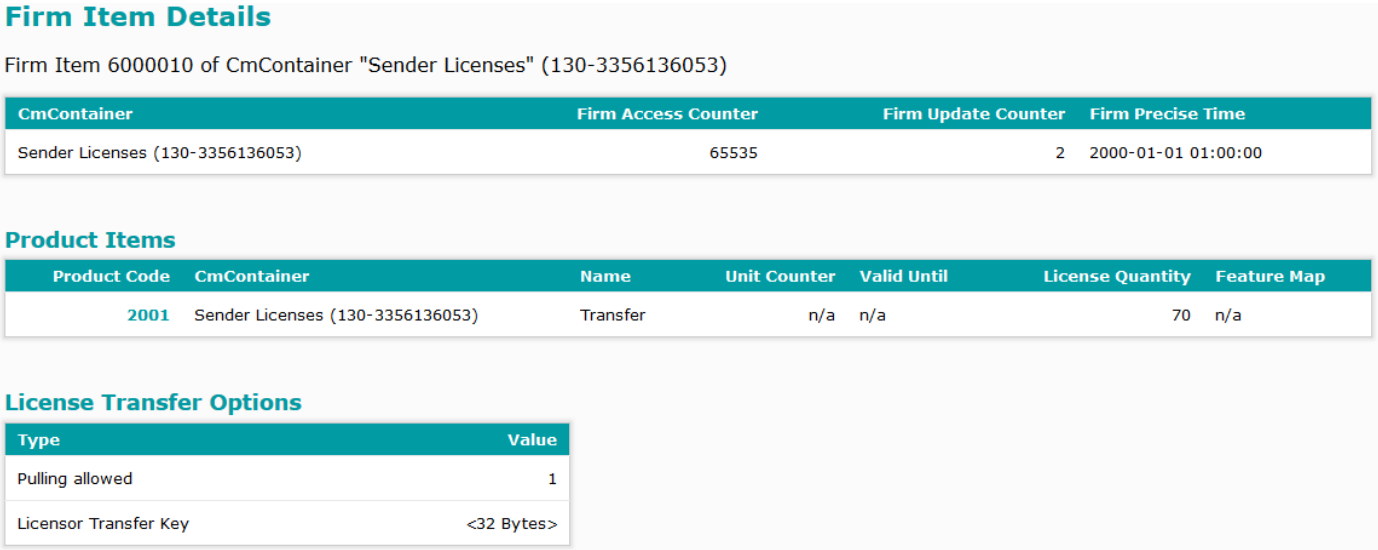


Receiver

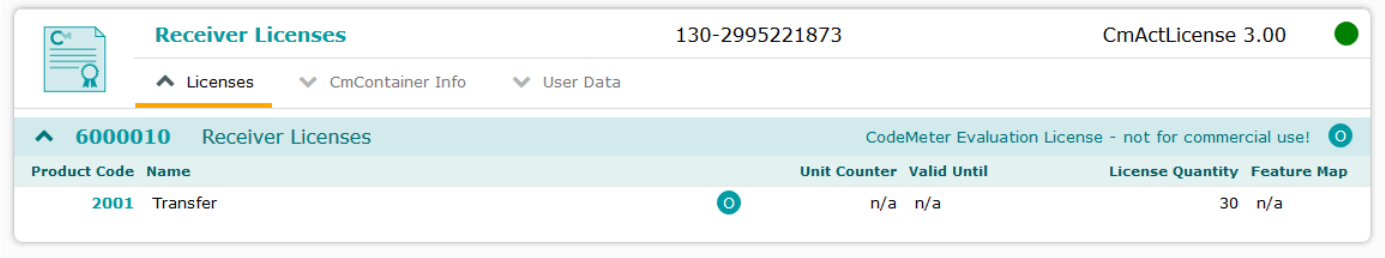


Firm Item Details

Sender



Receiver



Product Item Details

Sender

Product Item Details

Product Item 6000010:2001 of CmContainer "Sender Licenses" (130-3356136053)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		8		Transfer
License Quantity		4		70

License Transfer Options

Option	Value
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Licenses
Transfer Depth	2

License Transfer History

#	Status	Type	Time	License Quantity	Transfer ID	Client (User)	CmContainer	Update Counter
1	Transferred & Receipted	Licenses	2016-01-20 16:24:59	30	efe325c6ef38814495c5	fs1 (WIBU\fs)	130-2995221873	0

Receiver

Product Item Details

Product Item 6000010:2001 of CmContainer "Receiver Licenses" (130-2995221873)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		8		Transfer
License Quantity		4		30

License Transfer Options

Option	Value
License received from	130-3356136053
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Licenses
Transfer Depth	1

License Monitoring

6000010 Vendor 1

Product Code	Name	Feature Map	License Quantity	User Limit (Borrowed)	No User Limit	Exclusive	Shared	Available
2001	Transfer	-	70	0 (-)	0	0	0	70
2001	Transfer	-	30	0 (-)	0	0	0	30

12.6.10.1.2 Return 'n' from 'm' licenses

CmContainer Details

Sender

Sender Licenses 130-3356136053 CmActLicense 3.00

Licenses | CmContainer Info | User Data

6000010 Sender Licenses — CodeMeter Evaluation License - not for commercial use!

Product Code	Name			Unit Counter	Valid Until	License Quantity	Feature Map
2001	Transfer	O	H	n/a	n/a	100	n/a

Receiver

Receiver Licenses 130-2995221873 CmActLicense 3.00

Licenses | CmContainer Info | User Data

6000010 Receiver Licenses — CodeMeter Evaluation License - not for commercial use!

Product Code	Name		Unit Counter	Valid Until	License Quantity	Feature Map
No Product Items available						
Expired and/or returned Product Items						
2001	Transfer	O	n/a	n/a	0	n/a

Firm Item Details

Sender

Firm Item Details

Firm Item 6000010 of CmContainer "Sender Licenses" (130-3356136053)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Sender Licenses (130-3356136053)	65535	2	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
2001	Sender Licenses (130-3356136053)	Transfer	n/a	n/a	100	n/a

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Receiver

Firm Item Details

Firm Item 6000010 of CmContainer "Receiver Licenses" (130-2995221873)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Receiver Licenses (130-2995221873)	65535	2	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
No Product Items available						
Expired and/or returned Product Items						
2001	Receiver Licenses (130-2995221873)	Transfer	n/a	n/a	0	n/a

Invalid

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Product Item Details

Sender

Product Item Details

Product Item 6000010:2001 of CmContainer "Sender Licenses" (130-3356136053)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		8		Transfer
License Quantity		4		100

License Transfer Options

Option	Value
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Licenses
Transfer Depth	2

License Transfer History

#	Status	Type	Time	License Quantity	Transfer ID	Client (User)	CmContainer	Update Counter
1	0	Licenses	2016-01-20 16:24:59	30	efe325c6ef38814495c5	fs1 (WIBU\fs)	130-2995221873	0

Receiver

Product Item Details

Product Item 6000010:2001 of CmContainer "Receiver Licenses" (130-2995221873)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		8		Transfer
License Quantity		4		0, <no network>

License Transfer Options

Option	Value
License received from	130-3356136053
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Licenses
Transfer Depth	1

License Monitoring

6000010 Vendor 1

Product Code	Name	Feature Map	License Quantity	User Limit (Borrowed)	No User Limit	Exclusive	Shared	Available
2001	Transfer	-	100	0 (-)	0	0	0	100
2001	Transfer	-	0	- (-)	-	-	-	0

12.6.10.2 License Borrowing

A brief example for performing a license transfer of type "Borrow Local License" covers borrowing 443 and returning 446 of licenses.

For a sender and a receiver the various *CodeMeter WebAdmin* views will display (*CmContainer*, *Firm Item*, *Product Item* Details, License Monitoring).

1 License is borrowed and returned.

12.6.10.2.1 Borrow

CmContainer Details

Sender

Sender Borrowing 130-3834989529 CmActLicense 3.00

Licenses | CmContainer Info | User Data

6000010 Vendor 1 CodeMeter Evaluation License - not for commercial use!

Product Code	Name		Unit Counter	Valid Until	License Quantity	Feature Map
2001	Licenses to borrow	O H	n/a	n/a	200	n/a

Receiver

Receiver Borrowing 130-3571548377 CmActLicense 3.00

Licenses | CmContainer Info | User Data

6000010 Vendor 1 CodeMeter Evaluation License - not for commercial use!

Product Code	Name		Unit Counter	Valid Until	License Quantity	Feature Map
2001	Licenses to borrow	O	n/a	n/a	1 (local)	n/a

Firm Item Details

Sender

Firm Item Details

Firm Item 6000010 of CmContainer "Sender Borrowing" (130-3834989529)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Sender Borrowing (130-3834989529)	23	43	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
2001	Sender Borrowing (130-3834989529)	Licenses to borrow	n/a	n/a	200	n/a

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Receiver

Firm Item Details

Firm Item 6000010 of CmContainer "Receiver Borrowing" (130-3571548377)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Receiver Borrowing (130-3571548377)	23	43	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
2001	Receiver Borrowing (130-3571548377)	Licenses to borrow	n/a	n/a	1 (local)	n/a

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Product Item Details

Sender

Product Item Details

Product Item 6000010:2001 of CmContainer "Sender Borrowing" (130-3834989529)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		19		Licenses to borrow
License Quantity		4		200

License Transfer Options

Option	Value
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Borrow Local License
Transfer Depth	1

License Transfer History

#	Status	Type	Time	Borrow Expiration Time	Transfer ID	Client (User)	CmContainer	Update Counter
1	Transferred	Borrow Local License	2016-01-21 11:49:21	2016-02-03 13:08:02	b75ae7e7f827e41313cd	fs1 (WIBU\fs)	130-3571548377	0

Receiver

Product Item Details

Product Item 6000010:2001 of CmContainer "Receiver Borrowing" (130-3571548377)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		19		Licenses to borrow
License Quantity		4		1 (local) (LocalOnly)

License Transfer Options

Option	Value
License received from	130-3834989529
Borrow Expiration Time	2016-02-03 13:08:02
Borrow Source	130-3834989529
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Borrow Local License
Transfer Depth	0

License Monitoring

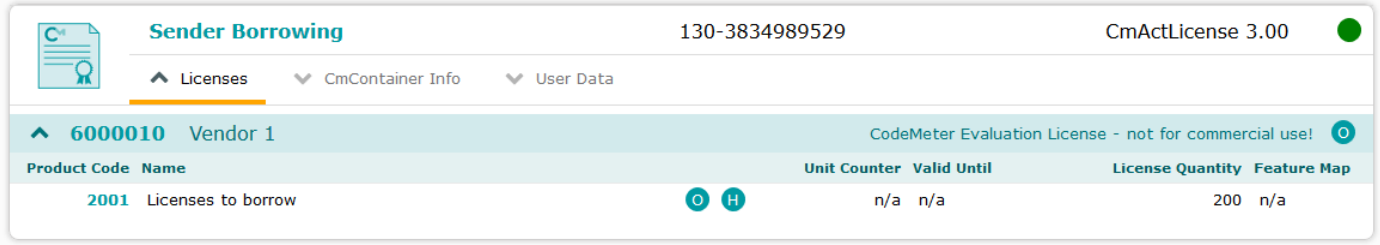
6000010 Vendor 1

Product Code	Name	Feature Map	License Quantity	User Limit (Borrowed)	No User Limit	Exclusive	Shared	Available
2001	Licenses to borrow	-	200	1 (1)	0	0	0	199
2001	Licenses to borrow	-	1 (local)	0 (-)	0	0	0	1

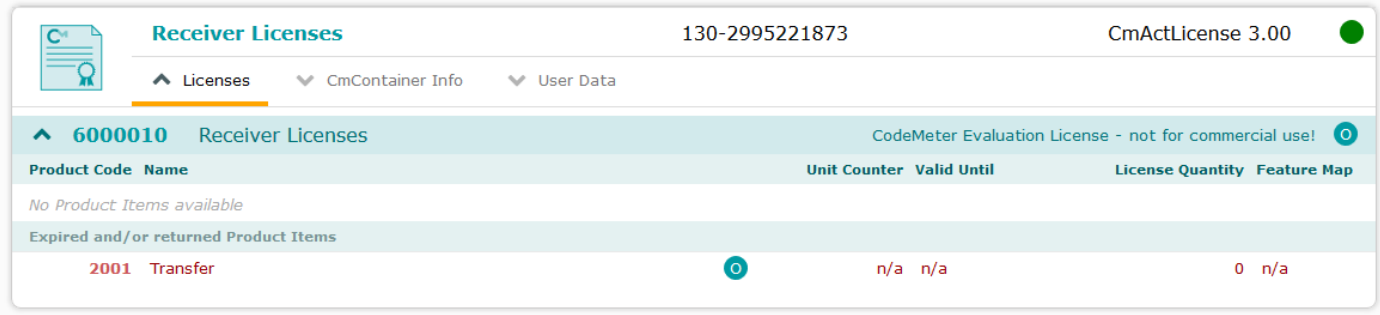
12.6.10.2.2 Return

CmContainer Details

Sender

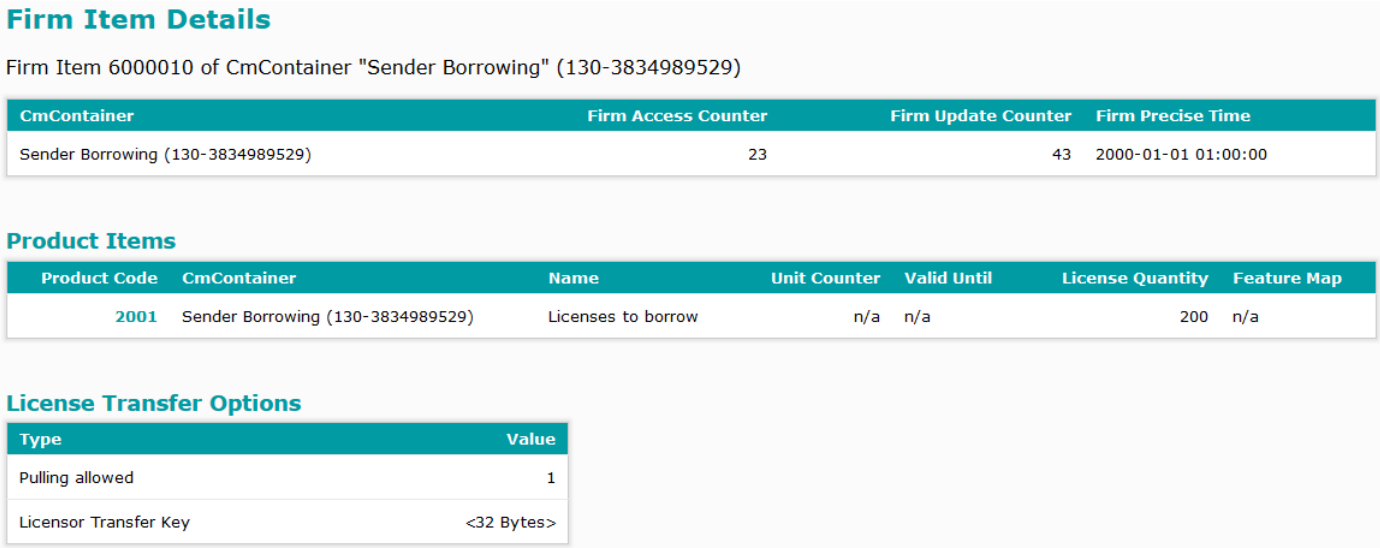


Receiver



Firm Item Details

Sender



Firm Item Details

Firm Item 6000010 of CmContainer "Sender Borrowing" (130-3834989529)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Sender Borrowing (130-3834989529)	23	43	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
2001	Sender Borrowing (130-3834989529)	Licenses to borrow	n/a	n/a	200	n/a

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Receiver

Firm Item Details

Firm Item 6000010 of CmContainer "Receiver Licenses" (130-2995221873)

CmContainer	Firm Access Counter	Firm Update Counter	Firm Precise Time
Receiver Licenses (130-2995221873)	65535	2	2000-01-01 01:00:00

Product Items

Product Code	CmContainer	Name	Unit Counter	Valid Until	License Quantity	Feature Map
No Product Items available						
Expired and/or returned Product Items						
2001	Receiver Licenses (130-2995221873)	Transfer	n/a	n/a	0	n/a

Invalid

License Transfer Options

Type	Value
Pulling allowed	1
Licensor Transfer Key	<32 Bytes>

Product Item Details

Sender

Product Item Details

Product Item 6000010:2001 of CmContainer "Sender Borrowing" (130-3834989529)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		19		Licenses to borrow
License Quantity		4		200

License Transfer Options

Option	Value
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Borrow Local License
Transfer Depth	1

License Transfer History

#	Status	Type	Time	Borrow Expiration Time	Transfer ID	Client (User)	CmContainer	Update Counter
1	0	Borrow Local License	2016-01-21 15:22:58	2016-02-03 13:08:02	535bbf992153b3a7b26d	fs1 (WIBU\fs)	130-2034680140	0

Receiver

Product Item Details

Product Item 6000010:2001 of CmContainer "Receiver Borrowing" (130-3571548377)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		19		Licenses to borrow
License Quantity		4		1 (local) (LocalOnly)

License Transfer Options

Option	Value
License received from	130-3834989529
Borrow Expiration Time	2016-02-03 13:08:02
Borrow Source	130-3834989529
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	No
Transfer Type	Borrow Local License
Transfer Depth	0

License Monitoring

6000010 Vendor 1

Product Code	Name	Feature Map	License Quantity	User Limit (Borrowed)	No User Limit	Exclusive	Shared	Available
2001	Licenses to borrow	-	200	0 (-)	0	0	0	200
2001	Licenses to borrow	-	0	0 (-)	0	0	0	0

12.6.1 Module Items

Module Items are supported for only, if using *Universal Firm Codes* with a number range bigger than 6.000.000.

