



CodeMeter Developer Guide

Version 5.10 - October 2013

Printed in Germany

All rights reserved. No part of this documentation, the accompanying software, or other components of the described product may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose other than the personal use of the purchaser without the express written permission of Wibu-Systems.

While the data contained in this document has been written with all due care, Wibu-Systems does not warrant or assume responsibility or represent that the data is free from errors or omissions.

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

WIBU, CodeMeter, SmartShelter are registered trademarks of Wibu-Systems. All other brand names and product names used in this documentation are trade names, service marks, trademarks, or registered trademarks of their respective owners.

Wibu-Systems is member of:



PCMCIA since 1993



USB Implementers Forum since 1997



SD Card Association since 2007



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



VDMA, German Engineering Federation since 2008



OPC Foundation since 2012

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



Microsoft Gold Certified Partner



Microsoft Embedded Partner



Strategic Software Partner Industrial and Medical

Table of Contents

| | | |
|------|--|----|
| I | Version | 13 |
| II | About this Guide | 15 |
| 1 | Safety Instructions | 17 |
| 2 | Installation | 17 |
| 3 | Shipped CmDongle | 17 |
| 4 | Additional Help Documentation | 18 |
| 5 | Typographical Conventions | 18 |
| 6 | Support by Wibu-Systems | 19 |
| 7 | About Wibu-Systems | 20 |
| III | Software Protection and License Management | 22 |
| 1 | CmDongle: CodeMeter Form Factors | 26 |
| 2 | CmActLicense: Binding and Activation | 27 |
| 2.1 | CmActLicense Binding | 28 |
| 2.2 | CmActLicense Activation | 29 |
| 3 | Operating Systems supported by CodeMeter | 30 |
| 4 | Additional Features | 31 |
| 5 | CodeMeter as Token | 32 |
| 6 | CodeMeter on Embedded Systems | 33 |
| IV | The CodeMeter Concept | 34 |
| 1 | Product Item Options - Custom-made License Entries | 37 |
| 1.1 | Product Code | 38 |
| 1.2 | Text | 38 |
| 1.3 | License Quantity | 39 |
| 1.4 | Activation Time | 40 |
| 1.5 | Expiration Time | 40 |
| 1.6 | Usage Period | 41 |
| 1.7 | Customer Owned License Information (COLI) | 42 |
| 1.8 | Unit Counter | 42 |
| 1.9 | Feature Map | 43 |
| 1.10 | Maintenance Period | 45 |
| 1.11 | Linger Time | 46 |
| 1.12 | User Data | 46 |

| | |
|---|-----------|
| 1.13 Protected Data | 47 |
| 1.14 Extended Protected Data | 47 |
| 1.15 Hidden Data | 48 |
| 1.16 Secret Data | 48 |
| 2 Security with Capital S | 49 |
| 3 License Models - Mapping Variety using CodeMeter | 50 |
| 3.1 Implementing License Models | 51 |
| 3.1.1 Local Single User Licenses..... | 51 |
| 3.1.2 Concurrent-/ Floating License in the Network..... | 52 |
| 3.1.3 Demo Versions | 52 |
| 3.1.4 Modular Licenses | 53 |
| 3.1.5 Leasing | 53 |
| 3.1.6 Pay-per-use Licenses..... | 53 |
| 3.1.7 Downgrade/Version Managment..... | 54 |
| 3.1.8 Overflow | 54 |
| 3.1.9 Hot / Cold Standby..... | 54 |
| 3.1.10 Named User Licenses..... | 55 |
| 3.1.11 Machine-bound Licenses..... | 55 |
| 3.1.12 License Borrowing | 55 |
| 4 Security by Encryption | 56 |
| 4.1 Key Derivation - One License Entry - Many Keys | 56 |
| 5 Cryptography | 58 |
| 5.1 Direct and Indirect Encryption | 59 |
| 5.2 Symmetric Enryption | 59 |
| 5.2.1 Streamcipher (AES_STREAM)..... | 60 |
| 5.2.2 Electronic Codebook Mode (AES_ECB)..... | 60 |
| 5.2.3 AES - Cipher Block Chaining Mode (CBC) (recommended)..... | 60 |
| 5.2.4 AES - Cipher Feedback Mode (CFB)..... | 61 |
| 5.3 Asymmetric Encryption | 61 |
| 5.3.1 ECC - Elliptic Curve Cryptography..... | 61 |
| 5.3.2 ECIES - Elliptic Curve Integrated Encryption Scheme..... | 61 |
| 5.3.3 ECDSA - Elliptic Curve Digital Signature Algorithm..... | 61 |
| 5.3.4 RSA | 62 |
| 5.4 Additional Encryption Algorithms | 62 |
| V CodeMeter Start Center | 63 |
| 1 Structure and Navigation | 63 |
| 1.1 Menu Bar | 63 |
| VI CodeMeter License Server | 65 |

| | |
|--|------------|
| VII Automatic Software Protection using AxProtector | 68 |
| 1 Structure and Navigation | 69 |
| 1.1 Menu Bar | 70 |
| 1.2 Navigation Window | 71 |
| 1.3 Input Window | 71 |
| 1.4 Note and Error Window | 71 |
| 1.5 Project type area | 71 |
| 2 Project Dialog | 72 |
| 3 Project Types | 72 |
| 4 AxProtector Tab | 73 |
| 4.1 Windows Application or DLL | 73 |
| 4.1.1 File to protect | 74 |
| 4.1.2 Licensing Systems | 75 |
| 4.1.3 License Handling | 77 |
| 4.1.4 Runtime Settings | 79 |
| 4.1.4.1 Advanced Runtime Settings | 81 |
| 4.1.5 Security Options | 84 |
| 4.1.5.1 Advanced Security Options | 88 |
| 4.1.6 Error Messages | 89 |
| 4.1.7 Advanced Options | 91 |
| 4.1.7.1 License Lists | 92 |
| 4.1.7.2 IxProtector | 96 |
| 4.1.7.3 File Encryption | 99 |
| 4.1.8 Summary | 103 |
| 4.2 .NET Assembly | 104 |
| 4.2.1 File to protect | 106 |
| 4.2.2 Licensing Systems | 107 |
| 4.2.3 License Handling | 109 |
| 4.2.4 Runtime Settings | 111 |
| 4.2.4.1 Advanced Runtime Settings | 113 |
| 4.2.5 Security Options | 116 |
| 4.2.6 Error Messages | 119 |
| 4.2.7 .NET Options | 121 |
| 4.2.8 Advanced Options | 122 |
| 4.2.8.1 License Lists | 123 |
| 4.2.8.2 IxProtector | 126 |
| 4.2.9 Summary | 129 |
| 4.3 Mac OS X Application or Dylib | 130 |
| 4.3.1 File to protect | 131 |

| | | |
|------------|---|------------|
| 4.3.2 | Licensing Systems | 132 |
| 4.3.3 | License Handling | 134 |
| 4.3.4 | Runtime Settings | 136 |
| 4.3.4.1 | <i>Advanced Runtime Settings</i> | 138 |
| 4.3.5 | Error Messages | 141 |
| 4.3.6 | Security Options | 141 |
| 4.3.6.1 | <i>Advanced Security Options</i> | 144 |
| 4.3.7 | Advanced Options | 145 |
| 4.3.7.1 | <i>License Lists</i> | 146 |
| 4.3.7.2 | <i>IxProtector</i> | 149 |
| 4.3.8 | Summary | 152 |
| 4.4 | Java Application (jar file) | 154 |
| 4.4.1 | File to protect | 157 |
| 4.4.2 | Licensing Systems | 158 |
| 4.4.3 | License Handling | 160 |
| 4.4.4 | Runtime Settings | 162 |
| 4.4.4.1 | <i>Advanced Runtime Settings</i> | 164 |
| 4.4.5 | Security Options | 167 |
| 4.4.6 | Error Messages | 168 |
| 4.4.7 | Java Options | 169 |
| 4.4.8 | Advanced Options | 171 |
| 4.4.9 | Summary | 172 |
| 4.5 | Linux Application or Shared Object | 173 |
| 4.5.1 | File to protect | 174 |
| 4.5.2 | Licensing Systems | 175 |
| 4.5.3 | License Handling | 177 |
| 4.5.4 | Runtime Settings | 180 |
| 4.5.4.1 | <i>Advanced Runtime Settings</i> | 182 |
| 4.5.5 | Security Options | 185 |
| 4.5.5.1 | <i>Advanced Security Options</i> | 187 |
| 4.5.6 | Error Messages | 188 |
| 4.5.7 | Advanced Options | 189 |
| 4.5.7.1 | <i>License Lists</i> | 190 |
| 4.5.7.2 | <i>IxProtector</i> | 193 |
| 4.5.8 | Summary | 196 |
| 5 | IxProtector Tab | 198 |
| 5.1 | Windows Application or DLL | 199 |
| 5.1.1 | File to protect | 200 |
| 5.1.2 | Error Messages | 201 |
| 5.1.3 | Advanced Options | 203 |