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

In the area "**Licenses**" of the **Container** navigation item existing *Module Items* display as arrow symbols ( ).

ModuleIT 130-535939122 CmActLicense 3.00

Licenses | CmContainer Info | User Data

6000010 Module Items CAD Inc. CodeMeter Evaluation License - not for commercial use!

Product Code	Name	Unit Counter	Valid Until	License Quantity	Feature Map
2001	CAD Inc.	1000	2016-03-30 23:59:59	100	0x32168
20330101	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000101	CAD Inc. - Components	n/a	n/a	n/a	n/a
20331103	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000504	CAD Inc. - Components	n/a	n/a	n/a	n/a
20430105	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000806	CAD Inc. - Components	n/a	n/a	n/a	n/a
20330307	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000008	CAD Inc. - Components	n/a	n/a	n/a	n/a
20330909	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21008810	CAD Inc. - Components	n/a	n/a	n/a	n/a
9002	Module Items	1000	2016-03-30 23:59:59	100	0x32168
20330101	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21002102	CAD Inc. - Components	n/a	n/a	n/a	n/a
20331103	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000504	CAD Inc. - Components	n/a	n/a	n/a	n/a
20430105	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000806	CAD Inc. - Components	n/a	n/a	n/a	n/a
20330307	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21000008	CAD Inc. - Components	n/a	n/a	n/a	n/a
20330909	CAD Inc. - Setup	52	2016-05-31 09:18:37	n/a	n/a
21008810	CAD Inc. - Components	n/a	n/a	n/a	n/a

Figure 301:*CodeMeter WebAdmin – Container | Lizenzen - Module Items*

On clicking the *Product Code*, which holds the *Module Items*, the **Product Item Details** page opens and in the lower part existing *Module Items* displays.

Product Item Details

Product Item 6000010:2001 of CmContainer "ModuleIT" (130-535939122)

Product Item Options	Type	Size (Bytes)	Dependencies	Value
Text		9		CAD Inc.
Feature Map		4		0000 0000 0000 0011 0010 0001 0110 1000 (0x32168)
Unit Counter		4	data, serial	1000
Aktivierungsdatum		4	data, serial, counter	2015-04-01 00:00:00
Activation Time				. . .
Secret Data	124	48	data, serial, counter	<secret>

Module Items

Product Code	Name
20330101	CAD Inc. - Setup
20330307	CAD Inc. - Setup
20330909	CAD Inc. - Setup
20331103	CAD Inc. - Setup
20430105	CAD Inc. - Setup
21000008	CAD Inc. - Components
21000101	CAD Inc. - Components
21000504	CAD Inc. - Components
21000806	CAD Inc. - Components
21008810	CAD Inc. - Components

License Transfer Options

Option	Value
Pulling allowed	Yes
Returning allowed	Yes
Firm Item at target required	Yes
Transfer Type	Units
Transfer Depth	2

Figure 302:*CodeMeter WebAdmin* – Product Item Details - *Module Items*

On clicking a *Module Item* the usual **Product Item Detail** 405 page opens.

12.7 CmDust

At times, it may necessary to receive help by our support when using *CodeMeter*. In order to ease identification of troubles, the program *CmDust* (**C**ode**M**eter En**du**ser **S**upport **T**ool) for the commandline has been developed.

> No secret information is transferred to Wibu-Systems. You are able to check the information saved in plain text.

CmDust on Windows

Open *CmDust* using the **"Start | All Programs | CodeMeter | Tools"** menu item.

Press "Windows" key to open Start screen | Type "CmDust" | Press "Enter" key.

The result of the program execution is written to the text file `CmDust-Result.log` and saved to the user directory which automatically opens when starting *CmDust*.

Alternatively, you are able to use the commandline application cmu 453 to create a log file.

For analyses this file can be sent to Wibu-Systems.

CmDust on macOS

For macOS you create the *CmDust* file using the *cmu* 452 commandline program. Calling *cmu* is stored in the search path.

To create a *CmDust* log, please proceed as follows:

1. Open *cmu* commandline
2. Type in the following command
   `cmu --cmdust`
   Using the option `--file` allows to add a name and a saving location.
   By default, the file is written to the directory from which you accessed *cmu*.
3. Send this file for analyzing to Wibu-Systems.

CmDust on Linux

For the operating systems Linux you create the *CmDust* file using the *cmu* 452 commandline program. Calling *cmu* is stored in the search path.

1. Open *cmu* commandline
2. Type in the following command
   `cmu --cmdust`
   Using the option `--file` allows to add a name and a saving location.
   By default, the file is written to the directory from which you accessed *cmu*.
3. Send this file for analyzing to Wibu-Systems.

CmDust output

CmDust reads out the following settings:

- Information on the operating system: version, installed service packs, language settings.
- CodeMeter relevant registry entries: installation path, settings of *CodeMeter License Server* and *CodeMeter WebAdmin*, backup and HTTP settings.
- AddOns: information on all *CodeMeter* AddOns.
- Information on CodeMeter and CmContainer: software and hardware version and all entries of connected *CmContainer*.

```
================================================================================
**************************** General Information ******************************
================================================================================
CmDust Version 4.40 Build 660 of 2011-11-10
Copyright (C) 2005-2011 by WIBU-SYSTEMS AG. All rights reserved.

CmDustLog created at 2011-11-17 15:24:40 (UTC)
CmDust was started from: C:\Program Files\CodeMeter\Runtime\bin
Current User has administrator rights
================================================================================
**************************** System Information *******************************
================================================================================
OS: Microsoft Windows 7 Business Edition, 32-bit Service Pack 1 (build 7601)
Computer Name:    FS2.wibu.local
Found IP address: 10.49.12.16 | 192.168.243.1 | 192.168.204.1 | 127.0.0.1
Not running inside Virtual Environment.

Language Settings:
  Machine:      English
  Current User: English

DataExecutionProtection state:
  OPTIN (Only Windows system components and services have DEP applied.)
Current User has administrator rights

Overview of available drives:
  C:\ = Fix Drive  (304336 MB)
  D:\ = CDROM
  E:\ = Removable Drive Bus=Usb;WIBU -  CodeMeter-StickM (7832 MB), contains codemtr.io
  ==============================================================================
************************ Relevant registry entries ******************************
================================================================================
[HKEY_LOCAL_MACHINE\SOFTWARE\WIBU-SYSTEMS\CodeMeter] <All>
RuntimeVersion <All> = "4.40.660.500"
```

12.8CMU - CodeMeter Universal Support Tool

You have also the option to alternatively execute some *CodeMeter Control Center* functions by the commandline based *CodeMeter Universal Support Tool* (*cmu*).

cmu supports you in:

- listing of *CmContainer* contents
- creating a simple test environment for *CmContainer*
- executing a certified time update, and creating and importing of license request and update files (Context Files and Update Files, `*.WibuRaC` and `*.WibuRaU`).

Call *cmu* in the directory `%\Program Files%\CodeMeter\Runtime\bin` using the command `cmu[32].exe`.

Alternatively, on Windows call *cmu* by the 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.

For the operating systems macOS and Linux this command is provided by the usual search path parameter.

The following list shows all existing *cmu* commands.

Command	Description
/h or **--help**	shows this help in the commandline window.
/v or **--version**	shows the versions of all available *CodeMeter* components.
/l or**--list**	lists all connected *CmContainer* by way of their serial numbers.
/x or **--list-content**	lists the contents of all connected *CmContainer*.
/k or **--list-server**	lists all available network license server.
/n or **--list-network**	lists the network license information of the own server.
	lists network license information also of remote *CodeMeter* server. **-list-network [--server <servername>|--all-servers] [--serial <serial>] [--firmcode <firmcode> [--productcode <productcode> [--featuremap <featuremap>]]]**
	--all-servers lists the network license information for all found servers.

Command	Description	
	--server <servername>	lists the network license information for the specified server **<servername>**.
	--serial <serial> **--firmcode <firmcode>** **--productcode <productcode>** **--featuremap <featuremap>**	specified parameter configure the output according to criteria selected.

--add-server	adds a server to the end of your server search list. As an argument pass the server's name, IP or CmWAN URL.
	Examples:
	`server.domain.local`
	`192.168.0.72`
	`fe80::ea06:88ff:fecf:df6f`
	`https://user:secretpassword@server.domain.local/cmwan`	

--delete-server	deletes a server from the server search list. As an argument pass the server's name, IP or CmWan URL as it is listed.	
-- clear-serversearchlist	deletes all the entries from the server search list. To reactivate automatic server broadcast, please add broadcast using option "--add-server 255.255.255.255".	
--show-serversearchlist	shows the entries in the server search list.	
/c <FI> or **--context <FI>**	creates a license request for an *license update via CmFAS* 394 creates a license request for an license update via *Firm Item* <FI>. Using option **--file** specifies the output file. If no option is set the standard output is used (stdout).	
/i or **--import**	imports a license update file received via CmFAS 397 for the available *CodeMeter* license. Using option **--file** specifies the file name. The update can cover a *CmDongle* or a *CmActLicense* license file.	
/d or **--firmware-update**	starts the firmware update of a *CmContainer*.	
/u or **--time-update**	starts the update of the Certified Time in each connected *CmContainer*.	
/e <s> or **--enable <s>**	allows the activation or deactivation of the selected *CmContainer*. Specifying the *CodeMeter* password is required. The required new *Enabling* status is specified by the parameter **<s>**. Parameter values cover 1 (disable), 2 (temporary enable), 3 (enable).	
/t <no> or **--test<no>**	starts some simple tests for each connected *CmContainer*. The number of tests is specified by parameter <no>. It is required that the *CmContainer* must be (temporarily) enabled.	
/vv or **--cmdust**	creates a *CmDust* report. This report is useful and required when requesting support. Wibu-Systems recommends to create a *CmDust* report before contacting the support. Using the option **--file** writes the result into a text file.	
--borrow	allows the borrowing of licenses from a license server to the local PC. You have to specify the *Firm Code* and the *Product Code* of the license using the options **--firmcode** and **--productcode**. As an additional option you may specify the *Feature Map* using the option **--featuremap**. Moreover, you have to specify the serial number of the client *CmContainer* and the server name using the options **--serial** and **--server**.	
--return	returns the borrowed license to the license server. You have to specify the *Firm Code* and the *Product Code* of the license using the options **--firmcode** and **--productcode** and the serial number of the client *CmContainer* and the server name using the options **--serial** and **--server**.	
--borrowlist	lists the borrowed licenses for the client and the server.	
--transferlist	creates a listing of the license transfer relevant data.	
--enabling	lists the enabling stati of all connected *CmContainer*. Combined with the command **-x** you can also display additional enabling information of the *CmContainer* content.	
--create-io	is used in combination with the option **--file** and makes sense only when using the hardware form factors *CmCard/SD* or *CmCard/CF*. A new `codemtr.io` file is created. Please call this command only if the `codemtr.io` file is deleted.	
--detect-proxy	prints the system proxy to standard output.. Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.	
--delete-cmact-license	deletes a *CmActLicense* license you specify using the command **--serial**. Once you delete a *CmActLicense* license it cannot be restored.	
--set-access-data	activates *WebAdmin* write authentication and saves the password. Use with option **--password** to define password.	
--set-proxy	sets *CodeMeter* proxy configuration. As argument pass the name or IP address of the proxy server. A port different to the default port (80) may be defined by **--port** followed by the port number. A user password authentication may be set by **--username** `<name of the user>` **--password** `<password>`. To remove the authentication use **--username** followed by **""**.	

Command	Description		
	--use-system-proxy	on setting the proxy the system proxy is used. In the profiling or the `server.ini` file then the entry `UseSystemProxy` is set to a value of **1**. The entry has the value `0`, if **--set-proxy** is used. Please note, that under Linux, the environment variable `http_proxy` is considered as *system proxy*. `https_proxy` is not yet supported.	
	Examples: **cmu --set-proxy** `proxy.company.com` --username `johndoe` **--password** `mypassword` **--port** `90` **cmu --set-proxy** `proxy.company.com` **--username ""**		
--reset-access-data	deletes both passwords for *WebAdmin* authentication (read- and write-passwords) and sets default (no read authentication, no write authentication). "Local Access only" will be set.		
--device-id	sets this parameter on import of WibuCmLiF files (`--import`) with binding scheme "Binding Extension". The input of the `device-id` is as 128 hex digits preceeded by "0x".		

License transfer options

All calls existing in CodeMeter Core API for the licenses transfer feature can also be called using *cmu*. Here the respective files are used (`WibuCmLIF`/`WibuCmRaC`/`WibuCmRaU`). The following *cmu* calls exist:

Command	Description	
--create-lt-context <parameters>	Enables to create a license transfer context. Used with `--lt-push` parameters to specify the nature of the transfer - license update with FSB if omitted. Additional parameters:	
	--lt-request-file <file>	is a mandatory parameter. It is used to specify the file `<file>` where the result of the license context creation should be written.
	--lt-context-file <file>	It is used to specify the file `<file>` containing the license context to be created.
	--serial <serial> or -s <serial>	selects the *CmContainer* with the specified *Serial Number*.
	--firmcode <fc>	sets the *Firm Code* of the transferred license.
	--productcode <pc>	sets the *Product Code* of the transferred license.
--create-lt-update <parameters>	Enables to transfer a license transfer context. Used with one of following parameters to specify the nature of the transfer:	
	--lt-move-licenses <licenses>	the specified number of licenses wil be transferred.
	--lt-borrow-local-license <expiration time>	expiration time as `<YYYY>-<MM>-<DD>[T<hh>:<mm>:<ss>]` (one license will be transferred for the specified time limit).
	--lt-renewborrow <expiration time>	time validity of an already borrowed license will be extended.
	Additional mandatory parameters:	
	--lt-request-file <file>	is a mandatory parameter. It is used to specify the Context File `<file>` containing the license content to be transferred.
	--lt-update-file <file>	is a mandatory parameter. It is used to specify the Update File `<file>` where the result of the license transfer should be written.
	--serial <serial> or -s <serial>	selects the *CmContainer* with the specified *Serial Number*.
	--firmcode <fc>	sets the *Firm Code* of the transferred license.
	--productcode <pc>	sets the *Product Code* of the transferred license.
	Additional optional parameters:	
	--lt-feature-code <number>	It is used to specify the *Feature Code* `<number>`. The default input for the `<number>` is in decimal unless it begins with `0x` or `0X`.
	--lt-product-ref <number>	It is used to specify the *Product Item Reference* `<number>`. The default input for the `<number>` is in decimal unless it begins with `0x` or `0X`.
--import-lt-update <parameters>	Updates license transfer data on the target side. Used with either `--lt-push`, `--lt-pull`, `--lt-fsb` or `--lt-return` parameters to specify the nature of the transfer. Additional parameters:	
	--lt-update-file <file>	is a mandatory parameter. It is used to specify the Update File `<file>` where the result of the license update should be written.

Command	Description	
	--serial <serial> or -s <serial>	optionally selects the *CmContainer* with the specified *Serial Number.*
	--firmcode <fc>	optionally sets the *Firm Code* of the transferred license.
	--productcode <pc>	sets the *Product Code* of the transferred license (optional, only with **--firmcode**).
--create-lt-receipt <parameters>	Creates a signature with a defined private key and gives it back as a receipt to specify the nature the transfer. Additional parameters:	
	--lt-receipt-file <file>	is a mandatory parameter. It is used to specify the Context File `<file>` where the receipt will be transferred to.
	--serial <serial> or -s <serial>	selects the *CmContainer* with the specified *Serial Number.*
	--firmcode <fc>	sets the *Firm Code* of the transferred license.
	--productcode <pc>	sets the *Product Code* of the transferred license.
--import-lt-receipt <parameters>	Checks the receipt with a defined public key confirming the validity of a transaction. Used with one of following parameters to specify the nature of the transfer. Additional parameters:	
	--lt-receipt-file <file>	is a mandatory parameter. It is used to specify the Context File `<file>` where the data will be transferred from.
	--serial <serial> or -s <serial>	selects the *CmContainer* with the specified *Serial Number.*
	--firmcode <fc>	sets the *Firm Code* of the transferred license.
	--productcode <pc>	sets the *Product Code* of the transferred license.
--lt-cleanup <parameters>	Possible cleanup action	
	--deleted	clean deleted licenses
	--disabled	clean disabled (dangerous)
	--hiddenhistory	clean history of given product item
	--container	removes whole license container (dangerous)
	Additional parameter	
	--serial <serial> or -s <serial>	selects the CmContainer with the specified Serial Number.
	--firmcode <fc>	selects the *Firm Code* of the transferred license.
	--productcode <pc>	selects the *Product Code* of the transferred license.
	--lt-product-ref <number>	It is used to specify the Product Item Reference `<number>`. The default input for the `<number>` is in decimal unless it begins with `0x` or `0X`.

Named User licenses

The values of the username and the domain are automatically set.

Options	Description
--nmu-userdefined <text>	Specifies via **<text>** a case-sensitive UTF-8 string for the user-defined text. The string is truncated after 127 bytes.

Additional *cmu* options

The following list shows additional *cmu* options:

Options	Description
/f <file> or **--file <file>**	Additional option which writes the command result into a file `<file>`. This option is used in combination with the commands **--context**, **--import**, **--cmdust**.
/s <serial> or **--serial <serial>**	Additional option which defines that a command is valid only for a *CmContainer* specified by its serial number `<serial>`, e.g. "1-10234242".
/p <pwd> or **--password <pwd>**	Additional option in combination with the commands **--enable** and **--firmware-update**. This option defines the required *CodeMeter* Password for this command.
--firmcode <fc>	Additional option in combination with the commands **--borrow** or **--return** specifying the *Firm Code* of the borrowed license.
--productcode <pc>	Additional option in combination with the commands **--borrow** or **--return** specifying the *Product Code* of the borrowed license.
--featuremap <fm>	Additional option in combination with the commands **--borrow** or **--return** specifying the *Feature Map* of the borrowed license.
--server <servername>	Additional option to borrow a license from another server. Is used in combination with command **--borrow**.
--write	Additional option used in combination with the command **--detect-proxy** which saves the setting using the *CodeMeter* profiling. These settings are written only if no proxy has been previously set in the profiling. For

Options	Description
	overwriting the settings use the option **--force**.
--force	Additional option used in combination with the command **--detect-proxy** which overwrites already existing proxy settings in the *CodeMeter*®profiling.
--show-config-disk	Shows the current settings of removable/fixed drives or of the type of the defined Master Boot Record (MBR). This option concerns the behavior of virtual flash memory partitions. Use only for *CmStick* and *CmStick/M*.
--set-config-disk <parameter>	Allows to define a special behavior of virtual flash memory partitions, e.g. drive settings, boot code or activations (*CmDongle* only). Please note that replugging of the *CmDongle* is required. Description / Parameter: Drive settings: `RemovableDisk,LocalDisk` Boot Code: `Int18Boot,ZeroBoot,LoopBoot,SwapBoot,VbrBoot` Activation: `ActivePartition,InactivePartition` FAT: `Fat16,Fat32` USB-Communication Device Class: `HidCommunication; MsdCommunication`
--check-cm-integrity	Allows to check the *CodeMeter* signature.
--licensing-terms	Print licensing terms of GNU Lesser General Public License (LGPL) 3.0.

Application examples

Action	Parameter
Displaying *cmu* options	`Cmu[32].exe -h`
Creating a *CodeMeter* Remote Activation Context File (here:`1-1040870.WibuCmRaC`) for the *Firm Code* 10 (*Firm Item level*)	`Cmu[32].exe -c10 -f1-140870.WibuCmRaC`
Importing a *CodeMeter* Remote Activation Update File (here:`1-1040870.WibuCmRaU`) --> reprograms the connected *CmContainer*	`Cmu[32].exe -i -f1-1040870.WibuCmRaU`
Showing the versions of current *CodeMeter* components.	`cmu32 --version`
Listing all available *CodeMeter* network license server and if existing all related licenses.	`cmu32 --list-server --list-content`
Starting 100 simple tests. The tests are executed only for the *CmContainer* specified by the serial number of `1-233232`.	`cmu32 --test 100 --serial 1-233232`
Changing the enabling status to "**temporarily enabled**" for the *CmContainer* `1-2345` by using the *CodeMeter* password "SECRET".	`cmu32 --enable2 --serial 1-2345 --password SECRET`

12.9CodeMeter License Tracking

Starting with Version 4.50 *CodeMeter* introduces license tracking allowing for the evaluation of licensing data based on structured logfiles. With it the actual use of licenses is recorded.

However, Wibu-Systems does not offer a separate application for license tracking but suggests that software vendors who want to evaluate how their licenses are used refer to tools by third parties able to aggregate information from real-time requests or logfiles.

Secure Licence Tracking

Starting with Version 5.20 *CodeMeter* supports Secure License Tracking. This ensures that license access log data is written manipulation-safe. Manipulation is prevented by authenticated check of data integrity using signatures.

For analyzing license access data not a single log file is created but separate logfiles for each *Firm Code*. Integrity and authenticity of the log files preventing tampering is ensured by signatures and a subsequent signature check.

Validation

In order to validate either the contents of a given signed *CodeMeter* log file or a sequence of log files located in a given directory use the option `/vslf` 336 in *CmBoxPgm* the developer tool for local programming of *CmContainer* using a commandline (console).

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.
	You obtain the required values from one of the created *CodeMeter* log files.

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`

Command	**/vslf** - Validation Of Signed Log Files
	the following <public key file> is derived: 5,20,10,0xa809304778d517c44a22d65e1fcedd51a4e2a956fa89e93bb1a24e2100000000a2ad17e685306d6e15eb6b7ebc8cc72ebc97c0f52721b584836696de00000000

Currently, the logfile content is saved locally but for future version its is planned that contents may also be retrieved using HTTP access and calls (real-time history).

> If the logfiles need to be read from other systems, you must share the folder where the logfiles are stored as read-only in your local area network.

The following sections briefly:

- show how to configure License Tracking 457
- introduce definitons and value ranges used in the logfile 458
- describe single logfile entry types

12.9.1 Requirements and Configuration

Using the *CodeMeter* feature License Tracking requires at least *CodeMeter License Server* Version 4.50.

Using the feature Secure License Tracking requires at least *CodeMeter License Server* Version 5.20.

Configuration

The logging of licensing data must be activated together with CodeMeter License Server. This you do by direct activation in the *CodeMeter* Profiling environment.

Profiling

For Windows operating systems you find the profiling entries stored in the registry, for other operating systems entries are set in the file `server.ini`. The following table shows you the respective locations.

Operating system	Registry / Server.ini Entry
Windows	`HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion`
macOS	`/Library/Preferences/com.wibu.CodeMeter.Server.ini`
Linux	`/etc/wibu/CodeMeter/Server.ini`

There exist two relevant profiling entries for *License Tracking*.

Entry	Property	Value
`LogLicenseTracking`	`[DWord]`	`[0;1]` Default value is is `0` and Logging for License Tracking is disabled.
`LogLicenseTrackingPath/[Firm Code]/`	`[SZ]`	`<path>` Default path on Windows operating systems is `%ProgramData%\CodeMeter\LicenseTracking`. For each Firm Code a separate directory is created. For example, the log file for *Firm Code* `10` is saved to the directory `<LogLicenseTrackingPath>/FC10/`. For other operating systems the default path has the same value of the general profiling entry `LogPath`.
`LogLicenseTrackingLogRotationSizeInMb`	`[DWord]`	`[1..3500]` If a license tracking log file exceeds a size of `x` MB, the rotation starts. The input range of `x` is from `1` to `3500` MB. The default value is `1000` MB. If the specified value is outside the input range, automatically the default value applies.
`LogLicenseTrackingLogRotationTimeInMinutes`	`[DWord]`	`[0..525600]` If the oldest entry of a license tracking log file is older than n minutes, the rotation starts. The input range of n is from `0` to `525600` minutes (approx. 1 year). Please note that after specifying n, the log file starts to rotate after a time delay of about 1 minute. The default value is `0` minutes. Then time-based log rotating is deactivated. If the specified value is outside the input range, automatically the default value applies.

> Please note that you must stop the *CodeMeter License Server* service, make the change, and then restart the *CodeMeter License Server* service before the change can take effect.

Logfile Rotation

In order to manage the constant growth of log files efficiently, *CodeMeter* offers a rotating system in the area of license tracking.

This system can be set via two parameters in profiling.

- `LogLicenseTrackingLogRotationSizeInMb`

  If a license tracking log file exceeds a size of x MB, the rotation starts.

  The input range of x is from 1 to 3500 MB.

  The default value is 1000 MB.

  If the specified value is outside the input range, automatically the default value applies.

- `LogLicenseTrackingLogRotationTimeInMinutes`

  If the oldest entry of a license tracking log file is older than n minutes, the rotation should start.

  The input range of n is from 0 to 525600 minutes (approx. 1 year).
  Please note that after specifying n, the log file starts to rotate after a time delay of about 1 minute.

  The default value is 0 minutes. Then time-based log rotating is deactivated.

  If the specified value is outside the input range, automatically the default value applies.

12.9.2 Logfile Format

The following logic applies to the format of the logfile.

1. Each line in the logfile can be handled separately. There exist separate lines for different entry types 459.
2. Each line that does not match to the described formats has to be ignored.
   This will allow us to enhance the output in future versions without causing trouble in working solutions.

> It is also recommended to do a parsing of the different arguments of a line and simply to ignore arguments that are not known.
> This allows us to enhance the output in future versions without causing trouble in working solutions.

12.9.2.1 Definitions and Value Ranges

For the logfile and single entry types the following definitions and value ranges are used:

Definition	Value Range
`access ID`	`string` The `<access ID>` is given by the server and extends the `<license ID>` by an index describing the slot, i.e. `<license ID>-<slot ID>`.
`application ID`	`[0..4294967295]`
`application text`	`string`
`enabling block index`	`[0..31]`
`expiration time`	`["never"|UTC Timestamp]`
`feature map`	`[0..4294967295]`
`firm code`	`[0..4294967295]`
`license ID`	`string` The `<license ID>` is automatically derived as `<mask>-<serial number>-<firm code>-<product item reference>`, e.g. “`2-1500002-100532-18`”. The `<license ID>` is a unique IDentifier for a license entry.
`license quantity`	`[0..4294967295]`
`logfileID`	`[0..4294967295]` ID value of the log file. In order to prevent that a single or several log files can be deleted each log file must have a separate ID.
`mask`	`[0..65535]`
`product code`	`[0..4294967295]`
`product item reference`	`[0..4294967295]`
`product item text`	`string`
`serial`	`[0..4294967295]`
`server`	`string`
`slot ID`	`[0..4294967295]`
`timestamp`	`UTC Timestamp` UTC Timestamp sample: “`2012-12-24T08:32:59`”.

Since the strings may contain quotation marks (") but may also be bracketed expressions, any quotation marks that are part of the string are quoted by a backslash (\). For example, the *application text The best of "John Doe."* will be issued as

```
...AppText: "The best of \"John Doe.\""
```

12.9.3 Entry Types

The *CodeMeter* license tracking logfile knows the following listed entry types.

List of Licenses 459
License 459
Access 459
Release 460
Borrow Access 460
Borrow Return 460
Denial 460
Administrative 460
SignedLogfile 460
Signature

12.9.3.1 List of Licenses Entry

Entry type	*List of Licenses* entry
Description	A list of *License* entries is preceded by a *List of Licenses* entry. This indicates that in the following lines all existing licenses of this server are listed. A previously retrieved list of *License* entries becomes invalid.
Writing time	The *List of Licenses* entry is written immediately before the list of *License* entries is written.
Syntax	`<timestamp> ListOfLicenses`

12.9.3.2 License Entry

Entry type	*License* entry
Description	The *License* entry describes an existing license.
Writing time	All *License* entries are written to the logfile: • on startup of *CodeMeter License Server* • each time when an entry is changed, e.g. by plugin / plugout or remote programming. In the cases mentioned above, all *License* entries of the current server are written preceded by a *List of Licenses* 459 entry.
Syntax	`<timestamp> License Server:"<server>", LicenseID:<license ID>, SN:<mask>-<serial>, FC:<firm code>, PC:<product code>, FM:<feature map>, ET:<expiration time>, LQ:<license quantity>, PT:"<product item text>"`

Before all *License* entries are re-written on changing entries all allocated licenses are released by a *Release* entry. Immediately after issuing the *License* entries the previously released licenses are again allocated by an *Access* entry.

This is necessary because `license IDs` can change on re-programming or on plugout and the subsequent rebooking. Moreover, the `access ID` may change by automatic rebooking after plugout.

Licenses with a *License Quantity* value of `0` (license for local use use) are not listed.

The *Expiration Time* contains the minimum of the Product Item Option *Expiration Time* and the value of an activated Product Item Option *Usage Period*. If neither an *Expiration Time* is set nor a *Usage Period* exists or is activated the value is `"never"`.

12.9.3.3 Access Entry

Entry type	*Access* entry
Description	An *Access* entry describes that a license on a server is allocated to a user.
Writing time	The *Access* entry is written at the moment a license is accessed.
Syntax	`<timestamp> Access Server:"<server>", LicenseID:<license ID>, AccessID:<access ID>, Client:"<computer name>", User:"<user name>", AppID:<application ID>, AppText:"<application text>"`

The *application ID* and *application text* are derived from `CMCREDENTIAL` structure using `mulUserDefinedID` and `mszUserDefinedText`.

12.9.3.4 Release Entry

Entry type	*Release* entry
Description	A *Release* entry describes that a user has released a formerly accessed license on a server.
Writing time	The *Release* entry is written at the moment a license is released.
Syntax	`<timestamp> Release Server:"<server>", AccessID:<access ID>`

12.9.3.5 Borrow Access Entry

Entry type	*Borrow Access* entry
Description	A *Borrow Access* entry describes that a user has borrowed a license from a server.
Writing time	The *Borrow Access* entry is written at the moment a license is borrowed. In addition, the *Borrow Access* entry is written when *CodeMeter License Server* is started and there already exist borrowed licenses.
Syntax	`<timestamp> Borrow Server:"<server>", LicenseID:<license ID>, BorrowID:<borrow ID>, Client:"<computer name>", User:"<user name>", Expires:<expiration time>, BorrowSn:<mask>-<serial>`

12.9.3.6 Borrow Return Entry

Entry type	*Borrow Return* entry
Description	A *Borrow Return* entry describes that either a user has returned a borrowed license on a server or the borrow duration has expired and the license was returned automatically.
Writing time	The *Borrow Return* entry is written at the moment a license is returned.
Syntax	`<timestamp> Return Server:"<server>", BorrowID:<borrow ID>`

12.9.3.7 Denial Entry

Entry type	*Denial* entry
Description	A *Denial* entry describes that a user requested a license but did not get one because no more licenses could be allocated. It will not show license requests of licenses that do not exist on this server.
Writing time	The *Denial* entry is written at the moment a license access has failed.
Syntax	`<timestamp> Denial Server:"<server>", LicenseID:<license ID>, Client:"<computer name>", User:"<user name>", AppID:<application ID>, AppText:"<application text>"`

A *Denial* entry is only logged if error 212 (`CMERROR_NO_MORE_LICENSES`) occurs.

12.9.3.8 Administrative Entry

Entry type	*Administrative* entry
Description	An *Administrative* entry describes some event on the *CodeMeter License Server*.
Writing time	The *Administrative* entry is written at the moment the described event occurred.

Syntax	`<timestamp> Admin Server:"<server>" CodeMeter_started <timestamp>`
	is written only on start of the TMR Setup and also only in the first logfile for each *Firm Code*.
	`<timestamp> Admin Server:"<server>" CodeMeter_stopped`

If *CodeMeter License Server* is stopped, all *Access* entries are automatically canceled. Only *Borrow Access* entries remain valid and will be restored on next start of *CodeMeter License Server*. Usually, the *Release* entries are automatically added to the log, but in some circumstances this is not possible, e.g. killing *CodeMeter License Server*.