| | |
|--|------------|
| 5.1.3.1 <i>License Lists</i> | 204 |
| 5.1.3.2 <i>IxProtector</i> | 207 |
| 5.1.4 Summary | 210 |
| 5.2 .NET Assembly | 212 |
| 5.2.1 File to protect | 213 |
| 5.2.2 Error Messages | 214 |
| 5.2.3 .NET Options | 216 |
| 5.2.4 Advanced Options | 217 |
| 5.2.4.1 <i>License Lists</i> | 218 |
| 5.2.4.2 <i>IxProtector</i> | 221 |
| 5.2.5 Summary | 224 |
| 5.3 Mac OS X Application or Dylib | 225 |
| 5.3.1 File to protect | 226 |
| 5.3.2 Error Messages | 227 |
| 5.3.3 Advanced Options | 229 |
| 5.3.3.1 <i>License Lists</i> | 230 |
| 5.3.3.2 <i>IxProtector</i> | 233 |
| 5.3.4 Summary | 236 |
| 5.4 Linux Application or Shared Object | 238 |
| 5.4.1 File to protect | 240 |
| 5.4.2 Error Messages | 241 |
| 5.4.3 Advanced Options | 243 |
| 5.4.3.1 <i>License Lists</i> | 244 |
| 5.4.3.2 <i>IxProtector</i> | 247 |
| 5.4.4 Summary | 250 |
| 6 Other Tab | 252 |
| 6.1 File Encryption | 253 |
| 6.1.1 File to protect | 254 |
| 6.1.2 Licensing Systems | 255 |
| 6.1.3 License Handling | 258 |
| 6.1.4 Advanced Options | 260 |
| 6.1.4.1 <i>License Lists</i> | 261 |
| 6.1.4.2 <i>File Encryption</i> | 264 |
| 6.1.5 Summary | 267 |
| 7 Commandline Options for AxProtector | 270 |
| 7.1 Basic Options | 270 |
| 7.2 Options for the Licensing System | 271 |
| 7.3 Options for Encrypting and Decrypting | 274 |
| 7.4 Runtime Options | 286 |
| 7.5 Java-specific Settings | 290 |

| | |
|--|------------|
| 7.6 Operational Options | 295 |
| VIII Individual Software Protection | 296 |
| 1 IxProtector and Software Protection API (WUPI) | 296 |
| 2 WUPI Functions | 297 |
| 2.1 WUPI: example of index-based placeholders | 301 |
| 2.1.1 Definition of Modules | 302 |
| 2.1.2 Placeholders in IxProtector License and Functions Lists..... | 302 |
| 2.1.3 Programming the CmContainer..... | 306 |
| 2.1.4 Integration into the Source Code..... | 307 |
| 2.1.5 Encryption using AxProtector..... | 308 |
| 3 The CodeMeter Core API | 308 |
| 3.1 Functional Areas | 309 |
| 3.1.1 Access API | 309 |
| 3.1.2 Authentication API..... | 310 |
| 3.1.3 Enabling API | 310 |
| 3.1.4 Encryption API | 310 |
| 3.1.5 Error Management API..... | 311 |
| 3.1.6 Management API | 311 |
| 3.1.7 Programming API | 311 |
| 3.1.8 Remote Update API..... | 312 |
| 3.1.9 Time Management API..... | 312 |
| 3.2 CodeMeter API Guide | 313 |
| 3.2.1 Structure and Navigation..... | 314 |
| 3.2.2 Menu Bar | 315 |
| 3.2.3 Tabs | 316 |
| 3.2.4 Tree View | 316 |
| 3.2.5 Handle Display Window..... | 317 |
| 3.2.6 Interactive Area | 317 |
| 3.2.7 Source Code Area..... | 317 |
| 3.3 Sample Applications: CmDemo, CmCalculator, WupiCalculator | 317 |
| 3.3.1 CmDemo | 318 |
| 3.3.2 CmCalculator | 319 |
| 3.3.3 WupiCalculator | 319 |
| IX Programming of CmContainer and Licensing Management | 320 |
| 1 CodeMeter License Editor | 322 |
| 1.1 Structure and Navigation | 323 |
| 1.1.1 Menu Bar | 324 |
| 1.1.2 Symbol Bar | 324 |
| 1.1.3 Tree View | 325 |

| | | |
|------------|--|------------|
| 1.1.4 | Display Window | 326 |
| 1.1.5 | Output Window | 326 |
| 1.2 | Working with CodeMeter License Editor | 327 |
| 1.2.1 | Starting CodeMeter License Editor..... | 327 |
| 1.2.2 | Display of connected CmDongles..... | 327 |
| 1.2.2.1 | <i>Refreshing Display</i> | 327 |
| 1.2.2.2 | <i>Remote Programming Mode</i> | 327 |
| 1.2.3 | Creating and Editing a Firm Item..... | 328 |
| 1.2.4 | Deleting Firm Items..... | 328 |
| 1.2.5 | Creating and Editing a Product Item..... | 329 |
| 1.2.6 | Deleting a Product Item..... | 330 |
| 1.2.7 | Executing the Programming..... | 330 |
| 2 | CmBoxPgm | 331 |
| 2.1 | Commandline Syntax | 331 |
| 2.2 | Using CmBoxPgm | 332 |
| 2.3 | Basic Commands | 332 |
| 2.4 | CmContainer Options | 333 |
| 2.5 | Firm Item Options | 335 |
| 2.6 | Product Item Options | 336 |
| 2.7 | CmActLicense Options | 344 |
| 2.8 | License Borrowing Options | 352 |
| 2.9 | FSB Entry Options | 354 |
| 2.10 | Enabling Options | 355 |
| 2.11 | Special Commands | 357 |
| 3 | CodeMeter License Central | 360 |
| 3.1 | The Principle | 360 |
| 3.2 | The Architecture | 362 |
| 3.3 | Functions | 363 |
| 3.3.1 | Sales Interface | 363 |
| 3.3.1.1 | <i>Connectors</i> | 364 |
| 3.3.1.2 | <i>Gateway</i> | 365 |
| 3.3.2 | Depot Interface | 366 |
| 3.3.3 | Admin Interface | 367 |
| 3.4 | Application Scenarios CodeMeter License Central | 367 |
| 4 | Programming by File Transfer | 368 |
| X | Deployment | 372 |
| 1 | Installation packages for Non-Windows Operating Systems | 373 |
| 2 | Deployment on Windows Operating Systems | 373 |

| | | |
|------------|--|-----|
| 2.1 | Pre-configured Installation Packages | 374 |
| 2.2 | Customizing Options for Installation Packages | 375 |
| 3 | Mobile Installation on CmDongle (Windows) | 379 |
| 4 | CodeMeter Copy Installation on Windows | 381 |
| XI | Advanced CodeMeter Features | 383 |
| 1 | Implicit Firm Item (IFI) | 383 |
| 2 | Enabling | 383 |
| 2.1 | Enabling Blocks as On/Off switches | 385 |
| 2.1.1 | Enabling Access Code | 385 |
| 2.1.2 | Access Type - Simple or Time PIN | 385 |
| 2.1.3 | Enabling Mode | 386 |
| 2.1.4 | Deleting and Editing Enabling Blocks | 386 |
| 2.2 | Mapping (Lookup) of Enabling Blocks | 387 |
| 2.2.1 | Privileges - Enabling Level | 387 |
| 2.2.2 | Required Flag | 388 |
| 2.3 | Enabling Example | 388 |
| 3 | Using Own Keys | 391 |
| 4 | Time Server: System Times and Certified Time | 394 |
| 5 | Locking a CmContainer | 396 |
| 6 | Backup of CmDongle Content | 398 |
| 7 | CodeMeter in a Wide Area Network (WAN) | 399 |
| 7.1 | WAN Infrastructure | 399 |
| 7.2 | CodeMeter-sided Implementation | 400 |
| 7.2.1 | Programming of licenses | 401 |
| 7.2.2 | Encrypting the application to be protected | 404 |
| 7.2.3 | Configuring CmWAN network communication | 404 |
| 7.2.3.1 | <i>CodeMeter WebAdmin Configuration</i> | 405 |
| 7.2.3.2 | <i>Profiling in Registry or in server.ini File</i> | 405 |
| XII | Manual | 408 |
| 1 | First important Information | 408 |
| 2 | CodeMeter Control Center | 411 |
| 2.1 | Structure and Navigation | 413 |
| 2.2 | Menu Bar | 414 |
| 2.3 | License Tab | 417 |
| 2.4 | Events Tab | 421 |
| 2.5 | Borrowing Tab | 421 |

| | |
|--|------------|
| 2.6 Status and Starting CodeMeter WebAdmin | 424 |
| 3 Importing and Updating Licenses | 424 |
| 3.1 The CmFAS Assistant in CodeMeter Control Center | 425 |
| 3.1.1 Create License Request File..... | 427 |
| 3.1.1.1 Extend Existing License..... | 428 |
| 3.1.1.2 Add a License of a new Producer..... | 430 |
| 3.1.2 Import License Update..... | 431 |
| 3.1.3 Create Receipt | 433 |
| 4 CodeMeter WebAdmin | 435 |
| 4.1 Basics | 436 |
| 4.2 Starting CodeMeter WebAdmin | 438 |
| 4.3 Status Infomation | 438 |
| 4.3.1 General Information..... | 439 |
| 4.3.2 Information on CmContainer..... | 441 |
| 4.4 Configuration | 442 |
| 4.4.1 Network | 443 |
| 4.4.2 Server | 445 |
| 4.4.3 Proxy Settings | 446 |
| 4.4.4 Access Control | 447 |
| 4.4.5 Certified Time | 450 |
| 4.4.6 WebAdmin | 451 |
| 4.4.7 Backup | 452 |
| 4.4.8 Borrowing | 453 |
| 4.5 License Display | 454 |
| 4.5.1 Local Licenses | 454 |
| 4.5.1.1 Licensor Information (ISV)..... | 455 |
| 4.5.1.2 Product Information..... | 456 |
| 4.5.2 User Data | 458 |
| 4.6 License Display on the Network | 459 |
| 4.6.1 Cluster - Licenses summarized..... | 460 |
| 4.6.1.1 Session Details | 461 |
| 4.6.2 Current User | 463 |
| 4.6.3 License Tracking | 464 |
| 4.7 Diagnosis | 469 |
| 4.8 Backup/Restore | 470 |
| 4.9 Info | 472 |
| 4.10 Help | 472 |
| 5 CmDust | 472 |
| 6 CMU - CodeMeter Universal Support Tool | 474 |