12.9.3.9 SignedLogfile Entry

Entry type	*SignedLogfile* entry
Description	The *SignedLogfile* entry corresponds to the header and holds the Public Key.
Writing time	The *SignedLogfile* entry is written at the moment the logfile is created.

Syntax	`<timestamp> SignedLogfile FirmCode: <FirmCode>, PublicKey: <PublicKey>, Runtime-Version: <Version>, LogfileID: <LogfileID>` `<timestamp>` UTC time stamp: `<YYYY>-<MM>-<DD>-<hh><mm><ss>`. `<FirmCode>` *Firm Code* of the content of this logfile. `<PublicKey>` Public Key belonging to the Private Key used to sign the logfile. `<Version>` *CodeMeter License Server* Version which generated the logfile. The version specification follows the short format: (`<Major>.<Minor two-numbered>.<Build>.<Count>`) `<LogfileID>` ID value of the logfile. In order to prevent that one or several logfiles are deleted, each logfile must have an ID. Each logfile created during a single running instance of *CodeMeter License Server* has the same `LogfileID`. The latest `LogfileID` is written from the profiling value "`SignedLogfileID`" and is incremented on the next start of *CodeMeter License Server*. If no profiling entry "`SignedLogfileID`" is found an initial value of 1 is assumed. e.g. `2014-02-07T10:34:33 SignedLogfile FirmCode:10, PublicKey:b4342ec15183992be75ee5e702ea7d118ebb489046df1b15393cdf8d000000007b3a92afafe35f6505222841d65610e75749bf9d572a0eea83d1d6be00000000, Runtime-Version:5.11.1343.201, LogfileID:13`

12.9.3.10 Signature Entry

Entry type	*Signature* entry
Description	The *Signature* entry hold the signature of the section last written.
Writing time	The *Signature* entry is written at the moment the section is signed.
Syntax	`<timestamp> Signature Signature: <Signature>` `<timestamp>` UTC time stamp: `<YYYY>-<MM>-<DD>-<hh><mm><ss>`. `<Signature>` Calculated signature value of the logfile using the secret signature key (Private Key). This value allows anybody to check the integrity of the logfile using the public verification key (Public Key 461). e.g. `2014-02-07T10:34:43 Signature Signature:75998652881c0c56ce7b391c3638c1a5540e12cab282e2c3c82a0a0a000000008cf46fd7a025939b7d86dfff8b4bdc01073da2eed7326bc351a335cb00000000`

12.1 HID Support

Starting with Version 5.0 *CodeMeter* supports devices that conform to the USB's Human Interface Device (HID) class specification.

The installation of a special USB host driver is not required since the communication via the USB HID class is standardized and the operating systems provide respective classes. Currently, the operating systems Windows, macOS, and Linux are supported.

Alternatively to the Mass Storage Device status, thus *CmDongles* can display as HID without a drive status.

> The communication class HID is available for many *CmDongle*. Please consult the respective data sheet for support details.

Requirements

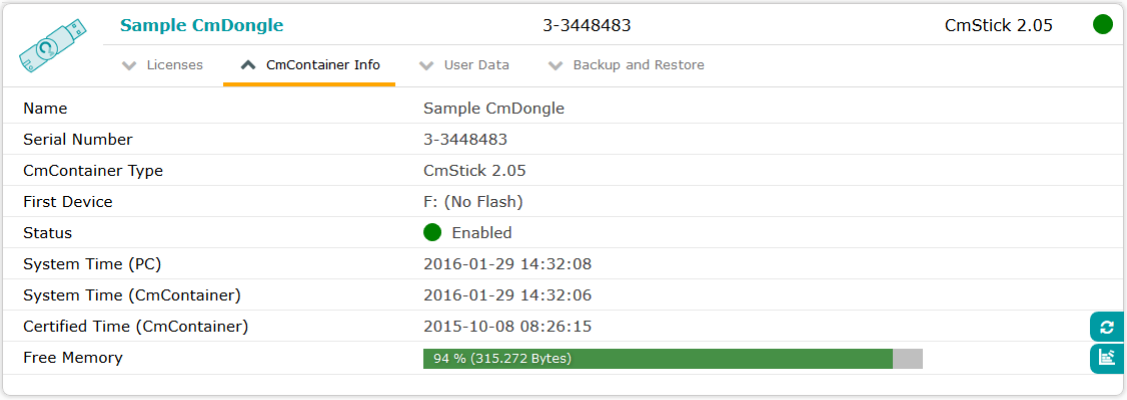
- Minimum *CodeMeter* Firmware 2.02
- Minimum *CodeMeter* Runtime 5.0

The USB communication standard can be switched any time from Mass Storage Device (MSD) to Human Interface Device (HID) or vice versa.

12.10. Set from Mass Storage to HID

To switch the USB communication standard from Mass Storage Device (MSD) to Human Interface Device (HID), please proceed as follows:

1. View the status in *CodeMeter WebAdmin* on page "**Content | CmContainer**".
   A drive is assigned and no flash memory is available.



2. Call *cmu* 452.

   For Windows OS call *cmu* call *cmu* by the 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). For the operating systems macOS and Linux this command is provided by the usual search path parameter.

3. Enter the following commandline:

```
cmu32 /s [Box mask-Serial number] --set-config-disk HidCommunication
```

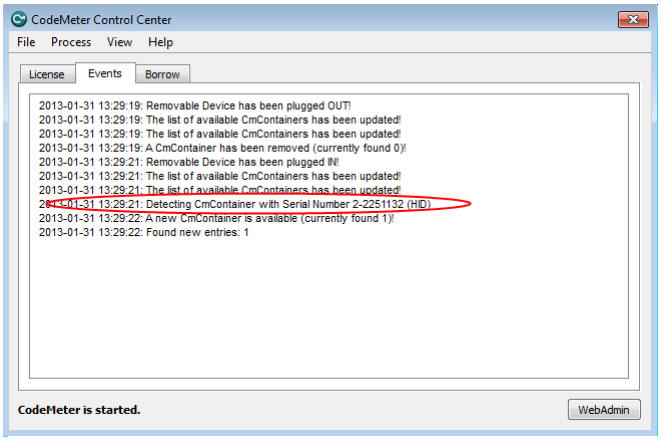
   The current status displays in the following commandline output:

```
cmu32 - CodeMeter Universal Support Tool.
Version 5.00 of 2013-Jan-30 (Build 1039) for Win32
Copyright (C) 2007-2013 by WIBU-SYSTEMS AG. All rights reserved.

- CmStick/C with Serial Number 2-2251132 and version 2.01
  Version:            2.01
  Flash Size:         no real flash available
  Virtual Drive:      E:
  Configuration:      LocalDisk with ActivePartition
  File System:        FAT32
  Communication:      Mass Storage Device
  Boot-Code:          Int18 Boot Code
  Mdfa:               0x539

  Please replug your CmDongle to apply the changes.
```

4. Unplug and replug the *CmDongle*.

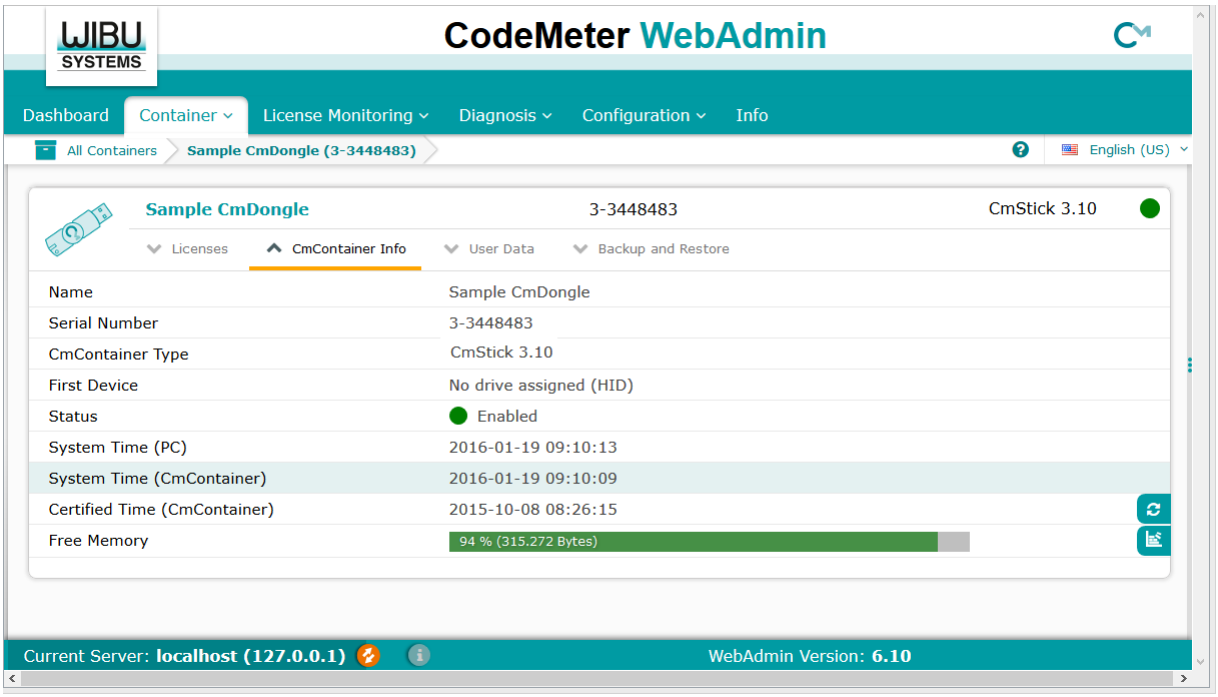
5. View logging in *CodeMeter Control Center* tab "**Events**".

   The information for the switch to HID displays.



5. Check in *CodeMeter WebAdmin* page "**Content | CmContainer**".

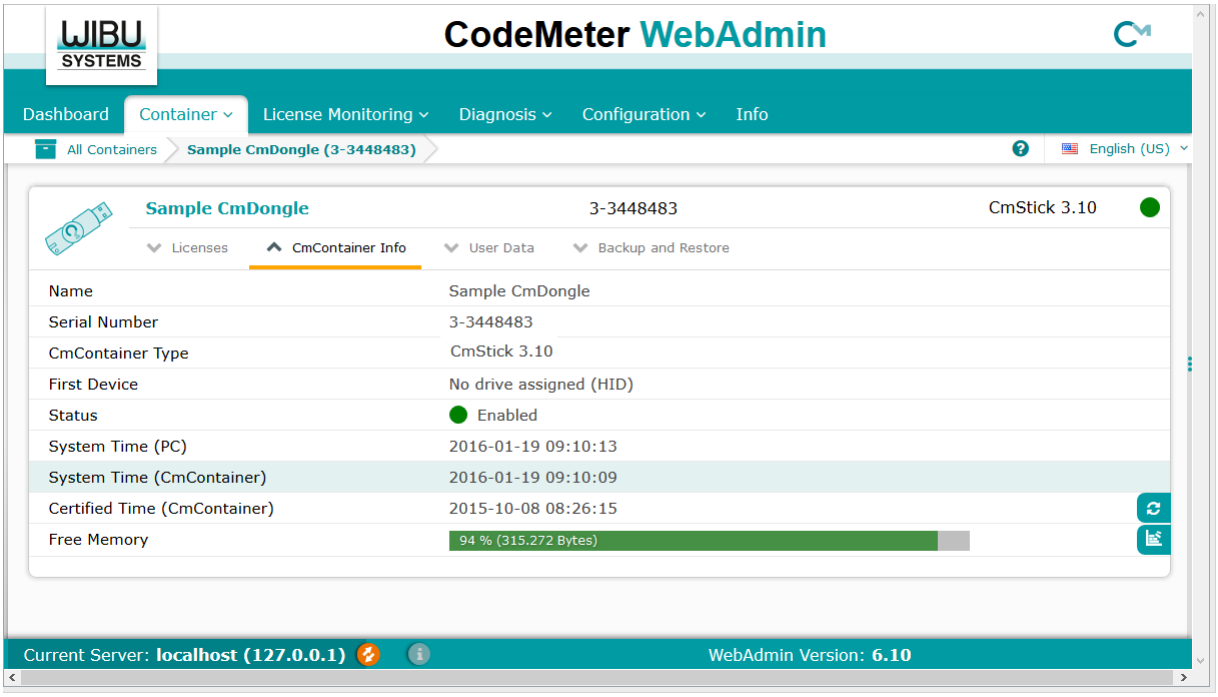
   No drive is assigned.



12.10. Set from HID to Mass Storage

To switch the USB communication standard from Human Interface Device (HID) to Mass Storage Device (MSD), please proceed as follows:

1. View the status in *CodeMeter WebAdmin* on page "**Content | CmContainer**".
   A drive is not assigned.



2. Call *cmu* 452
   For Windows OS call *cmu* call *cmu* by the start menu item **"Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt"** (Press "Windows" key to open Start screen | Type "CodeMeter Start Center" | Press "Enter" key). For the operating systems macOS and Linux this command is provided by the usual search path parameter.
3. Enter the following commandline:

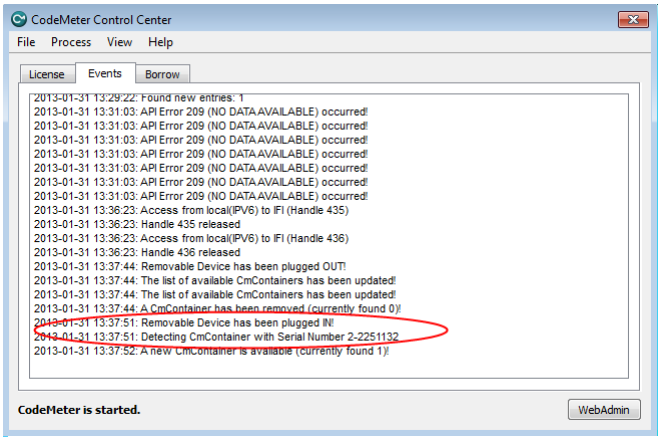
```
C:\Users\fs>cmu32 /s [Box mask-Serial number] --set-config-disk MsdCommunication
```

The current status displays in the following commandline output:

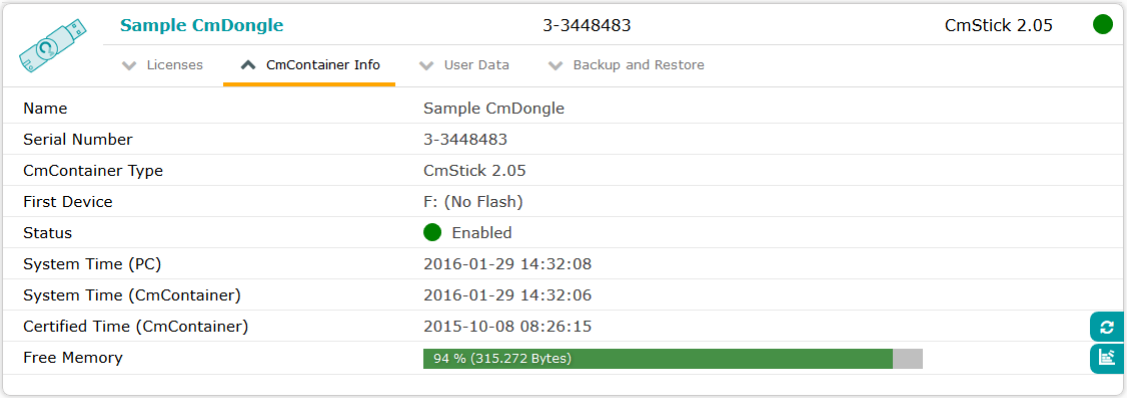
```
cmu32 - CodeMeter Universal Support Tool.
Version 5.00 of 2013-Jan-30 (Build 1039) for Win32
Copyright (C) 2007-2013 by WIBU-SYSTEMS AG. All rights reserved.

- CmStick/C with Serial Number 2-2251132 and version 2.01
```

```
Version:            2.01
Flash Size:         no real flash available
Virtual Drive:      No drive assigned (HID)
Communication:      Human Interface Device (HID)

Please replug your CmDongle to apply the changes.
```

4. Unplug and replug the *CmDongle*.
5. View logging in *CodeMeter Control Center* tab "**Events**".
   The information for the switch to MSD displays.



5. Check in *CodeMeter WebAdmin* page "**Content | CmContainer**".
   A drive is assigned and no flash memory available.



12.10. Linux Kernel Settings

CodeMeter requires a working USB hotplugging infrastructure for the detection of a *CmDongle*. Thus make sure that the hotplug is working properly.