| | |
|--|------------|
| 7 CodeMeter License Tracking | 477 |
| 7.1 Requirements | 478 |
| 7.2 Configuration | 478 |
| 7.2.1 Profiling | 478 |
| 7.3 Logfile Format | 479 |
| 7.3.1 Definitions and Value Ranges..... | 479 |
| 7.4 Entry Types | 480 |
| 7.4.1 List of Licenses Entry..... | 480 |
| 7.4.2 License Entry | 480 |
| 7.4.3 Access Entry | 481 |
| 7.4.4 Release Entry | 481 |
| 7.4.5 Borrow Access Entry..... | 481 |
| 7.4.6 Borrow Return Entry..... | 482 |
| 7.4.7 Denial Entry | 482 |
| 7.4.8 Administrative Entry..... | 482 |
| 8 HID Support | 482 |
| 8.1 Set from Mass Storage to HID | 483 |
| 8.2 Set from HID to Mass Storage | 485 |
| XIII Glossary | 488 |
| Index | 491 |

1 Version

CodeMeter Developer Guide Version 5.10, 16.10.2013.

Copyright © 2007-2013

by WIBU-SYSTEMS AG, Karlsruhe / Germany

All rights reserved.

Wibu-Systems contact information:

Address: WIBU-SYSTEMS AG
Rueppurrer Strasse 52-54
D-76137 Karlsruhe, Germany

Phone: +49 (0)-721-93172-0

Internet: <http://www.wibu.com>

E-mail: support@wibu.com

AxProtector Java uses the ASM Java program library.

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.

CodeMeter WebAdmin uses jQuery functions.

Copyright 2013 jQuery Foundation and other contributors

<http://jquery.com/>

Permission is hereby granted, free of charge, to any person obtaining

a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2 About this Guide

CodeMeter® is the technology of Wibu-Systems providing secure protection and effective license management of software and 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](#)²² sketches the outstanding features of *CodeMeter®* in the areas of security, hardware and software-based software protection, and flexible license management. [Part III](#)³⁴ follows describing how the concept of *CodeMeter®* meets protection, licensing, and security requirements. Moreover, basic terms are introduced.

[Part IV](#)⁶³ describes *CodeMeter Start Center*, the communication turntable to open single *CodeMeter®* tools, while [Part V](#)⁶⁵ 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](#)⁶⁸ and [Part VII](#)²⁸⁶ 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](#)²⁰⁰ 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](#)³⁷² follows with a description of deployment options: what does your customer need for running the protected software?

[Part X](#)³⁸⁸ 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](#)⁴⁰⁸ 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.

| Software Protection (Encryption) and Software Integration | Navigation | Programming and Production of CmContainer, Management of Licenses |
|---|--|--|
| <p>Automatic Integration</p> <ul style="list-style-type: none"> AxProtector Protection Technology and Tool for automatic protection of applications without changes in the sourcecode while developing. <p>Individual Integration</p> <ul style="list-style-type: none"> IxProtector Protection Technology integrated in AxProtector for individual protection of applications with changes in the sourcecode while developing. Software Protection API WUPI Lean Interface (WIBU Universal Protection Interface) and Tool for individual protection of applications with changes in the sourcecode while developing. Core API Interface and Tools for individual protection of applications with changes in the sourcecode. Including CodeMeter API Guide for interactive use while developing and at runtime. | <p>CodeMeter Start Center Introduction page for accessing the basic Tools and Interfaces..</p>  <p>Runtime Components for the Developer, Administrator and End Customer</p> <ul style="list-style-type: none"> CodeMeter Control Center User Interface for local configuration of CodeMeter License Server including License Updating using, CmFAS Assistant. CodeMeter WebAdmin Application for configuration of CodeMeter License Server and display of existing licenses in the CmContainer. | <p>CodeMeter License Central (Desktop / Internet) Tool for integration of software protection into sales, production, and support processes (Backoffice Integration).</p> <p>CodeMeter Producer Database-oriented Tools for programming of CmContainer replaced by CodeMeter License Central. Development ceased with version 4.1.</p> <p>CmBoxPgm Commandline Tool for batch programming of CmContainer with scripts in the production.</p> <p>CodeMeter License Editor Graphical Tool for programming of CmContainer to test license strategies.</p> <p>Programming API Interface for programming of CmContainer and license management in own applications, (HIP, High Level Programming Interface).</p> |

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](http://www.wibu.com/en/certificates.html) of Wibu-Systems (<http://www.wibu.com/en/certificates.html>).

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

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

Non-compliance with the safety instructions result 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.



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 CmDongle

Together with the *CodeMeter®* Software Development Kit (SDK) you received a dongle, the *CodeMeter® CmDongle*.

This dongle simultaneously acts as the 'leading' Master *CmDongle*, the so-called Firm Security Box which allows you to program other *CmContainer*.



For *CmDongle* an evaluation license entry with the public Firm Code 10 and Product Code 13 is already programmed.



For the software-based *CodeMeter®* variant *CmActLicense* you have an evaluation public Firm Code 5010 and Product Code 13.

When you later decide to go live with *CodeMeter®* you will receive your own individual Firm Code. You receive a *CmFirm.wbc* file (if you use *CmActLicense* an additional *CmActFI.wbc*). For importing the file drag & drop them into *CodeMeter Control Center*. You find the files in the directory "C:\ProgramData\CodeMeter\DevKit".

Then using this Firm Security Box as licensor you are able to transfer license information into other *CmContainer*.

2.4 Additional Help Documentation

In addition to this Developer Guide (accessible via "**Start | All Programs | CodeMeter – Documentation**") , the following help files are available. You open them either in the respective applications, or you find them in "**Start | All Programs**" menu after installing the SDK (Software Development Kit).

| Help File | Accessible by: |
|--|--|
| <i>CodeMeter®</i> User Help as HTML files including the parts <i>CodeMeter® Runtime Kit</i> , <i>CodeMeter License Server</i> , <i>CodeMeter Control Center</i> , <i>cmu</i> commandline program, <i>CodeMeter WebAdmin</i> , Licensing - Field-Activation-Service, <i>CodeMeter® FAQ</i> (German and English) | respective menu items, buttons or " Start All Programs CodeMeter Documentation " [%Program Files%\CodeMeter\Runtime\help\CMUserHelp] |
| <i>AxProtector</i> online help as compiled HTML help in German and English | " Start All Programs AxProtector Help " |
| <i>Software Protection API</i> as compiled HTML help in English | " Start All Programs CodeMeter Documentation " |
| <i>Core API</i> as compiled help file in English | " Start All Programs CodeMeter Documentation " |
| <i>CodeMeter Java-API</i> as HTML files in English | " Start All Programs CodeMeter Documentation " |
| <i>Programming API</i> as HTML files in English | " Start All Programs CodeMeter Documentation Programming API " for the programming languages C++, Delphi, Java. |
| Samples for programming of <i>CmContainer</i> and related Sample Help documentation | " Start All Programs CodeMeter Samples " [%\Users%\Public\Documents\WIBU-SYSTEMS] Help " Start All Programs CodeMeter Documentation " |

2.5 Typographical Conventions

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

| Format definition | Information type |
|-----------------------------|----------------------|
| <i>Italics</i> | <i>Product names</i> |
| Arial Narrow <i>Italics</i> | Important terms |

| Format definition | Information type |
|-------------------------------|---|
| Arial Narrow <i>Italics</i> | <i>Properties</i> |
| "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®** 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 100, the majority in the headquarters facility in Karlsruhe, Germany. Subsidiaries are in Seattle (USA), Shanghai and Beijing (China), with sales offices as well in Belgium, Great Britain, the Netherlands, Portugal and Spain, and distributors in more than 25 countries. Corporate efforts stress achieving world-class quality in the areas of security, reliability, durability, support, and customer service.

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

- *CmDongle* the hardware-based variant of the protection and licensing technology *CodeMeter®* is available in many form factors for a variety of interfaces and allows for multiple ISVs to share a single *CmDongle*, easy online license transfers, and optional Flash disk in different sizes.
- *CmActLicense* is a completely software-based variant of the protection and licensing technology *CodeMeter®* that protects software by binding to the characteristics of an individual PC or any target system.
- *CodeMeter License Central* creates, manages, 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 a certified ISO 9001:2008 manufacturer and 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 | +86 (0) 21-55661790 | info@wibu.com.cn |

| Contact Information | | |
|---------------------|-----------------------------------|-----------------------|
| | Beijing +86 (0) 10-82961560/61 | info@wibu.com.cn |
| Great Britain, UK | +44 (0) 20 314 747 27 | sales@wibu.co.uk |
| Ireland | +44 (0) 20 314 747 27 | sales@wibu.uk.co |
| 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.pt |
| 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 "end user" 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 IME II IT I/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.