If hotplugging seems not to work on a new Linux distribution, please check if the new "udev" mechanism is used. If "udev" is used, the problem can be fixed by copying the file `/usr/share/CodeMeter/52-codemeter.rules` into the directory `/etc/udev/rules.d/`.

For the communication between *CodeMeterLin* and the *CmDongle*, *CodeMeterLin* requires a Linux kernel (Kernel 2.4.x or Kernel 2.6.x) with support for USB mass storage, human interface (HID) and scsi-generic devices. If you use a self compiled Linux Kernel, please check if you have activated usb-storage and sg-device support (as modules).

General Kernel configuration display settings including USB mass storage or HID (Human Interface Device):

```
CmDongle via USB as MSD (CmStick, CmStick/M, CmCard):
CONFIG_SYSFS
CONFIG_USB_SUPPORT
CONFIG_USB_*_HCD*       (respective Host Controller)
CONFIG_USB_STORAGE
CONFIG_BLK_DEV_SD
CONFIG_BLK_DEV_SG       (for passthrough communication)
CONFIG_*_FS             (for file I/O, the respective file system)

CmStick via USB as HID:
CONFIG_SYSFS
CONFIG_USB_SUPPORT
CONFIG_USB_*_HCD*       (respective Host Controller)
CONFIG_HID
CONFIG_HIDRAW
CONFIG_USB_HID
```

```
CmCard via internal card reader (e.g. SDHCI):
CONFIG_SYSFS
CONFIG_MMC
CONFIG_MMC_BLOCK
CONFIG_MMC_*            (for the respective card reader)
CONFIG_BLK_DEV_SG       (for passthrough communication)
CONFIG_*_FS             (for file I/O, the respective file system)

CmCard via ATA:
CONFIG_SYSFS
CONFIG_BLK_DEV_SD
CONFIG_ATA              +  respective ATA Host adapter/controller
CONFIG_BLK_DEV_SG       (for passthrough communication)
CONFIG_*_FS             (for file I/O, the respective file system)
```

If your Linux Kernel is configured properly, use the command:

```
codemeter-info -L
```

to get a list of all *CmDongles* connected to your system. If this is not the case, you have probably a general problem with your Linux Kernel or a Kernel module could not be loaded correctly.

13 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. As tool of *CodeMeter Protection Suite* 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 Certificate Vault	works as a PKCS#11 compliant token provider, integrating with the Microsoft Cryptographic API Next Generation (CNG) as a Key Storage Provider (KSP), and working with OpenSSL API e.g. to keep and use the keys for TLS certificates. It is fully integrated with many essential applications including browsers, VPNs, and email.
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 Keyring	Protection solution based on user and password credentials created in the *Password Manager* component and supplied by the *Password Provider* component. *Password Manager* connects to *CodeMeter License Central* after startup and retrieves the configured users and passwords from it. There is no local data storage. Created passwords and users are used in *Password Provider*, e.g. for implementing the feature of protecting the source code.
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 Protection Suite	Toolbox for the automatic encryption of applications and libraries. The individual tools have been tailored specifically to work with each platform or environment (see *AxProtector*, *IxProtector*).
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. `*.WibuCmRaC` file (license request file). Based on this license request file use the *CodeMeter* programming tools to create an update file (`*.WibuCmRaU`) (license update). Subsequently, this file is safely transferred into the *CmContainer*. In addition, on creating the `*.WibuCmRaU` file automatically also a `*.WibuCmRaM` 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*

Term	Description
HIP	*High Level Programming API* see *Programming API*
IFI	see *Implicit Firm Item*
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 as tool of *CodeMeter Protection Suite*. '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.
CodeMeter 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.
Soft license	see *CmActLicense*.
Wibu Universal Protection Interface	see *Software Protection API*
WUPI	see *Wibu Universal Protection Interface*

14 Copyright information of software licenses used

Flot

Project	Runtime component CodeMeter WebAdmin
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
Version	v3.4.1 (jQuery)
Operating system	Windows, macOS, Linux

```
Copyright (c) 2010-2017, Christian Johansen, christian@cjohansen.no
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 Christian Johansen nor the names of his 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.
```

JQuery-ui

Project	Runtime component CodeMeter WebAdmin
Version	v1.12.1 (jQuery-ui)
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,
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
Version	1.2.1.0219
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.

"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:

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.

END OF TERMS AND CONDITIONS

APPENDIX: 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.
```

golang.org (/x/sys / and /x/crypto)

Project	Runtime component CodeMeter WebAdmin
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.
```

libcurl

Project	Runtime component CodeMeter License Server
Version	7.28.0
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 (native), 3.7.1 (java)	2.6.1 (native), 3.7.1 (java)
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.
```

ASN.1 Compiler

Project	Runtime component CodeMeter License Server	SDK component Programming API (HIP)
Version	1.0 starting with CodeMeter 6.0	1.0 starting with CodeMeter 6.0
Operating system	Windows, macOS, Linux	Windows, macOS, Linux

Bootstrapper dotnetInstaller

Project	Runtime component dotnetInstaller	SDK component dotnetInstaller
Version	2.3	2.3
Operating system	Windows	Windows

LLVM compiler and toolchain technology

Project	SDK component AxProtector
Version	3.8.0
Operating system	Windows, macOS, Linux

http://llvm.org/releases/3.8.0/LICENSE.TXT

```
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	7.1
Operating system	Windows, macOS, Linux

http://asm.ow2.io/license.html

```
Copyright (c) 2000-2011 INRIA, France Telecom
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.

"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 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.

      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.
```

CoreFX

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.
```

XStream

Project	SDK component AxProtector Java
Version	1.4.11.1 (http://x-stream.github.io/)
Operating system	Windows, macOS, Linux

```
Copyright (c) 2003-2006, Joe Walnes
Copyright (c) 2006-2015 XStream Committers
```

Public Suffix List

Project	Runtime component CodeMeter Control Center
Version	https://publicsuffix.org/
Operating System	Windows, macOS, Linux

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.

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 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 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.

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
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

Text Codec: GBK

Project	Runtime component CodeMeter Control Center
Origin	http://doc.qt.io/qt-5/qtcore-attribution-qbkcodec.html
Operating System	Windows, macOS, Linux

```
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
Origin	https://doc.qt.io/qt-5/qtcore-attribution-pcre2.html
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
```

Parts of QTemporaryFile

Project	Runtime component CodeMeter Control Center
Origin	https://doc.qt.io/qt-5/qtemporaryfile.html
Operating System	Windows, macOS, Linux

```
   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.

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
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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-cocoa-platform-plugin.html
Operating System	macOS

IAccessible2 IDL Specification

Project	Runtime component CodeMeter Control Center
Origin	http://doc.qt.io/qt-5/qtgui-attribution-iaccessible2.html
Operating System	macOS

```
 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
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
    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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype.html
Operating System	Windows, macOS, Linux

```
The FreeType License (FTL) is used.
```

```
                    The FreeType Project LICENSE
                    ----------------------------
                            2006-Jan-27
                    Copyright 1996-2002, 2006 by
          David Turner, Robert Wilhelm, and Werner Lemberg
Introduction
============
  The FreeType  Project is distributed in  several archive packages;
  some of them may contain, in addition to the FreeType font engine,
  various tools and  contributions which rely on, or  relate to, the
  FreeType Project.
  This  license applies  to all  files found  in such  packages, and
  which do not  fall under their own explicit  license.  The license
  affects  thus  the  FreeType   font  engine,  the  test  programs,
  documentation and makefiles, at the very least.
  This  license   was  inspired  by  the  BSD,   Artistic,  and  IJG
  (Independent JPEG  Group) licenses, which  all encourage inclusion
  and  use of  free  software in  commercial  and freeware  products
  alike.  As a consequence, its main points are that:
    o We don't promise that this software works. However, we will be
      interested in any kind of bug reports. (`as is' distribution)
    o You can  use this software for whatever you  want, in parts or
      full form, without having to pay us. (`royalty-free' usage)
    o You may not pretend that  you wrote this software.  If you use
      it, or  only parts of it,  in a program,  you must acknowledge
      somewhere  in  your  documentation  that  you  have  used  the
      FreeType code. (`credits')
  We  specifically  permit  and  encourage  the  inclusion  of  this
  software, with  or without modifications,  in commercial products.
  We  disclaim  all warranties  covering  The  FreeType Project  and
  assume no liability related to The FreeType Project.
  Finally,  many  people  asked  us  for  a  preferred  form  for  a
  credit/disclaimer to use in compliance with this license.  We thus
  encourage you to use the following text:
   """
    Portions of this software are copyright © <year> The FreeType
    Project (www.freetype.org).  All rights reserved.
   """
  Please replace <year> with the value from the FreeType version you
  actually use.
Legal Terms
===========
0. Definitions
--------------
  Throughout this license,  the terms `package', `FreeType Project',
  and  `FreeType  archive' refer  to  the  set  of files  originally
  distributed  by the  authors  (David Turner,  Robert Wilhelm,  and
  Werner Lemberg) as the `FreeType Project', be they named as alpha,
  beta or final release.
  `You' refers to  the licensee, or person using  the project, where
  `using' is a generic term including compiling the project's source
  code as  well as linking it  to form a  `program' or `executable'.
  This  program is  referred to  as  `a program  using the  FreeType
  engine'.
  This  license applies  to all  files distributed  in  the original
  FreeType  Project,  including  all  source  code,  binaries  and
  documentation,  unless  otherwise  stated  in  the  file  in  its
  original, unmodified form as  distributed in the original archive.
  If you are  unsure whether or not a particular  file is covered by
  this license, you must contact us to verify this.
  The FreeType  Project is copyright (C) 1996-2000  by David Turner,
  Robert Wilhelm, and Werner Lemberg.  All rights reserved except as
  specified below.
1. No Warranty
--------------
  THE FREETYPE PROJECT  IS PROVIDED `AS IS' WITHOUT  WARRANTY OF ANY
  KIND, EITHER  EXPRESS OR IMPLIED,  INCLUDING, BUT NOT  LIMITED TO,
  WARRANTIES  OF  MERCHANTABILITY   AND  FITNESS  FOR  A  PARTICULAR
  PURPOSE.  IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS
  BE LIABLE  FOR ANY DAMAGES CAUSED  BY THE USE OR  THE INABILITY TO
  USE, OF THE FREETYPE PROJECT.
2. Redistribution
-----------------
  This  license  grants  a  worldwide, royalty-free,  perpetual  and
  irrevocable right  and license to use,  execute, perform, compile,
  display,  copy,  create  derivative  works  of,  distribute  and
  sublicense the  FreeType Project (in  both source and  object code
  forms)  and  derivative works  thereof  for  any  purpose; and  to
  authorize others  to exercise  some or all  of the  rights granted
  herein, subject to the following conditions:
    o Redistribution of  source code  must retain this  license file
      (`FTL.TXT') unaltered; any  additions, deletions or changes to
      the original  files must be clearly  indicated in accompanying
      documentation.  The  copyright  notices  of  the  unaltered,
      original  files must  be  preserved in  all  copies of  source
      files.
```

```
    o Redistribution in binary form must provide a  disclaimer  that
      states  that  the software is based in part of the work of the
      FreeType Team,  in  the  distribution  documentation.  We also
      encourage you to put an URL to the FreeType web page  in  your
      documentation, though this isn't mandatory.
  These conditions  apply to any  software derived from or  based on
  the FreeType Project,  not just the unmodified files.   If you use
  our work, you  must acknowledge us.  However, no  fee need be paid
  to us.
3. Advertising
--------------
  Neither the  FreeType authors and  contributors nor you  shall use
  the name of the  other for commercial, advertising, or promotional
  purposes without specific prior written permission.
  We suggest,  but do not require, that  you use one or  more of the
  following phrases to refer  to this software in your documentation
  or advertising  materials: `FreeType Project',  `FreeType Engine',
  `FreeType library', or `FreeType Distribution'.
  As  you have  not signed  this license,  you are  not  required to
  accept  it.   However,  as  the FreeType  Project  is  copyrighted
  material, only  this license, or  another one contracted  with the
  authors, grants you  the right to use, distribute,  and modify it.
  Therefore,  by  using,  distributing,  or modifying  the  FreeType
  Project, you indicate that you understand and accept all the terms
  of this license.
4. Contacts
-----------
  There are two mailing lists related to FreeType:
    o freetype@nongnu.org
      Discusses general use and applications of FreeType, as well as
      future and  wanted additions to the  library and distribution.
      If  you are looking  for support,  start in  this list  if you
      haven't found anything to help you in the documentation.
    o freetype-devel@nongnu.org
      Discusses bugs,  as well  as engine internals,  design issues,
      specific licenses, porting, etc.
  Our home page can be found at
    https://www.freetype.org
```

FreeType 2

Project	Runtime component CodeMeter Control Center
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype.html
Operating System	Windows, macOS, Linux

The FreeType License (FTL) is used.

```
                    The FreeType Project LICENSE
                    ----------------------------
                            2006-Jan-27
                    Copyright 1996-2002, 2006 by
          David Turner, Robert Wilhelm, and Werner Lemberg
Introduction
============
  The FreeType  Project is distributed in  several archive packages;
  some of them may contain, in addition to the FreeType font engine,
  various tools and  contributions which rely on, or  relate to, the
  FreeType Project.
  This  license applies  to all  files found  in such  packages, and
  which do not  fall under their own explicit  license.  The license
  affects  thus  the  FreeType   font  engine,  the  test  programs,
  documentation and makefiles, at the very least.
  This  license   was  inspired  by  the  BSD,   Artistic,  and  IJG
  (Independent JPEG  Group) licenses, which  all encourage inclusion
  and  use of  free  software in  commercial  and freeware  products
  alike.  As a consequence, its main points are that:
    o We don't promise that this software works. However, we will be
      interested in any kind of bug reports. (`as is' distribution)
    o You can  use this software for whatever you  want, in parts or
      full form, without having to pay us. (`royalty-free' usage)
    o You may not pretend that  you wrote this software.  If you use
      it, or  only parts of it,  in a program,  you must acknowledge
      somewhere  in  your  documentation  that  you  have  used  the
      FreeType code. (`credits')
  We  specifically  permit  and  encourage  the  inclusion  of  this
  software, with  or without modifications,  in commercial products.
  We  disclaim  all warranties  covering  The  FreeType Project  and
  assume no liability related to The FreeType Project.
  Finally,  many  people  asked  us  for  a  preferred  form  for  a
  credit/disclaimer to use in compliance with this license.  We thus
  encourage you to use the following text:
   """
    Portions of this software are copyright © <year> The FreeType
    Project (www.freetype.org).  All rights reserved.
   """
  Please replace <year> with the value from the FreeType version you
  actually use.
Legal Terms
===========
0. Definitions
--------------
  Throughout this license,  the terms `package', `FreeType Project',
  and  `FreeType  archive' refer  to  the  set  of files  originally
  distributed  by the  authors  (David Turner,  Robert Wilhelm,  and
  Werner Lemberg) as the `FreeType Project', be they named as alpha,
  beta or final release.
  `You' refers to  the licensee, or person using  the project, where
  `using' is a generic term including compiling the project's source
  code as  well as linking it  to form a  `program' or `executable'.
  This  program is  referred to  as  `a program  using the  FreeType
  engine'.
  This  license applies  to all  files distributed  in  the original
  FreeType  Project,  including  all  source  code,  binaries  and
  documentation,  unless  otherwise  stated  in  the  file  in  its
  original, unmodified form as  distributed in the original archive.
  If you are  unsure whether or not a particular  file is covered by
  this license, you must contact us to verify this.
  The FreeType  Project is copyright (C) 1996-2000  by David Turner,
  Robert Wilhelm, and Werner Lemberg.  All rights reserved except as
  specified below.
1. No Warranty
--------------
  THE FREETYPE PROJECT  IS PROVIDED `AS IS' WITHOUT  WARRANTY OF ANY
  KIND, EITHER  EXPRESS OR IMPLIED,  INCLUDING, BUT NOT  LIMITED TO,
  WARRANTIES  OF  MERCHANTABILITY   AND  FITNESS  FOR  A  PARTICULAR
  PURPOSE.  IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS
  BE LIABLE  FOR ANY DAMAGES CAUSED  BY THE USE OR  THE INABILITY TO
  USE, OF THE FREETYPE PROJECT.
2. Redistribution
-----------------
  This  license  grants  a  worldwide, royalty-free,  perpetual  and
  irrevocable right  and license to use,  execute, perform, compile,
  display,  copy,  create  derivative  works  of,  distribute  and
  sublicense the  FreeType Project (in  both source and  object code
  forms)  and  derivative works  thereof  for  any  purpose; and  to
  authorize others  to exercise  some or all  of the  rights granted
  herein, subject to the following conditions:
    o Redistribution of  source code  must retain this  license file
      (`FTL.TXT') unaltered; any  additions, deletions or changes to
      the original  files must be clearly  indicated in accompanying
      documentation.  The  copyright  notices  of  the  unaltered,
      original  files must  be  preserved in  all  copies of  source
      files.
```

```
    o Redistribution in binary form must provide a  disclaimer  that
      states  that  the software is based in part of the work of the
      FreeType Team,  in  the  distribution  documentation.  We also
      encourage you to put an URL to the FreeType web page  in  your
      documentation, though this isn't mandatory.
  These conditions  apply to any  software derived from or  based on
  the FreeType Project,  not just the unmodified files.   If you use
  our work, you  must acknowledge us.  However, no  fee need be paid
  to us.
3. Advertising
--------------
  Neither the  FreeType authors and  contributors nor you  shall use
  the name of the  other for commercial, advertising, or promotional
  purposes without specific prior written permission.
  We suggest,  but do not require, that  you use one or  more of the
  following phrases to refer  to this software in your documentation
  or advertising  materials: `FreeType Project',  `FreeType Engine',
  `FreeType library', or `FreeType Distribution'.
  As  you have  not signed  this license,  you are  not  required to
  accept  it.   However,  as  the FreeType  Project  is  copyrighted
  material, only  this license, or  another one contracted  with the
  authors, grants you  the right to use, distribute,  and modify it.
  Therefore,  by  using,  distributing,  or modifying  the  FreeType
  Project, you indicate that you understand and accept all the terms
  of this license.
4. Contacts
-----------
  There are two mailing lists related to FreeType:
    o freetype@nongnu.org
      Discusses general use and applications of FreeType, as well as
      future and  wanted additions to the  library and distribution.
      If  you are looking  for support,  start in  this list  if you
      haven't found anything to help you in the documentation.
    o freetype-devel@nongnu.org
      Discusses bugs,  as well  as engine internals,  design issues,
      specific licenses, porting, etc.
  Our home page can be found at
    https://www.freetype.org
```

LibJPEG-turbo

Project	Runtime component CodeMeter Control Center
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:

- 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
Origin	https://doc.qt.io/qt-5/macos.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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-bdfbcf.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 "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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-bcf.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
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
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
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 © 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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-harfbuzz-ng.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.
```

OpenGL ES 2 Headers

Project	Runtime component CodeMeter Control Center
Origin	http://doc.qt.io/qt-5/qtgui-attribution-opengl-es2-headers.html

Operating System	Windows, macOS, Linux

OpenGL Headers

Project	Runtime component CodeMeter Control Center
Origin	http://doc.qt.io/qt-5/qtgui-attribution-opengl-headers.html
Operating System	Windows, macOS, Linux

XCB

Project	Runtime component CodeMeter Control Center

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.
```

xkbcommon

Project	Runtime component CodeMeter Control Center
Origin	https://doc.qt.io/qt-5.9/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:

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.

END OF SOFTWARE COPYRIGHT/LICENSE STATEMENTS

BEGINNING OF LICENSE STATEMENTS FOR UNDISTRIBUTED DATA FILES IN test/data,
derived from xkeyboard-config:

```
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
Origin	http://doc.qt.io/qt-5/qtgui-attribution-freetype-zlib.html
Version	1.2.1
Operating System	Windows, macOS, Linux

```
Version 1.2.11, January 15th, 2017
Copyright (C) 1995-2017 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
   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

The data format used by the zlib library is described by RFCs (Request for
Comments) 1950 to 1952 in the files http://tools.ietf.org/html/rfc1950
(zlib format), rfc1951 (deflate format) and rfc1952 (gzip format).
```

libsodium

Project	Runtime component CmCloud Client (CodeMeter License Server)
Version	1.0.13 starting with CodeMeter 6.90
Operating system	Windows, macOS, Linux

https://github.com/jedisct1/libsodium

```
 * ISC License
 *
 * Copyright (c) 2013-2017
 * Frank Denis <j at pureftpd dot org>
 *
 * Permission to use, copy, modify, and/or 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" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
 * WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
 * MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
 * ANY SPECIAL, DIRECT, 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.
```

Index

- * -

*.wbc and/or *.xml files
editing 286

*.WibuCmLIF (license infromation file) 22

- . -

.NET
WibuCmNet-Bibliothek 51

- A -

Access Password
Product Item Option 36

Activation Time
Product Item Option 30

add automatic server search
CodeMeter WebAdmin 418

Allocation order of licenses 39

Apache Commons
Open Source 475

ApiCommunicationMode
WebAdmin 401, 435

ASM Java Program Library
Open Source 475

ASN.1 Compiler
Open Source 473

AxProtector
IP Protection 287
Translocated Execution (OOPE) 286

AxProtector .NET
.NET Options 96
Activate Hardware Locking 94
Activate IxProtector 97
Activate Plug-out Check 90
Activate Runtime Check 90
Activation Time 92
Add code integrity check 95
Advanced Options 97
Also at Runtime Check 90
Automatic Trap Generation 94
Basic Debugger Check 94
Check Certified Time 92
CmContainer / PC System Time check 93
Create Logfile 97
Create mobile application 93
Customized Error Messages 96
Decrement by 90
Default Error Messages 95
Destination File 84
Encryption Time check 92
Error Messages 95, 96
Exclusive Mode 86, 88
Expiration Time 91
Extended Commandline Options 97
Feature Code 86, 88
File To Protect 84
Firm Code 85, 88
Ignore Linger Time 87, 89
Inline Messages (.NET only) 96
IxProtector Bytes 103
IxProtector Methods 103
IxProtector Name 103
IxProtector Views 102
License access lock (Configuration) 94
License Handling 86, 88, 89
License List Feature Code 100
License List Firm Code 99
License List Product Code 100
License List Subsystem 100
License List Description 99, 122
License List Id 98, 121
License List Ignore Linger Time 101
License List License Options 100
License List Licensing Systems 99
License List Minimum Driver Version 100
License List Minimum Firmware 101
License List Release Date 101
License List WupiReadData 101
License List WupiWriteData 101
License Option 86, 88, 89
Licensing Systems 85, 86, 87, 88, 89
Logging 97
Maintenance Period 92
Max. Allowed Ignores 90
Maximum Certified Time Age 92
Minimum driver 86, 89
Minimum Firmware 87, 89
Minutes to be allowed older 93
Minutes to be allowed younger 93
No Strong Name 96
No User Limit 86, 89
Normal User Limit 86, 88
Obfuscation 93, 274
Optimization 97
Period 90
Period without time checking 92
Product Code 85, 88
Reflector defence 94
Release Date 87, 89
Resource encryption 95
Runtime Settings 90, 91, 114
Runtime Settings, advanced 91, 92, 93
Security Options 94, 95
Set Certified Time 92
Source File 84
Station Share 86, 88
Strong Name from Container 96
Strong Name from File 96
Subsystem Local 86, 88
Subsystem Network 86, 88
Summary Back 104
Summary Finish 104
Summary Protect now 104
Terminate host application 93
Thresholds Expiration Time 91, 114
Thresholds Unit Counter 91, 114
Unit Counter 90, 91
User Defined Text 91
User Message DLL 95
WibuKey Compatibility Mode 86, 88
WupiReadData 89
WupiWriteData 89

AxProtector .NET Standard
.NET Options 119
Activate Hardware Locking 117
Activate IxProtector 120
Activate Plug-out Check 113
Activate Runtime Check 113
Activation Time 115
Add code integrity check 118
Advanced Options 120
Also at Runtime Check 113
Automatic Trap Generation 117
Basic Debugger Check 117
Check Certified Time 115
CmContainer / PC System Time check 116
Create Logfile 120
Create mobile application 116
Customized Error Messages 119
Decrement by 113
Default Error Messages 118
Destination File 107
Encryption Time check 115
Error Messages 118, 119
Exclusive Mode 109, 111
Expiration Time 114
Extended Commandline Options 120

AxProtector .NET Standard
Feature Code 109, 111
File To Protect 107
Firm Code 108, 111
Ignore Linger Time 110, 112
Inline Messages (.NET only) 119
IxProtector Bytes 126
IxProtector Methods 126
IxProtector Name 126
IxProtector Views 125
License access lock (Configuration) 117
License Handling 109, 111
License List Feature Code 123
License List Firm Code 122
License List Product Code 123
License List Subsystem 123
License List Ignore Linger Time 124
License List License Options 123
License List Licensing Systems 122
License List Minimum Driver Version 123
License List Minimum Firmware 124
License List Release Date 124
License List WupiReadData 124
License List WupiWriteData 124
License Option 109, 111
Licensing Systems 108, 109, 110, 111, 112
Logging 120
Maintenance Period 115
Max. Allowed Ignores 113
Maximum Certified Time Age 115
Minimum driver 109, 111
Minimum Firmware 110, 112
Minutes to be allowed older 116
Minutes to be allowed younger 116
No Strong Name 119
No User Limit 109, 111
Normal User Limit 109, 111
Obfuscation 116
Optimization 120
Period 113
Period without time checking 115
Product Code 108, 111
Reflector defence 117
Release Date 110, 112
Resource encryption 118
Runtime Settings 113, 114
Runtime Settings, advanced 114, 115, 116
Security Options 117, 118
Set Certified Time 115
Source File 107
Station Share 109, 111
Strong Name from Container 119
Strong Name from File 119
Subsystem Local 109, 111
Subsystem Network 109, 111
Summary Back 127
Summary Finish 127
Summary Protect now 127
Terminate host application 116
Unit Counter 113, 114
User Defined Text 114
User Message DLL 118
WibuKey Compatibility Mode 109, 111
WupiReadData 112
WupiWriteData 112
AxProtector Commandline
-! (Create *.wbc file) 285
-# (Logging) 285
-? -h (Options and help) 285
-@cmds.wbc (Parameter in executable file) 285
-A[AES] (Encryption algorithm) 263
-ANF (Message if assembly not found, .NET) 279
AutoCAD2011 - ARX files (pliug-in) 271
-CAA (Security options) 266, 267
-CACT (CmContainer SystemTime) 267
-CAD (File Access Mode) 267
-CAE (Plug-Out) 267
-CAG (Anti-Debugging .NET) 269
-CAG (Anti-Debugging Java) 269
-CAG (Anti-Debugging) 267, 268
-CAG (Anti-Debugging, MAC OS X) 268
-CAL (areas of encryption) 270
-CAM (System Menu) 270
-CAR (Runtime Check) 270
-CAS (Percentage Encrypted) 270
-CAT (Certified and System Time) 270
-CAV Code Integrity Check - .NET 271
-CAV (Code Integrity Check) 270, 271
-CAV (Code Integrity Check) - Java 270, 271
-CAZ (Encryption Time) 271
-CCA (Target platform and option -CCX) 271
-CCB (.reloc Section translation) 271
-CCC (ActiveX / OCX Images) 272
-CCD (Shared Objects options) 272
-CCE (PE is not enlarged) 271
-CCF (Extended protection / special handling) 271
-CCH (Preventing global hooking) 271
-CCI (Load sequence) 271, 272
-CCK (DLL unload Windows XP) 272
-CCQ (Clear licenses on Exit Process) 272
-CCR (Deactivation renaming of Sections) 272
-CCS (Licenses from first connected CmContainer) 272
-CCT (Specification architecture AxEngine) 272
-CCX (Mixed Mode Assemblies) 272
-CDC (File name extension for file encrypting) 272
-CDH (Access to license file encryption) 273
-CDK (Licensing system file encryption) 273
-CFx (Feature Code) 264
-CI (IxProtector) 274
-CIH (WUPI and Hooking) 273
-CIN (No error messages IxProtector) 273
-CIP (IxProtector) 273
-CK (RID key cache, .NET) 274
-CMD(n) (Reencryption of methods, .NET) 275
-CML (Min. size methods, .Java) 282
-CML (Min. size methods, .NET) 275
-CO (Obfuscation) 274, 275
-CP (Cleanup mechanism Windows) 275
-CPA (Encrypting property accessors, .NET) 275
-D (Driver version) 264, 265
-EA (Activation Time check) 275
-EC (MSIL Code class construction, .NET) 275
-EE (Expiration Time check) 275, 276
-EF (Decrement Firm Access Counter) 276
-EM (Maintenance Period required) 276
-ET (Enforce Certified Time update) 276
-EU (Unit Counter check) 276
-EXTRACT (Assembly printout, .NET) 285
-FW (Firmware Version) 265
-FW (minim. Firmware on encryption) 276
-Fx (Firm Code) 264
-G (Exclude areas fropm encryption) 276
-I (Exception handling Plugin DLLs) 277
-j:gs- Deactivate Getter Setter methods 283
-ja (Main Class arguments runtimet Java) 282
-jb (Exception handling Java) 282
-jcl (Class Loader) 282
-jd (Min.- Max. Version Java) 282, 283
-jff:[c|w] (external Class files Java) 283
-jfx (JavaFX initialize) 283
-jh (Hide and rename of classes Java) 283
-jip IP Protection 283
-jl (White/ blacklist classes Java) 284
-jm (Starting Main Class Java) 284
-jn (Class loading Java) 284
-jo (Output options *.jar Java) 284
-jom (Output options modular.jar Java) 284
-jpc (define additional JAR files Java) 282

AxProtector Commandline
-jx (Exit application Java) 284, 285
-K (Licensing System) 263
-KIP - IP Protection 264
-L (Language message texts) 277
-M (Outout texts for message texts) 277
-N (Network access mode) 265
-NNI Ignore Linger Time 265
-O (Name and path of encrypted target file) 279, 280
-prio (set process priority, Windows only) 279
-PROBING (Path information when assembly found, .NET) 279
-Px Product Code) 264
-RD (Release Date) 264
-RID (Numberl RID variants) 276
-RIDI (Number of RID and Trap variants IxProtector) 276
-S (Search order for licenses) 265
-SIG (Signature Private Key Certificate) 266
-SNK (Strong Name Key, .NET) 279
-SW (Search order for licenses, WAN) 265
Syntax 263
-TRAP (Hacker traps, .NET. Java) 279
-U (Calling message DLL) 278
-UDT (user defined text) 279
-UI (Inline message assembly, .NET) 278, 279
-UM (Call message assembly, .NET) 279
-UN (switch off error messages) 279
-V (Verbose mode) 285
-WC (Threshold Certified Time) 277
-WE (Threshold Expiration Time) 277
-WP (Threshold Usage Period) 277
-WU (Threshold Unit Counter) 277
-X (Static linking) 263
-XC AxProtector Linux using CmEmbedded (CmDongle) 263
-XCA AxProtector Linux using CmEmbedded (CmActLicense) 263

AxProtector File Encryption
Advanced Options 254
Create Logfile 254
Destination File 248
Exclusive Mode 250, 252
Extended Commandline Options 254
Feature Code 249, 251
File Encryption Extension 260
File Encryption File Access Mode 260
File Encryption Name 260
File Encryption Player Check 260
File Encryption Writing existing file 261
File Encryption Writing New File 261
File To Protect 248
Firm Code 249, 251
Ignore Linger Time 250, 252
License Handling 249, 250, 251, 252
License List Licensing Systems 257
License List Description 256
License List Feature Code 257
License List Firm Code 256
License List Id 255
License List Ignore Linger Time 258
License List License Options 257
License List Licensing Systems 256
License List Minimum Firmware 258
License List Product Code 257
License List Release Date 258
License List Subsystem 257
License List WupiReadData 258
License List WupiWriteData 258
License Option 249, 250, 252
Licensing Systems 249, 250, 251, 252, 253
Logging 254
Minimum driver 250, 252
Minimum Firmware 250, 252
No User Limit 250, 252
Normal User Limit 249, 252
Product Code 249, 251
Release Date 250, 252
Source file 248
Station Share 249, 252
Subsystem Local 249, 251
Subsystem Network 249, 251
Summary Back 262
Summary Finish 262
Summary Protect Now 262
WibuKey Compatibility Mode 249, 252
WupiReadData 253
WupiWriteData 253