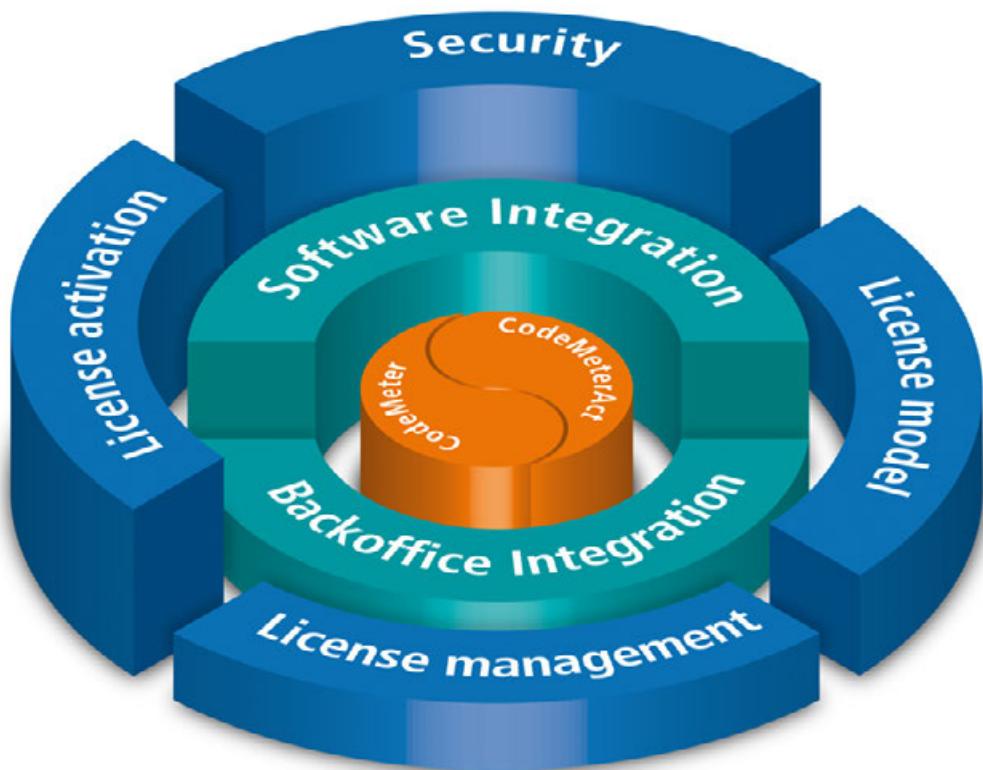


Figure 3: Overview - Software Protection and License Management using *CodeMeter*®

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.
- ⌚ 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 when 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-proof 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 |

| License strategy | License model |
|--|--|
| <ul style="list-style-type: none"> ⌚ The <i>CodeMeter® SmartCard Chip</i> with 60/384 kByte memory allows the programming of up to 6,000 license entries into a single <i>CmDongle</i>. ⌚ Vendor-independent use and management of license entries by unique and secure separation of individual license container in a <i>CmDongle</i>. Thus several software vendors are able to share a single <i>CmDongle</i>. | Machine bound Licenses License Borrowing Volume Licenses |

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, Mac Os X, 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*.

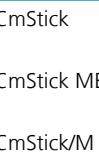
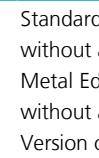
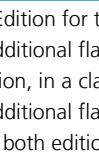
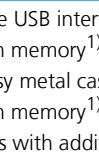
- Decryption and encryption of *IxProtector* protected areas at runtime using WUPI (*WIBU Universal Protection Interface*). This lean *Software Protection API* providing few but essential functions is universally applicable for many programming languages.
- ➲ Additional requirements (encryption and decryption 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 when 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 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 <i>CmDongle</i> |
|  | 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](#)⁴⁸².

3.2 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.

3.2.1 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](#)³⁴⁵ optimizes assuring the validity of licenses, in the case of changing hardware properties of the computer or device to which the licenses are bound.

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.

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 as default binding scheme. For programming of *CmActLicense* licenses using the binding scheme *SmartBind* with *CmBoxPgm* see [here](#)³⁴⁸.

For single cases *CmActLicense* also supports binding schemes which refer either to specify [fix](#)³⁴⁵ or [configurable](#)³⁴⁶ hardware features of a computer or a device. However, Wibu-Systems recommends to contact Wibu-Systems support before using these options.

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](#)³⁴⁶ 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](#)³⁴⁴ files when the binding value is known. The step to create a license request file on the target system then is only optional at a later activation.

None Binding

Using the binding scheme [None](#)³⁴⁶ allows you to deliver protected software without the binding to a

specific computer or device.

This is the case, for example, if 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](#)"³⁴⁹ license model allowing you to create demo licenses which are valid for a maximum of 90 days. These licenses expire after this period and are not re-importable.

An additional use case is creating time-unlimited and re-importable licenses for any computer or device. This is relevant when primarily preventing reverse engineering is wanted. Here Wibu-Systems offers the "[Protection Only](#)"³⁵¹ license model.

For both 'None-Bind' based license models a separate license entry in the [Firm Security Box](#)³⁶ (FSB) is required you receive from Wibu-Systems on request.

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. |

3.2.2 CmActLicense Activation

Largely, activating CmActLicense licenses is based on the standardized CodeMeter® procedure for file-based remote programming of [CmDongles](#)³⁶⁸. 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 (*.lif). This file corresponds to an empty license container however holds specifications on [binding schemes](#)³²⁸ and [additional activation options](#)³²⁹ 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. 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:

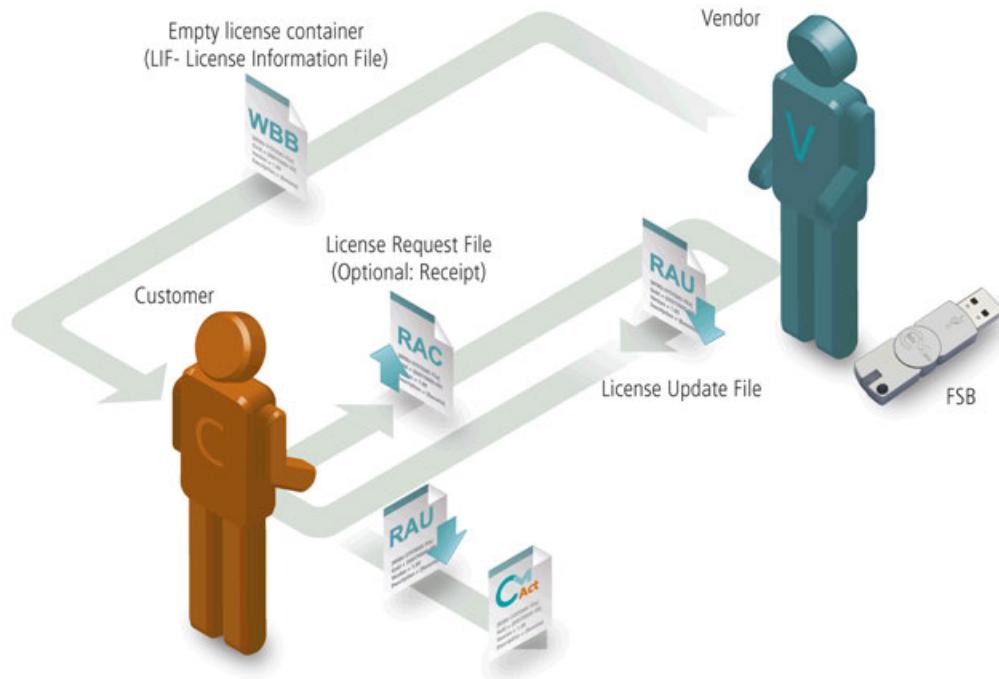


Figure 5: *CmActLicense* - Activation by file-based remote programming (CmFAS, CodeMeter Field Activation Service)

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 (*.lif file) 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.

3.3 Operating Systems supported by CodeMeter

CodeMeter® is available for many operating systems and runtime environments, such as, Windows 32-bit/64-bit, Mac OS X, Linux 32-bit/64-bit, Java, Sun Solaris 10, .NET.

| Operating systems | CodeMeter |
|-------------------|-----------|
| Windows XP | ✓ |
| Windows Vista | ✓ |

| Operating systems | CodeMeter |
|---------------------|-----------|
| Windows 7 | ✓ |
| Windows 8 | ✓ |
| Windows 2000 Server | ✓ |
| Windows 2003 Server | ✓ |
| Windows 2008 Server | ✓ |
| MacOS X | ✓ |
| Linux | ✓ |
| Sun Solaris 10 | ✓ |
| Windows XP Embedded | ✓ |
| Windows CE 5.0 | ✓ |
| Windows CE 6.0 | ✓ |
| VxWorks | ✓ |

3.4 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, Mac) 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.

3.5 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 2048 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, Mac OS X, 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.

3.6 CodeMeter on Embedded Systems

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

CodeMeter® Compact Driver 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® Compact Driver* 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.



When 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.

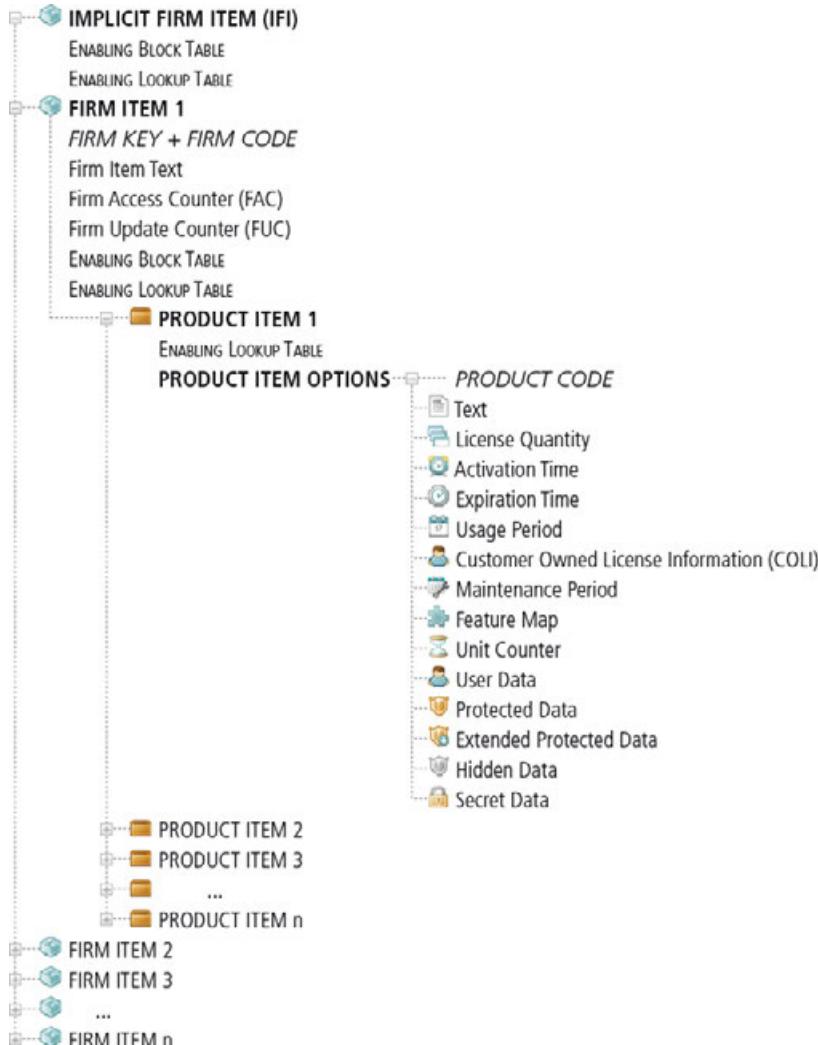


Figure 6: CmContainer License Entry Organization

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](#))³⁷.

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](#)³⁸).

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

And finally, 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 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 lose 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 <i>CodeMeter WebAdmin</i> | ✓ | ✓ | ✗ |
| License Quantity | Number of simultaneously usable licenses, use for floating concurrent licenses on the network | ✓ | with FSB | ✗ |
| Activation Time | Use for time-limited versions | ✓ | with FSB | ✓ |
| Expiration Time | Use for time-limited versions | ✓ | with FSB | ✓ |
| Usage Period | Use for time-limited versions | ✓ | initially at first start | ✓ |
| Customer Owned License Information | 128 character, use for customer-specific data (e.g. name of the licensee) | ✓ | with FSB | ✓ |
| 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 | ✓ |
| 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 | ✓ | ✓ | ✗ |
| Protected Data | 256 byte for saving additional data | ✓ | with FSB | ✗ |
| Extended Protected Data | (128 +128) x 256 bytes ¹⁾ | ✓ | with FSB | ✗ |
| Hidden Data | (128 +128) x 256 byte data, use as key source ^{1) 2)} | with password | with FSB | as separate key |

| Product Item Option | Description | Read Access | Write Access | Integrated in Encryption |
|---------------------|---|-------------|--------------|--------------------------|
| Secret Data | (128 +128) x 256 byte data, use as key source ¹⁾ | X | with FSB | as separate key |

¹⁾ (128 +128) x 256 bytes: 128 (0-127) available, 128 (128-255) reserved for Wibu-Systems

²⁾ The [reading](#)³⁰⁰ and [writing](#)³⁰⁰ 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 | |
|---------------------------------------|---|
| <i>CodeMeter License Editor</i> | Programming the Product Codes ³²⁹ |
| <i>CmBoxPgm</i> | Programming the PIO /p ³³⁷ |
| <i>CodeMeter License Central</i> | Programming ³⁶⁷ the PIO |
| Query/Check | |
| <i>AxProtector</i> | Product Code has to be defined. |
| <i>Software Protection API (WUPI)</i> | WapiCheckLicense ²⁹⁸ or WapiAllocateLicense ²⁹⁸ WapiQueryInfo ²⁹⁹ Query information about the currently allocated license entry. |
| <i>Core API</i> | CmAccess ³⁰⁰ and handle editing using CmCrypt ³¹⁰ |

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 ³²⁹ the PIO Text |
| CmBoxPgm | Programming the PIO /pt ³⁴² |
| CodeMeter License Central | Programming ³⁶⁷ the PIO |
| Programming API | Call class ProductItemParamSet ³²¹ and subsequently SetProductItemText ³²¹ |
| Query/Check | |
| AxProtector | --- |
| Software Protection API (WUPI) | --- |
| Core API | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

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 <i>CmContainer</i> 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.

| |
|---|
|  Setting this option to a value of 0 defines an exclusively local license. |
|---|

- 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 ³²⁹ the PIO License Quantity |
| CmBoxPgm | Programming the PIO /plq ³⁴⁰ |
| CodeMeter License Central | Programming ³⁶⁷ the PIO |
| Programming API | Call class ProductItemParamSet ³²¹ and subsequently SetAbsoluteLicenseQuantity ³²¹ or SetRelativeLicenseQuantity ³²¹ |
| Query/Check | |
| AxProtector | License handling and License Options |
| Software Protection API (WUPI) | --- |
| Core API | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

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](#)⁴⁰, 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](#)³⁹⁴.
- The Activation Time is part of the [key derivation](#)⁵⁶. 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 ³²⁹ the PIO  Activation Time |
| CmBoxPgm | Programming the PIO /pat ³³⁷ |
| CodeMeter License Central | Programming ³⁶⁷ the number of days the application allowed to run from the first start |
| Programming API | Call class ProductItemParamSet ³²¹ and subsequently SetAbsoluteActivationTime ³²¹ or SetRelativeActivationTime ³²¹ |
| Query/Check | |
| AxProtector | Select options in advanced runtime settings |
| Software Protection API (WUPAPI) | WupiQueryInfo ²⁹⁹ Query information about the currently allocated license entry. |
| Core API | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

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](#)⁴⁰, you are able to implement time-limited license models, e.g. leasing, subscription, etc.

- An Expiration Time defines a split second value in intervals between January 1st, 2000, 0:00:00 and December 31st, 2099, 23:59:59. This value is always saved in the time zone format UTC (Universal Time Coordinated) and is independent of a time zone or a daylight savings time interval.
- Access to the license is granted only when the Box Time and the Certified Time in the *CmContainer* precede the defined Expiration Time. For the fail safe and manipulation safe control mechanism see [here](#)
- The Expiration Time is part of the [key derivation](#)⁵⁶. 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 | |
|--------------------------------------|---|
| <i>CodeMeter License Editor</i> | Programming ³²⁹ the PIO  Expiration Time |
| <i>CmBoxPgm</i> | Programming the PIO /pet ³³⁸ |
| <i>CodeMeter License Central</i> | Programming ³⁶⁷ the PIO |
| <i>Programming API</i> | Call class ProductItemParamSet ³²¹ and subsequently SetAbsoluteExpirationTime ³²¹ or SetRelativeExpirationTime ³²¹ |
| Query/Check | |
| <i>AxProtector</i> | Selecting options in advanced runtime settings. |
| <i>Software Protection API (WUP)</i> | WupiQueryInfo ²⁹⁹ Query information about the currently allocated license entry. |
| <i>Core API</i> | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

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.

- The definition and programming of the Usage 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 | |
|----------------------------------|---|
| <i>CodeMeter License Editor</i> | Programming ³²⁹ the PIO  Usage Period |
| <i>CmBoxPgm</i> | Programming the PIO /pup ³⁴³ |
| <i>CodeMeter License Central</i> | Programming ³⁶⁷ of number of days how long the application is allowed to run |

| Create/Edit/Delete | |
|---------------------------------------|--|
| | after the initial start |
| <i>Programming API</i> | Call class ProductItemParamSet ³²¹ and subsequently SetUsagePeriod ³²¹ |
| Query/Check | |
| <i>AxProtector</i> | --- |
| | (however, applying Expiration Time settings) |
| <i>Software Protection API (WUPI)</i> | WupiQueryInfo ²⁹⁹ |
| | Query information about the currently allocated license entry. |
| <i>Core API</i> | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

4.1.7 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.
- 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 | |
|---------------------------------------|---|
| <i>CmBoxPgm</i> | Programming the PIO /pcoli ³³⁸ |
| <i>Programming API</i> | --- |
| Query/Check | |
| <i>AxProtector</i> | --- |
| <i>Software Protection API (WUPI)</i> | --- |
| <i>Core API</i> | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

4.1.8 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 bits) 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 99999. 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](#)⁵⁶. 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 | |
|---------------------------------------|--|
| <i>CodeMeter License Editor</i> | Programming ³²⁹ the PIO Unit Counter |
| <i>CmBoxPgm</i> | Programming the PIO puc ³⁴² |
| <i>CodeMeter License Central</i> | Programming ³⁶⁷ the PIO |
| <i>Programming API</i> | Call class <i>ProductItemParamSet</i> ³²¹ and subsequently <i>SetAbsoluteUnitCounter</i> ³²¹ or <i>SetRelativeUnitCounter</i> ³²¹ |
| Query/Check | |
| <i>AxProtector</i> | If an Unit Counter exists in the <i>CmContainer</i> , <i>AxProtector</i> automatically will decrement it when the software is started. However, you are able to change the decrement. On the <i>AxProtector</i> page "Runtime settings" you are able to check the license also at runtime using an existing Unit Counter. |
| <i>Software Protection API (WUPI)</i> | <i>WupiDecreaseUnitCounter</i> ²⁹⁸ Decrementing of the Unit Counter of a license which in <i>AxProtector</i> is defined with the <code>Id = LicenseId</code> <i>WupiQueryInfoId</i> ²⁹⁹ Query information about the currently allocated license entry. |
| <i>Core API</i> | <i>CmCrypt</i> ³¹⁰ |

4.1.9 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.