AxProtector Java
Activate license access lock 161
Activate obfuscation 161
Activate Runtime Check 157
Activation Time 159
Add folders and files to class path 164
Advanced Options 165
Anwendungsserver 280
Application Server 280
Automatic Trap Generation 161
Basic Debugger Check 161
Blacklist 163, 164
Callback Manipulation Check 160
Check Certified Time 159
Class Name 162
Classes to encrypt 164
CmContainer / PC System Time check 160
Create Logfile 165
create valid java class files 165
Customized Error Messages 162
Deactivate Getter / Setter generation 163
Default Error Messages 162
Destination File 151
Encrypt constant Pool entries 160
Encrypt method control flow (affects performance) 160
Encryption Time check 159
Error Messages 162
Exclusive Mode 153, 155
Expiration Time 158
Extended Commandline Options 165
Feature Code 153, 155
File To Protect 151
Firm Code 152, 154
Ignore detection of reflection 161
Ignore Linger Time 154, 156
Initialize JavaFX 163
Integrity verification 160
IxProtector 165, 280
IxProtector Bytes 171
IxProtector Methods 171
IxProtector Name 171
IxProtector Views 171
jar, war, ear, aar Archive 280
JVMPI / JVMTI Detection 160
License Handling 153, 155
License List Feature Code 168
License List Firm Code 167
License List Product Code 168
License List Subsystem 168
License List Description 167
License List Id 166
License List Ignore Linger Time 169
License List License Options 168
License List Licensing Systems 167
License List Minimum Driver Version 168
License List Minimum Firmware 169
License List Release Date 169
License List WupiReadData 169
License List WupiWriteData 169
License Option 153, 155
Licensing Systems 152, 153, 154, 155, 156
Logging 165
Logging.properties file 285
Maintenance Period 159

AxProtector Java
Max. Allowed Ignores 157
Maximum Certified Time Age 159
Method encryption 280
Minimum driver 153, 155
Minimum Firmware 154, 156
Minutes to be allowed older 160
Minutes to be allowed younger 160
No User Limit 153, 155
Normal User Limit 153, 155
Obfuscation 274, 275
Optimization 165
Options Call System.exit() 163
Options Java Runtime 163
Options main class 163
Options Minimum Java Version 163
Options Split Output 163
Output logging console at runtime 285
Parameter main class 163
Period 157
Period without time checking 159
Print name mapping 161
Product Code 153, 155
Release Date 154, 156
Rename encrypted classes 164
Runtime Settings 157, 158
Runtime Settings, advanced 158, 159, 160
Security Options 160, 161
Set Certified Time 159
Source File 151
Station Share 153, 155
Subsystem Local 153, 155
Subsystem Network 153, 155
Summary Back 173
Summary Finish 173
Summary Protect Now 174
Threshold Unit Counter 158
Thresholds Expiration Time 158
Unit Counter 158
Unit Counter Also at Runtime Check 157
Unit Counter Decrement by 157
User Defined Text 158
User Message Class 162
User Messages 162
VM Verification 161
Web-Application 280
Whitelist 163, 164
WibuKey Compatibility Mode 153, 155
WupiReadData 156
WupiWriteData 156
AxProtector Linux
Activate Ixprotector / WUPI 188
Activate Runtime Check 181
Add code integrity check 186
Advanced Options 188
CmContainer / PC System Time check 184
Create Logfile 188
Customized Error Messages 187
Default Error Messages 187
Destination File 175
Encryption Time check 183
Error Messages 187
Exclusive Mode 177, 179
Extended Commandline Options 188
Feature Code 177, 179
File to protect 175
Firm Code 176, 178
Ignore Linger Time 178, 180
IxProtector Function Description 193
IxProtector Function Id 193
IxProtector Function Length 193
IxProtector Function License List 194
IxProtector Function Name 193
IxProtector Function Trap 194
License Handling 177, 179
License List Minimum Firmware 191
License List Release Date 191
License List Id 189
License List Ignore Linger Time 192
License List License Options 191
License List Subsystem 191
License List WupiReadData 192
License List WupiWriteData 192
License Option 177, 179
Licensing Systems 176, 177, 178, 179, 180
Link API statically to Application 187
Logging 188
Maintenance Period 183
Minimum driver 177, 179
Minimum Firmware 178, 180
Minutes to be allowed older 184
Minutes to be allowed younger 184
No User Limit 177, 179
Normal User Limit 177, 179
Period without time checking 183
Product Code 177, 179
Release Date 178, 180
Runtime Settings 181, 182
Runtime Settings, advanced 183, 184
Security Option, advanced 186, 187
Size of encrypted Code (in %) 187
Source file 175
Station Share 177, 179
Subsystem Local 177, 179
Subsystem Network 177, 179
Summary Back 195
Summary Finish 195
Summary Protect Now 195
Threshold Expiration Time 182
Threshold Unit Counter 182
User Defined Text 182
WibuKey Compatibility Mode 177, 179
WupiReadData 180
WupiWriteData 180
AxProtector macOS
Activate Ixprotector / WUPI 141
Activate license access lock 138, 184
Activate Plug-out Check 134, 181
Activate Runtime Check 134
Activation Time 136, 183
Add code integrity check 140
Advanced Debugger Check 138, 184
Advanced Options 141
Basic Debugger Check 138, 184
Check Certified Time 136, 183
CmContainer / PC System Time check 137
Create Logfile 141
Customized Error Messages 137
Default Error Messages 137
Destination File 128
Encryption Time check 136
Error Messages 137
Exclusive Mode 130, 132
Expiration Time 135, 182
Extended Commandline Options 141
Feature Code 130, 132
File to protect 128
Firm Code 129, 132
Ignore Linger Time 131, 133
IxProtector Function Description 147
IxProtector Function Id 147
IxProtector Function Length 147
IxProtector Function License List 147
IxProtector Function Name 147
IxProtector Function Trap 148
License access lock (Configuration) 138, 184
License Handling 130, 132
License List Description 143, 190

AxProtector macOS
License List Feature Code 144, 191
License List Firm Code 143, 190
License List Licensing Systems 143, 190
License List Minimum Driver Version 144, 191
License List Minimum Firmware 145
License List Product Code 144, 191
License List Release Date 145
License List Id 142
License List Ignore Linger Time 145
License List License Options 144
License List Subsystem 144
License List WupiReadData 145
License List WupiWriteData 145
License Option 130, 132
Licensing Systems 129, 130, 131, 132, 133
Link API statically to Application 140
Logging 141
Maintenance Period 136
Max. Allowed Ignores 134, 181
Maximum Certified Time Age 136, 183
Minimum driver 130, 132
Minimum Firmware 131, 133
Minutes to be allowed older 137
Minutes to be allowed younger 137
No User Limit 130, 132
Normal User Limit 130, 132
Period 134, 181
Period without time checking 136
Product Code 130, 132
Release Date 131, 133
Runtime Settings 134, 135, 136, 181, 183
Runtime Settings, advanced 135, 136, 137, 182, 183
Security Option, advanced 140
Security Options 138, 184
Set Certified Time 136, 183
Size of encrypted Code (in %) 140
Source file 128
Station Share 130, 132
Subsystem Local 130, 132
Subsystem Network 130, 132
Summary Back 149
Summary Finish 149
Summary Protect now 149
Threshold Expiration Time 135
Threshold Unit Counter 135
Unit Counter 135, 182
Unit Counter Also at Runtime Check 134, 181
Unit Counter Decrement by 134, 181
User Defined Text 135
Virtual Machine Detection 138, 184
WibuKey Compatibility Mode 130, 132
WupiReadData 133
WupiWriteData 133

AxProtector Windows
Activate Automatic File Encryption 72
Activate IxProtector 72
Activate runtime check 64
Activates license access lock 68
Activating plugout check (CmDongle) 64
Activation Time 66
Add code integrity check 69
Add control and about menu 67
Advanced Options 72
Also at runtime check 64
Basic Debugger Check 68
Check Cetrified Time 66
CmContainer / PC System Time check 67
Create Logfile 72
Create mobile application 67
Customized Error Messages 72
Decrement by 64
Default Error Messages 71
Destination file 58
Dynamic Code Modification 68
Dynamic loading of Wibu-Systems libraries 72
Encryption Time check 66
Error Message 71
Error Messages 71, 72
Exclusive Mode 60, 62
Expiration Time 65, 66
Extended Commandline Options 72
Extended Static Modification 67
Feature Code 60, 62
File Encryption Extension 80
File Encryption Player Check 80
File Encryption Writing Existing File 80
File Encryption Writing New File 80
File Encryption File Access Mode 80
File Encryption Name 80
File to protect 58
Firm Code 59, 61
Generic Debugger Detection 68
IDE Debugger Check 68
Ignore Linger Time 61, 63
Inline Messages (not available) 71
IxProtector Function Description 78
IxProtector Function Id 78
IxProtector Function Length 78
IxProtector Function License List 78
IxProtector Function Trap 78
Kernel Debugger Check 68
License access lock (Configuration) 68
License Handling 60, 62
License List Id 73
License List Description 73
License List Feature Code 75
License List Firm Code 74
License List Ignore Linger Time 76
License List License Options 75
License List Licensing Systems 74
License List Minimum Driver Version 75
License List Minimum Firmware 76
License List Product Code 74
License List Release Date 76
License List Subsystem 75
License List WupiReadData 76
License List WupiWriteData 76
License Option 60, 62
Licensing Systems 59, 60, 61, 62, 63
Link API statically to Application 70
Logging 72
Maintenance Period 66
Max. Allowed Ignores 64
Maximum Certified Time Age (hours) 66
Minimum driver 60, 62
Minimum Firmware 61, 63
Minutes to be allowed older 67
Minutes to be allowed younger 67
No User Limit 60, 62
Normal User Limit 60, 62
Period 64
Period without time checking (hours) 66
Product Code 59, 62
Release Date 61, 63
Resource Encryption 67
Runtime Settings 64, 65
Runtime Settings, advanced 65, 66, 67
Security Options 67, 68
Security Options, advanced 69, 70
Set Certified Time 66
Size of encrypted Code (in %) 70
Source file 58
Static Code Modification 67
Station Share 60, 62
Subsystem Local 60, 62
Subsystem Network 60, 62
Summary Back 82

AxProtector Windows
Summary Finish 81
Summary Protect Now 82
Suppress IxProtector Error Messages 71
Terminate host application 67
Threshold 65
Unit Counter 64, 65
User Defined Text 65
User Message DLL 71
Virtual Machine Detection 68
WibuKey Compatibility Mode 60, 62
WupiReadData 63
WupiWriteData 63

- B -

Backup of CmDongle 360
Binding Extension 22
Binding scheme 21
Bootstrapper dotnetInstaller
Open Source 473

- C -

Certificate signing
AxProtector Commandline 266
Certified Time 357
update 408
CmActLicense
Activating licenses 22, 368
additional activation options 22
Binding Extension 22
Binding scheme 21
'broken' license 389
CodeMeter SmartBind 21
None-Binding 22
Protection-Only license model 22
Smart Bind 329
SmartBind behaviour in VM 329
Trial License license model 22
CmActLicense License
Programming (CmBoxPgm) 331
CmBoxPgm 315
<public key file> 336
Access Password -papwd 319
Activation by Phone CmActLicense 330
Activation Time -pat 320
Activation Time, absolute -pata 320
Activation Time, relative -patr 320
Add / Update -cau 316
Add -ca 316
Allowed CmAct Ids -lacids 328
Backup File -bkp 335
Box Index -qb 317
Box Index Range -qnx 317
Box Passwort -pwd 317
Cleanup Registry -rcl 335
CmActLicense Activation Code -lac 328
CmActLicense Binding Value-lbind 329
CmActLicense in VM -lopt:ewffbwf 330
CmActLicense in VM -lopt:vm 330
CmActLicense License ID -lpid 331
CmActLicense License Information File (phone) -lip 330
CmActLicense License Information File -lif 330
CmActLicense License Options -lopt 330
CmActLicense -lopt:container 330
CmActLicense reimport -lopt:reimport 330
CmActLicense Target Operating System -los 330
CmCloud Creation of the CmCloud Key Certificate -clkc 335
Create RaC File -crac 335
Customer Owned License Information -pcoli 320
Delete -cd 317
Delete if possible -cdx 317
Display CmActLicense Binding Scheme -lfs 329
Display CmActLicense Installation ID -ldi 329
Display CmActLicense License File -ldf 329
EWF/FBWF 330
Expiration Time -pet 321
Expiration Time, absolute -peta 321
Expiration Time, relative -petr 321
Extended Protected Data -ped 320
Feature Map -pfm 321
File-based Activation CmActLicense 330
Firm Access Counter -fac 318
Firm Code -f 318
Firm Item License Transfer Permissions 318
Firm Item Text -ft 319
Firm Key -fk 335
Firm Precise Time, absolute -fpta 318
Firm Precise Time, relative -fptr 318
Firm Update Counter -fuc 319
FSB Entry -fsb 335
Help -? 336
Hiidden Data -phd 321
License Quantity -plq 322
License Quantity, absolute -plqa 322
License Quantity, relative -plqr 322
License Transfer 323
Licensé Transfer Depth -pltdepth 323
License Transfer LTK targets -pltltktarg 323
License Transfer Maximum Borrow Period -pltmbp 323
License Transfer Permission -pltperm 323
License Transfer Targets-plttarg 323
License Transfer Type -plttype 323
Linger Time -plt 323
List -l 317
Logging -log 336
Maintenance Period -pmp 324
Maintenance Period, Date -pmpd 324
Maintenance Period, Integer-pmpi 324
Maximum Encryption Rate -pmer 324
Merge Remote Activation Update File -mrau 318
Minimum required runtime (CmActLicense) 330
Minimum Runtime Version -pmrt 325
Module Item -pmi 324
Named User -pnu 325
Network License Counter -pnwc 325
Product Code -p 319
Protected Data -ppd 325
Recursive Removal -r 318
Remote Activation Update -rau 318
Secret Data -psd 325
Secure License Tracking 336
Serial number -qs 317
Smart Bind 329
Text -pt 326
Triple Mode Redundancy 322
Unit Counter -puc 326
Unit Counter, absolute -puca 326
Unit Counter, relative -pucr 326
Universal Data AES key storage -puvdaes 327
Universal Data Custom Data -puvddata 327
Universal Data Password -puvdpwd 328
Universal Data -puvd 327
Universal Data RSA key storage -puvdrsa 328
Universal Data User Permissions -puvdperm 327
Universal Firm Code License Programming Options -lpo 331
Universal Firm Code License Update File -laf 329
Update -cu 316
Usage Period -pup 326
Usage Period, absolute -pupa 327
Usage Period. relative -pupr 327
User Data -pud 326
Validation Mode -val 336
Validation of Signed Log Files -vslf 336
Verbose Mode -v 336
WUPI Data -pwupidata 328
CmCard
Detection on Linux 368
CmDongle

CmDongle
First connection 368
CmDust 450
CmFAS Assistant 394
CmFirm.wbc 14
CmLicense Editor 306
Display Window 309
Menu Bar 307
Output Window 309
Remote Programming 309
Structure and Navigation 307
Symbol Bar 308
Tree View 308
WibuCmRaC 309
WibuCmRaM 309
WibuCmRaU 309
Working with 309
CmStick /BMC
Detection on Linux 368
CmStick/T 359
cmu
CodeMeter Universal Support Tool 452
CmWAN
AxProtector (encrypting) 364
CmBoxPgm (programming) 362
CodeMeter API Guide (accessing) 362
CodeMeter WebAdmin (configuring) 365
profiling 365
registry, server.ini (configuring) 365
CodeMeter
Concept 26
Form factors 20
Installation 370
Operating Systems 24
Token 25
CodeMeter API Guide 300
Blocks Tab 301
Function Tab 301
Handle Display Window 302
Interactive Area 302
Menu Bar 301
Record Area 302
Source Code Area 302
Structure and Navigation 300
Tree View 302
WUPI Tab 301
CodeMeter Control Center 384
Activation invalid 389
Activation status 387
Borrowing Tab 392
Certified Time Update 388
CmDongle register 388
Event Tabs 391
Firmware Update 389
License import 387
License Tab 389
Logging, activate 387
Menu Bar 387
Start CodeMeter Service 388
Status and Open 393
Stucture and Navigation 386
CodeMeter Core API 297
Access API 298
Authentication API 298
Encryption API 298
Error Management API 299
Functional Areas 298
License Transfer API 300
Management API 299
Programming API 299
Remote Update API 299
Time Management API 300
CodeMeter Embedded 25
CodeMeter FAQ 369
CodeMeter License Central 337
Admin-Interface 341
Application Scenarios 341
Architecture 338
Connectors 339
Depot-Interface 341
Gateway 340
Principle 337
CodeMeter License Editor
Firm Code 310
Linger Time 313
PIO Activation Time 313
PIO Expiration Time 313
PIO Extented Protected Data 314
PIO Feature Map 313
PIO Hidden Data 313
PIO License Quantity 312
PIO Linger Time 313
PIO Maintenance Period 313
PIO Product Code 311
PIO Protected Data 314
PIO Secret Data 314
PIO Text 312
PIO Unit Counter 312
PIO Usage Period 312
PIO User Data 314
CodeMeter License Server 50
Run CmWAN Server (WebAdmin) 426
Run Network Server (WebAdmin) 425
Start after Installation 346
CodeMeter License Tracking 456
Access Entry 459
Administrative Entry 460
Borrow Access Entry 460
Borrow Return Entry 460
Configuration 457
Denial Entry 460
License Entry 459
List of Licenses Entry 459
Logfile Format 458
Profiling 457
Release Entry 460
Requirements 457
Secure License Tracking 456
Signature-Eintrag 461
SignedLogfile-Eintrag 460
CodeMeter on Embedded Systems
Embedded 25
CodeMeter Sample Applications
CmCalculator 304
CmDemo 303
WupiCalculator 304
CodeMeter service
Behavior at system startup 368
start (Linux) 385
start (macOS) 385
start (Windows) 384
stop (Linux) 385
stop (macOS) 385
stop (Windows) 384
CodeMeter SmartBind 21
CodeMeter Start Center 48
CodeMeter Time Server
Box Time (System Time CmContainer) 357
Certified Time 357
System Time 357
CodeMeter Universal Support Tool
cmu 452
CodeMeter WebAdmin
Certified Time Update 408