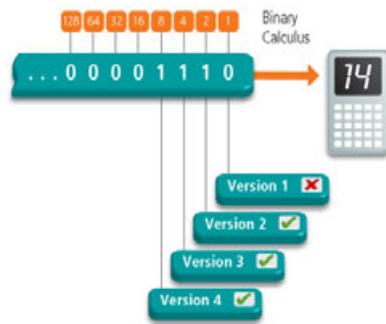


Figure 7: Version Management using PIO Feature Code

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](#)⁵⁶ 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](#)⁵⁶. 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

| | |
|----------------------------------|---|
| <i>CodeMeter License Editor</i> | Programming ³²⁹ the PIO Feature Map |
| <i>CmBoxPgm</i> | Programming the PIO /pfm ³³⁹ |
| <i>CodeMeter License Central</i> | Programming ³⁶⁷ the PIO |
| <i>Programming API</i> | Call class ProductItemParamSet ³²¹ and subsequently SetFeatureMap ³²¹ |

| Query/Check | |
|--------------------------------|--|
| AxProtector | Must be defined |
| Software Protection API (WUPI) | WupiQueryInfo ²⁹⁹ Query information about the currently allocated license entry. |
| Core API | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

4.1.10 Maintenance Period

The PIO Maintenance Period serves to store an absolute 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 exists 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 ³³⁰ the PIO Maintenance Period |
| CmBoxPgm | Programming the PIO /pmd ³⁴⁰ |
| CodeMeter License Central | Not yet implemented |
| Programming API | Call class MaintenancePeriodParamSet ³²¹ and subsequently MaintenancePeriodPIO ³²⁰ |
| Query/Check | |
| AxProtector | May be activated and checked |
| Software Protection API (WUPI) | WupiQueryInfo ²⁹⁹ Query information about the currently allocated license entry. |
| Core API | CmCrypt2 ³¹⁰ , CmAccess2 ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

4.1.11 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 <i>CmContainer</i> 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 <u>once</u> 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 <u>not</u> 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 | |
|----------------------------------|--|
| <i>CodeMeter License Editor</i> | Not yet implemented |
| <i>CmBoxPgm</i> | Programming the PIO /plt ³⁴⁰ |
| <i>CodeMeter License Central</i> | Not yet implemented |
| <i>Programming API</i> | Call class LingerTimeParamSet ³²¹ and subsequently LingerTimePIO ³²⁰ |
| Query/Check | |
| <i>AxProtector</i> | May be ignored |
| <i>Core API</i> | CmAccess2 ³⁰⁹ |

4.1.12 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 | |
|--------------------|---|
| <i>CmBoxPgm</i> | Programming the PIO /pud ³⁴³  |

| Create/Edit/Delete | |
|--------------------------------|---|
| CodeMeter License Central | Programming  the PIO |
| Programming API | Call class ProductItemParamSet  and subsequently SetUserData  |
| Query/Check | |
| AxProtector | --- |
| Software Protection API (WUPI) | --- |
| Core API | CmAccess  and in Managing API GetBoxContents  |

4.1.13 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 double byte.
- 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 | |
|--------------------------------|--|
| CmBoxPgm | Programming the PIO /ppd  |
| CodeMeter License Central | Programming  the PIO |
| Programming API | Call class ProductItemParamSet  and subsequently SetProtectedData  |
| Query/Check | |
| AxProtector | --- |
| Software Protection API (WUPI) | --- |
| Core API | CmAccess  and in Managing API GetBoxContents  |

4.1.14 Extended Protected Data

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

- The PIO Extended Protected Data comprises (128 + 128) types. Each type may have a length of 256 bytes.



Of the types 128 (0-127) are reserved for the licensor and 128 (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 | |
|---------------------------|--|
| CmBoxPgm | Programming the PIO /ped  |
| CodeMeter License Central | Programming  the PIO |

| Create/Edit/Delete | |
|--------------------------------|--|
| Programming API | Call class ProductItemParamSet ³²¹ and subsequently SetExtendedProtectedData ³²¹ |
| Query/Check | |
| AxProtector | --- |
| Software Protection API (WUPI) | --- |
| Core API | CmAccess ³⁰⁹ and in Managing API GetBoxContents ³¹¹ |

4.1.15 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 (128 + 128) types. Each type may have a length of 256 bytes.



Of the types 128 (0-127) are reserved for the licensor and 128 (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](#)³⁰⁰ and [writing](#)³⁰⁰ of data from or into a *CmContainer* is featured also without FSB access at runtime using [WUPI functions](#)²⁹⁷, 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 | |
|--------------------------------|---|
| CmBoxPgm | Programming the PIO /phd ³³⁹ |
| CodeMeter License Central | Programming ³⁶⁷ the PIO |
| Programming API | Call class ProductItemParamSet ³²¹ and subsequently SetHiddenData ³²¹ |
| Query/Check | |
| AxProtector | --- |
| Software Protection API (WUPI) | WupiReadData ³⁰⁰ or WupiReadDataInteger ³⁰⁰ , WupiWriteData ³⁰⁰ or WupiWriteDataInteger ³⁰¹ |
| Core API | CmAccess ³⁰⁹ and in the Managing API GetBoxContents ³¹¹ |

4.1.16 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 (128 + 128) types. Each type may have a length of 256 bytes.



Of the types 128 (0-127) are reserved for the licensor and 128 (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 | |
|---------------------------------------|---|
| <i>CmBoxPgm</i> | Programming the PIO /psd   |
| <i>CodeMeter License Central</i> | Programming the PIO |
| <i>Programming API</i> | Call class ProductItemParamSet  and subsequently SetSecretData  |
| Query/Check | |
| <i>AxProtector</i> | --- |
| <i>Software Protection API (WUPI)</i> | --- |
| <i>Core API</i> | CmAccess  and in the Managing API GetBoxContents  |

4.2 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 <i>CmDongle</i> . 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 <i>CmDongle</i> 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 <i>CmDongle</i> behaves as if those licenses (and the keys involved) were not present. The hacker does not have a second try. |
| Counters cannot be set back by a backup | Counters are safely stored in the Smartcard chip of the <i>CmDongle</i> . 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 <i>CmDongle</i> no longer exist. By transfer of a receipt, the developer is sure that the license does not exist in the current <i>CmDongle</i> 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 <i>CmDongles</i> . This locking is stored in these <i>CodeMeter®</i> time server, and as soon as the affected <i>CmDongle</i> is trying to update this time it is locked. |

| Advantage | Description |
|--|--|
| 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 <i>CodeMeter®</i> , both the user and developer are winners; since the license is contained on the <i>CmDongle</i> , the user can move it by simply relocating the <i>CmDongle</i> . 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 <i>CmDongle</i> 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

4.3 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 (<i>CmActLicense</i>) or in a <i>CmDongle</i> 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. |
| 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. |

| License model | Description |
|----------------------------------|---|
| Volume license (without control) | The key customer is sent an activation code which they can use as often as they want. The number of licenses appears in the contract but is not controlled (<i>CmActLicense</i> 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 (<i>CmActLicense</i>) or in a <i>CmDongle</i> for a fixed time. When the license expires, it is automatically returned to the license server and can no longer be locally accessed. It is also possible to manually return the license before the expiry date. |
| User-specific licenses | The license is associated with a specific user name. |
| Computer-specific licenses | The license is associated with a specific computer name. |
| Time zone licenses | The license can only be used in the geographical region (time zone) specified by you. |

Table 4: Mapping License Models using *CodeMeter*[®]

4.3.1 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](#)³²⁰. 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.

4.3.1.1 Local Single User Licenses

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



Define a freely chosen Product Code at the Product Item level. Set the Product Item Option License Quantity to a value of 0.



Each license is also an automatically floating license on the network. By setting the option to a value of 0 you receive an exclusively local license.

4.3.1.2 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.

 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](#)⁴⁴⁵. 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.

4.3.1.3 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.

 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](#)³⁹⁴).

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.

 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](#)⁵⁵³).

4.3.1.4 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

 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

 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.

4.3.1.5 Leasing

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

 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](#) ▶ 394).

4.3.1.6 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.

 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.

4.3.1.7 Downgrade/Version Management

Downgrade

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

 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

 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.

4.3.1.8 Overflow

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

 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.

4.3.1.9 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.



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.



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

4.3.1.10 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.



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.

4.3.1.11 Machine-bound Licenses

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



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

4.3.1.12 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.



For license borrowing you require [prepared](#)³⁵² *CmContainer*, one at the server and one at the client side.

The licensee borrows and returns licenses using the "[Borrowing](#)"⁴²¹ tab in *CodeMeter Control Center*. In *CodeMeter WebAdmin* the license allocation is displayed. Moreover, here the number of borrowed licenses and the maximum borrowing period are customizable.

4.4 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.

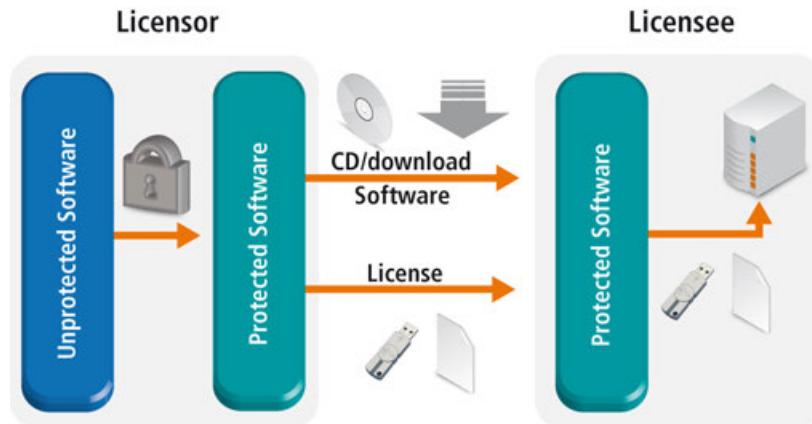


Figure 8: Security by Encryption

4.4.1 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".

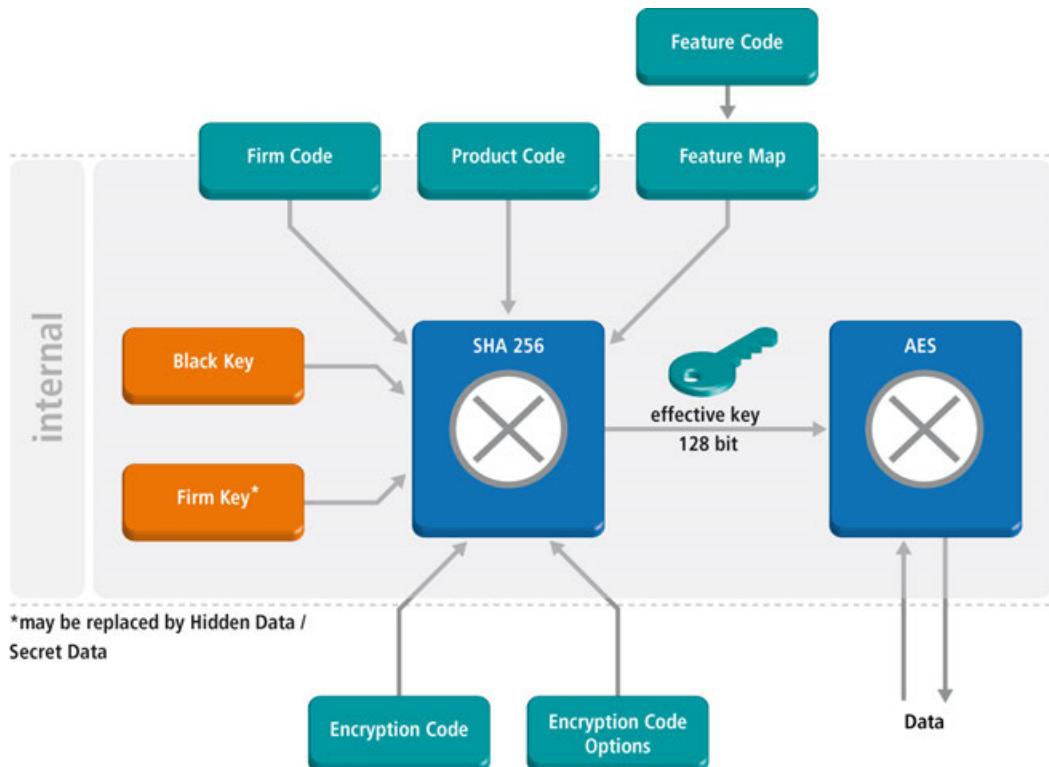


Figure 9: Key Derivation

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

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



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

Firm Code, Product Code, Feature Code

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

Encryption Code

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

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

Encryption Code Options

Moreover, the runtime Encryption Code Options complete the composition of the parameters. The Encryption Code Options contain information about required Product Item Options used in the key derivation and how they are checked. They cover the following options (for the detailed description of how to use the single options see in the *CodeMeter API Guide* online help the section "[Functions | Encryption API | CmCryt2](#)":)

- 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 are used by all *CodeMeter*® tools and interfaces when data is encrypted or decrypted.

4.5 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 128 bits in symmetric encryption and decryption operations.
Asymmetric encryption and decryption operations are executed by ECC (Elliptic Curve Cryptography) 224-bit and RSA with 2048-bit.

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

4.5.1 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.

4.5.2 Symmetric Encryption

In the case of symmetric cryptographic methods for encryption operations the same key is used (see [Encryption API](#)³¹⁰).

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.

There are two basic types of algorithms for symmetric encryption operations: stream cipher and block cipher.

Stream Cipher

In the case of a stream cipher, the plain text is not encrypted in blocks, but as digits one at a time. Each plain text digit is linked with a digit of the key.

4.5.2.1 Streamcipher (AES_STREAM)

In this mode, *CodeMeter®* uses the AES algorithm as a random number generator. The key stream is then combined with the plain text using the exclusive "or" operation (XOR).

This means that no separate decryption function is required since encryption and decryption are identical. Where a sequence, i.e. a key, is used a second time this mode is insecure and can be hacked. A one-bit plain text change results in a one-bit change in the cipher text. Therefore this mode has the lowest error possibility. The technical literature also describes this mode as "output feedback mode".

Block Cipher

Block ciphers take as input a block of plain text and a key, and output a block of cipher text of the same size.

4.5.2.2 Electronic Codebook Mode (AES_ECB)

The AES algorithm in Electronic Code Book mode divides the plain text into blocks and each block is encrypted separately. The disadvantage of this method is that identical plain text blocks are encrypted into identical cipher text blocks; thus, it does not hide data patterns well.



Unless there are pressing reasons, this mode is not recommended for use in cryptographic protocols at all.

4.5.2.3 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.

4.5.2.4 AES - Cipher Feedback Mode (CFB)

The AES algorithm in Cipher Feedback Mode encrypts plain texts longer than the allowed block length, i.e. data does not need to be padded to a multiple of the cipher block size. However, decryption is possible only from cipher text start. For encryption and decryption the same initialization vector has to be stated. Compared to other modes, CFB is considerably slower since for each byte a separate call for the encryption is started.



This mode should be used in special cases only.

4.5.3 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](#)³¹⁰, [Encryption API](#)³¹⁰, [Blocks](#)³¹⁶

4.5.3.1 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.

4.5.3.2 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.

4.5.3.3 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](#)³¹⁰ needed for performing the authentication procedures.

4.5.3.4 RSA

The RSA algorithm is named after its inventors (Ron **Rivest**, Adi **Shamir** and Leonard **Adleman**) and is suitable for signing as well as encryption.

4.5.4 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, and 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**" system menu. The user interface is divided into two areas: an upper menu bar, and a lower display window, allowing access to single applications.



Figure 10: *CodeMeter Start Center*

5.1.1 Menu Bar

File Menu

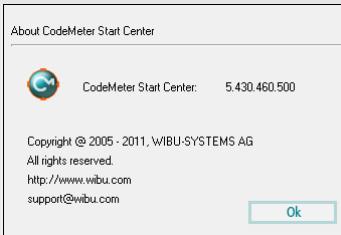
| Element | Description |
|----------|---|
| Language | <i>CodeMeter Start Center</i> provides several language settings for the user interface. Current- |

| Element | Description |
|-------------|---|
| | ly, you may choose from eight languages: Chinese, German, English, Spanish, French, Japanese, Dutch, and Portuguese. |
| Exit | The "Exit" menu item closes <i>CodeMeter Start Center</i> . Alternatively, you may close the window via the "x" control or the "Exit" button in <i>CodeMeter Start Center</i> . |

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's 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 ⁶⁸ 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 ³¹³ provides a way for you to integrate software protection into your source code. It can generate source code for you and explains the <i>CodeMeter Core API</i> and the Software Protection API (WUPI) ²⁹⁶ . |
| CodeMeter License Editor | CodeMeter License Editor ³²² is an intuitive graphical tool for creating, editing and deleting licenses in <i>CmDongles</i> . It supports locally connected <i>CmDongles</i> and also file-based remote programming ³⁶⁸ . |
| Samples | Finds samples for various programming languages explaining how to integrate the interfaces (<i>Software Protection API (WUPI)</i> , <i>Core API</i>). Click the button to open the related directories and get a short overview of existing examples. |
| Exit | Use the "Exit" button to close <i>CodeMeter Start Center</i> . |

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](#)⁴⁸²) 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, 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](#).

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](#)⁴⁶² licenses again.

CodeMeter Control Center and CodeMeter WebAdmin

Set local configurations for *CodeMeter License Server* in [CodeMeter Control Center](#)⁴¹¹. And [CodeMeter WebAdmin](#)⁴³⁵ 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.