CodeMeter WebAdmin
Configuration| Access Control (advanced mode) 428, 429, 430
Configuration| Access Control (basic mode) 426
Firewall 400
Free licenses 412
Globale access rules 429
License Transfer History 438
License Transfer Options 437
Network Port 401
Profiling 427
Run CmWAN Server 426
Run Network Server 425
Server search list 417, 418
Specific access rules 430
Start 401
White and Blacklist 427
CodeMeter.ini
Profiling 375
CodeMeter.ini file 350
CodeMeterCSSI 25
Communication mode
IPv4, IPv5 401, 435
Platform-specific defaults 401, 435
Profiling 401, 435
Shared Memory 401, 435
Connecting the CmDongle 368
Copyright software licenses 468
CoreFX
Open Source 478
Customer Owned License Information (COLI)
Product Item Option 34
- D -
Deployment 345
"silent" installing of the runtime 348
Copy installation on Windows 351
Customizing Installation Packages (Windows) 347
directed installing of features (Windows) commandline 348
Installation roll back 346
Merge Module configuration parameter (Windows) 349
Merge Modules (Windows) 347
Mobile installation on CmDongle 350
Non-Windows operating systems 345
Preconfigured Installation Packages (Windows) 346
Start CodeMeter service 346
Windows operating systems 346
Disable Time 354
supending 354
Docker 330
Driver version (minim.)
AxProtector Commandline 264, 265
- E -
Enabling 353
Disable Time 354
Enabling Block 354
Enabling Level 355
Enabling Mode 354
Enabling Status 354
Lookup 355
Required Flag 355
Simple PIN 354
Time PIN 354
Encryption 44
Asymmetric 47
Direct 46
Indirect 46
Key derivation 44
Symmetric 46
Encryption Code Options 44
EWF/FBWF 330
Expiration Time
Product Item Option 30
Extended Protected Data
Product Item Option 35
- F -
Feature Code 31
AxProtector Commandline 264
Feature Map
Product Item Option 31
Version Management 31
FIO (Firm Item Options) 26
Firm Code 26
AxProtector Commandline 264
Firm Item 26
Firm Item Options (FIO) 26
Firm Item Text
define 317
Firm Key 26
Firm Security Box (FSB) 27
Firmware Version
AxProtector Commandline 265
FixKey
AxProtector 287
Flot
Open Source 468
Form Factors
CodeMeter 20
Human Interface Device (HID) 20
FSB (Firm Security Box) 27
- G -
golang.org (/x/sys / and /x/crypto)
Open Source 471
go-macaron
Open Source 469
Google Protocol Buffers
Open Source 472
- H -
Handles
automatic release 289
CodeMeter license access 289
HID
cmu programming 456
Set to HID 461
Set to Mass Storage Device 463
HID (Human Interface Device) 20, 368, 461
Hidden Data
Product Item Option 35
Human Interface Device (HID) 20, 461
- I -
IFI (Implicit Firm Item) 353
Implicit Firm Item 26
Individual Software Protection 289
Installation
32/64-bit Windows 370
Linux Operating Systems 373
macOS Operating Systems 372
IP Protection
AxProtector 287
AxProtector Commandline 'IP Protection' 264
IPv4, IPv6 401, 435
IxProtector
Modular Software Protection 290
IxProtector .NET
.NET Options 209
Activate IxProtector 210
Advanced Options 210
Create Logfile 210
Customized Error Messages 209
Default Error Messages 208
Destination File 207
Error Messages 208, 209

IxProtector .NET
Extended Commandline Options 210
File To Protect 207
Inline Messages 208
IxProtector Bytes 215
IxProtector Methods 215
IxProtector Name 215
IxProtector Views 214
License List Description 212
License List Feature Code 212
License List Firm Code 212
License List Id 211
License List Ignore Linger Time 213
License List Licensing Systems 212
License List Minimum Driver Version 213
License List Minimum Firmware 213
License List Product Code 212
License List Release Date 213
License List Subsystem 213
License List WupiReadData 213
License List WupiWriteData 213
Logging 210
No Strong Name 209
Optimizing 210
Source File 207
Strong Name from Container 209
Strong Name from File 209
Summary Back 216
Summary Finish 216
Summary Protect Now 216
User Message DLL 208
IxProtector .NET Standard
.NET Options 219
Activate IxProtector 220
Advanced Options 220
Create Logfile 220
Customized Error Messages 219
Default Error Messages 218
Destination File 217
Error Messages 218, 219
Extended Commandline Options 220
File To Protect 217
Inline Messages 219
IxProtector Bytes 225
IxProtector Methods 225
IxProtector Name 225
IxProtector Views 224
License List Description 222
License List Feature Code 222
License List Firm Code 222
License List Id 221
License List Ignore Linger Time 223
License List Licensing Systems 222
License List Minimum Driver Version 223
License List Minimum Firmware 223
License List Product Code 222
License List Release Date 223
License List Subsystem 223
License List WupiReadData 223
License List WupiWriteData 223
Logging 220
No Strong Name 219
Optimizing 220
Source File 217
Strong Name from Container 219
Strong Name from File 219
Summary Back 226
Summary Finish 226
Summary Protect Now 226
User Message DLL 218
IxProtector Java 280
IxProtector Linux
Advanced Options 239, 240
Create Logfile 240
Customized Error Messages 239
Default Error Messages 238
Destination File 238
Error Messsages 238, 239
Extended Commandline Options 239
File To Protect 238
IxProtector Function Length 244
IxProtector Function Description 244
IxProtector Function Id 244
IxProtector Function License List 245
IxProtector Function Name 244
IxProtector Function Trap 245
License List Feature Code 241
License List Firm Code 241
License List Minimum Driver Version 242
License List Minimum Firmware 242
License List Product Code 241
License List Release Date 242
License List Description 241
License List Id 240
License List Ignore Linger Time 242
License List Licensing Systems 241
License List Subsystem 242
License List WupiReadData 242
License List WupiWriteData 242
Logging 240
Source File 238
Summary Back 246
Summary Finish 246
Summary Protect Now 246
Suppress IxProtector Error Messages 238
User Message DLL 238
IxProtector macOS
Advanced Options 229
Create Logfile 229
Customized Error Messages 229
Default Error Messages 228
Destination File 228
Dynamic loading of Wibu-Systems libraries 229
Error Messsages 228, 229
Extended Commandline Options 229
File To Protect 227, 228
IxProtector Function Length 234
IxProtector Function Description 234
IxProtector Function Id 234
IxProtector Function License List 235
IxProtector Function Name 234
IxProtector Function Trap 235
License List Feature Code 231
License List Firm Code 231
License List Minimum Driver Version 232
License List Minimum Firmware 232
License List Product Code 231
License List Release Date 232
License List Description 231
License List Id 230
License List Ignore Linger Time 232
License List Licensing Systems 231
License List Subsystem 232
License List WupiReadData 232
License List WupiWriteData 232
Logging 229
Source File 227
Summary Back 236
Summary Finish 236
Summary Protect Now 236
User Message DLL 228
IxProtector Windows
Advanced Options 199
Create Logfile 199
Customized Error Messages 199
Default Error Messages 198
Destination File 197
Dynamic loading of Wibu-Systems libraries 199

IxProtector Windows
Error Messsages 198, 199
Extended Commandline Options 199
File To Protect 197
IxProtector Function Length 204
IxProtector Function Description 204
IxProtector Function Id 204
IxProtector Function License List 204
IxProtector Function Name 204
IxProtector Function Trap 204
IxProtector Funktion Name 78
License List Feature Code 201
License List Firm Code 201
License List Minimum Driver Version 202
License List Minimum Firmware 202
License List Product Code 201
License List Release Date 202
License List Description 201
License List Id 200
License List Ignore Linger Time 202
License List Licensing Systems 201
License List Subsystem 202
License List WupiReadData 202
License List WupiWriteData 202
Logging 199
Source File 197
Summary Finish 205
Summary Back 206
Summary Protect Now 206
Suppress IxProtector Error Messages 198
User Message DLL 198

- J -

Java IxProtector 280
JQuery
Open Source 468
JQuery-ui
Open Source 468

- K -

Key Derivation
Black Key 44
Feature Map 44
Firm Code 44
Product Code 44
Release Date 44

- L -

libcurl
Open Source 472
libsodium
Open Source 504
License information file (*.WibuCmLIF) 22
License Model
Concurrent License 41
Demo Version 41
Downgrade Management 42
Floating License 41
Hot / Cold Standby 42
License Borrowing 43
Local Single User Licenses 41
Machine-bound Licenses 43
Modular Licenses 41
Named User Licenses 43
Network License 41
Overflow Licenses 42
Pay-per ... 40, 42
Renting, Leasing 42
Standard 40
Version Management 42
License Quantity
Product Item Option 29
License request file
Add a license of a new ISV 396
create 394
Extend existing licenses 395
License Transfer
CmBoxPgm 323
cmu commands 454
concept 43
License update file
import 397
LicenseLock.log file 360
Licenses
*.WibuCmRac 393
*.WibuCmRaU 393
CmFAS 393
import 393
license request file 393
License update file 393
update 393
Linger Time
Product Item Option 33
Linux
CmDongle 464
LLVM compiler and toolchain technology
Open Source 474
Locale license
Remote Desktop 29
Server operating systems 29
Locking a CmContainer 359
Logging Informationen Java 285
Logging.properties file
AxProtector Java 285

- M -

Maintenance Period
Product Item Option 32
Maximum Encryption Rate
Product Item Option 36
Minimum Runtime Version
Product Item Option 33
Modular Software Protection 290
Module Item
concept 39

- N -

Named User
Product Item Option 34
Named User Licenses
cmu commands 455
Network access mode
AxProtector Commandline 265
None-Binding 22
Notations
IxProtector Java 280
Method encryption 280

- O -

Obfuscation AxProtector .NET 274
Obfuscation AxProtector Java 274, 275
On-demand Decryption 44
OOPE
Translocated Execution 286
Open Source
Anti-aliasing rasterizer from FreeType 2 490
Apache Commons 475
ASM Java Program Library 475
ASN.1 Compiler 473
Bootstrapper dotnetInstaller 473
Cocoa Platform Plugin 488
Contributions to the Cocoa Platform Plugin Files 495
CoreFX 478
Flot 468
FreeType 2 492
Freetype 2 - Bitmap Distribution Format (BDF) support 495
Freetype 2 - Portable Compiled Format (PCF) support 496
Freetype 2 - zlib 504

Open Source
golang.org (/x/sys / and /x/crypto) 471
go-macaron 469
Google Protocol Buffers 472
HarfBuzz 496
HarfBuzz NG 497, 498
IAccessible2 IDL Specification 488
JQuery 468
JQuery-ui 468
libcurl 472
LibJPEG-turbo 494
libsodium 504
LLVM compiler and toolchain technology 474
OpenGL ES 2 Headers 498
OpenGL Headers 499
Parts of QTemporaryFile 486
PCRE2 485
Public Suffix List 479
QEventDispatcher on macOS 487
qtmain Library 489
Text Codec: GBK 484
Text Codecs: EUC-JP, ISO 2022-JP (JIS), Shift-JIS 484
XCB 499
xkbcommon 500
XStream 478
Open source licenses 468
Operating Systems
CodeMeter 24
Own Key
Hidden Data 356
Secret Data 356
- P -
Password for CmDongle 353
CodeMeter Control Center 390, 391
PIO (Product Item Options) 27
Preconfigured Installation Packages (Windows)
Downgrade behaviour 346
Full Installation Package 346
Installation Package for applications using FSB functions 346
Merge modules 346
Reduced Installation Package 346
Product Code 26
AxProtector Commandline 264
Product Item Option 28
Product Item 26
Product Item Option
Access Password 36
Activation Time 30
Customer Owned License Information (COLI) 34
Expiration Time 30
Extended Protected Data 35
Feature Map 31
Hidden Data 35
License Quantity 29
Linger Time 33
Maintenance Period 32
Maximum Encryption Rate 36
Minimum Runtime Version 33
Named User 34
Product Code 28
Protected Data 35
Secret Data 36
Text 29
Unit Counter 31
Universal Data 37
Usage Period 31
User Data 34
Product Item Options (PIO) 26, 27
Profiling 401, 435
CodeMeter.ini 375
EWF (Enhanced Write Filter) 376
FBWF (File Based Write Filter) 376
Location different operating systems 457
Programming (CmBoxPgm)
CmActLicense License 331
Programming of CmContainer
*.wbb 342
*.WibuCmRaC 342
*.WibuCmRaM 342
*.WibuCmRaU 342
CmBoxPgm 315
CmLicense Editor 306
CodeMeter License Central 337
LIF, License Information File 342
Programming via file transfer 342
Programming Samples
Samples 14
Protected Data
Product Item Option 35
Protection Only license
CmActLicense binding 22
Programming example 334
Public Suffix List
Open Source 479
- R -
Receipt 398
Release Date 32
AxProtector Commandline 264
- S -
Samples
Programming 14
Schreibfilter
EWF (Erweiterter Schreibfilter, Enhanced Write Filter) 366
FBWF (Dateibasierter Schreibfilter, File Based Write Filter) 366
Search order for licenses
AxProtector Commandline 265
Search order for licenses (WAN)
AxProtector Commandline 265
Secret Data
Product Item Option 36
Secure License Tracking 456
Server operating systems
local license 29
Server Search List 417
*.ini configuration file 418
Server search list - automatic server search 418
Shared Memory 401, 435
Smart Bind
CmActLicense 329
CmBoxPgm 329
Support
Wibu-Systems 16
System startup
CodeMeter License Server 368
- T -
TCP/IP
Use in CodeMeter 52, 400
Temporary Enabling 354
Text
Product Item Option 29
Token 25
CodeMeter 25
Translocated Execution
OOPE 286
Trial license
CmActLicense binding 22
Programming example 333
Triple Mode Redundancy 322
- U -
UIK (User Individual Key) 353
Unit Counter

Unit Counter
Product Item Option 31
Universal Data
Product Item Option 37
Universal Firm Code
evaluation license 14
Usage Period 31
Product Item Option 31
User Data
Product Item Option 34
User Individual Key (UIK) 353

- W -

WAN
infrastructure 361
WAN, Wide Area Network 361
wbb file (CmActLicense) 368
wbc file 14
WibuCmNet-Bibliothek 51
Wide Area Network, WAN 361
Write Filter (EWF, FBWF) 376
WUPI
Example: WupiCalculator 293
WUPI Function
WupiAllocateLicense 291
WupiCheckDebugger 291
WupiCheckLicense 291
WupiDecreaseUnitCounter 291
WupiDecryptCode 291
WupiEncryptCode 291
WupiFreeLicense 291
WupiGetHandle 291
WupiGetLastError 293
WupiQueryInfo 291
WupiReadData 292
WupiReadDataInteger 292
WupiWriteData 293
WupiWriteDataInteger 293
WupiAllocateLicense 291
WupiCheckDebugger 291
WupiCheckLicense 291
WupiDecryptCode 291
WupiEncryptCode 291
WupiFreeLicense 291
WupiGetHandle 291
WupiGetLastError 293
WupiQueryInfo 291
WupiReadData 292
WupiReadDataInteger 292
WupiWriteData 293
WupiWriteDataInteger 293

- X -

X.509 Certificates
CodeMeter 25
XStream
Open Source 478