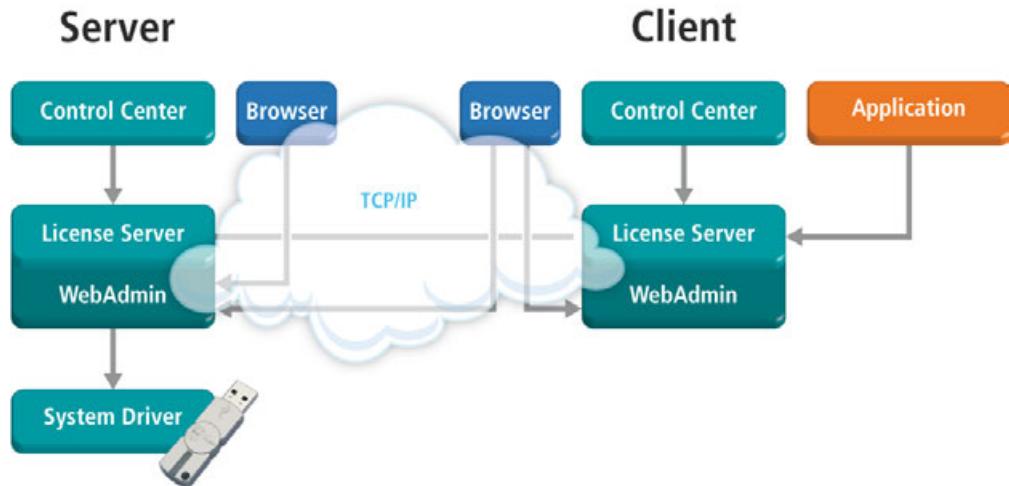


Figure 11: *CodeMeter License Server*

Diversity of Operating Systems

CodeMeter License Server is available for these operating systems: Windows 7, 8, Windows Vista, Windows XP, Windows CE, Windows Embedded, Mac OS X, Linux (different 32 and 64-bit versions), Sun Solaris 10, and VxWorks.

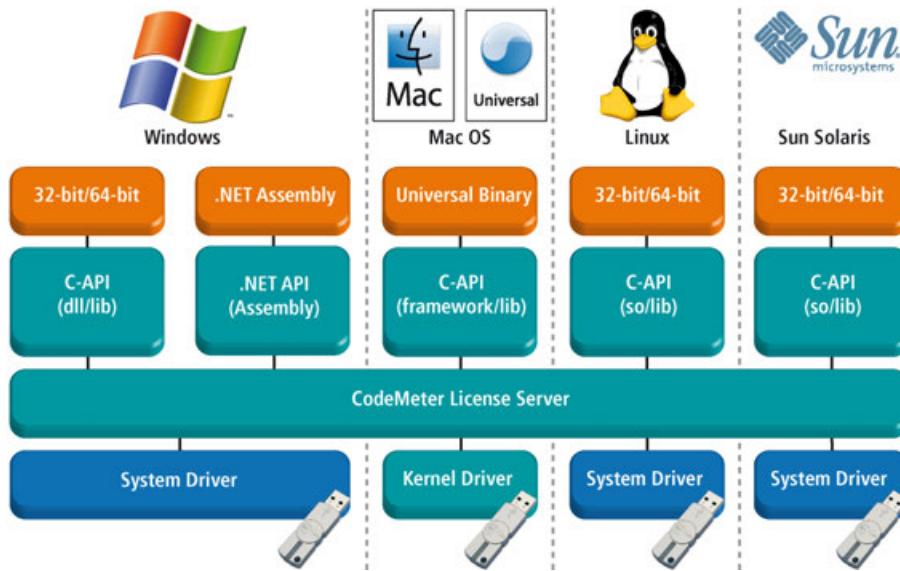


Figure 12: *CodeMeter License Server* and Operating Systems

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 Mac OS in the same network.

7 Automatic Software Protection using AxProtector

No Programming Skills required

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

In just a few minutes, *AxProtector* encrypts and protects your application for a variety of [project types](#)  ⁷².

AxProtector is also available as a [commandline variant](#)  ²⁷⁰ for Windows 32-bit / 64-bit, .NET, Linux, Mac OS, 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.

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

| Application to be protected | Project type | GUI Windows | Commandline |
|--|--|-------------|---|
| Windows Application or DLL |  AxProtector Windows ⁷³  IxProtector Windows ¹⁹⁹ | ✓ | Windows commandline |
| .NET Assembly |  AxProtector .NET ¹⁰⁴  IxProtector .NET ²¹² | ✓ | .NET commandline |
| Mac OS X Application or Dylib |  AxProtector Mac OS X ¹³⁰  IxProtector Mac OS X ²²⁵ | ✓ | Windows commandline Commandline available for Mac OS X (runs on Mac OS X operating systems) |
| Java Application (Archive Format *.jar, Webarchive Format *.war) |  AxProtector Java ¹⁵⁴ | ✓ | Windows commandline Commandline available for Java (runs on Windows, Mac Os X, Linux, and Solaris operating systems) |
| Linux Application or Shared Object |  AxProtector Linux ¹⁷³  IxProtector Linux ²³⁸ | ✓ | Windows commandline Commandline available for Linux (runs on Linux operating systems) |
| Files your protected application uses |  AxProtector File Encryption ²⁵³ | ✓ | 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](#)⁶³ or, alternatively, by the "**Start | All Programs | AxProtector**" system menu item.

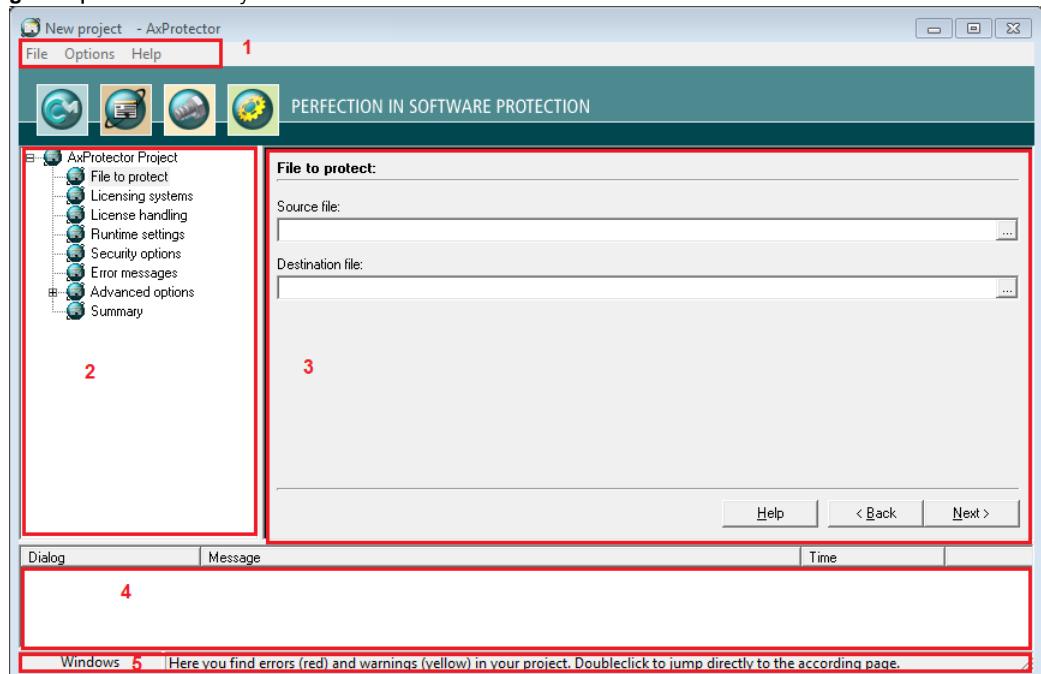


Figure 13: AxProtector – GUI and Navigation

The AxProtector GUI consists of five separate areas:

- [Menu bar](#)⁷⁰ (1)
- [Navigation window](#)⁷¹ (2)
- [Input window](#)⁷¹ (3)
- [Note and error window](#)⁷¹ (4)
- [Project type area](#)⁷¹ (5)

7.1.1 Menu Bar

File menu

| Element | Description |
|-----------------|--|
| Project | <p>New Project To create a new project, please proceed as follows:</p> <ol style="list-style-type: none"> Select the "File New Project" menu item. Alternatively, press the <CTRL+N> key combination. The "New Project" dialog opens for selecting the project type. <p>Open Project To open an existing project, please proceed as follows:</p> <ol style="list-style-type: none"> 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. Select the project file name to be opened, and click the "Open" button. <p>Save Project To save a created or edited project, please proceed as follows:</p> <ol style="list-style-type: none"> Select the "File Save Project" menu item. Alternatively, press the <CTRL+S> key combination. <p>Save Project as To save an opened project using another project name, please proceed as follows:</p> <ol style="list-style-type: none"> Select the "File Save Project as" menu item. Select a destination folder in the "Save as" window and specify the new name of the project file. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  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. </div> |
| Export Wbc file | 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 ²⁷⁰ . |
| | <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  This menu item is active only after the project has passed all necessary checks. </div> |
| 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, Dutch, and Portuguese. |

? menu

| Element | Description |
|---------|--|
| Content | Select this menu item to open the <i>AxProtector</i> online help. |
| About | Select this menu item to open a window holding <i>AxProtector</i> 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 will be executed. |



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

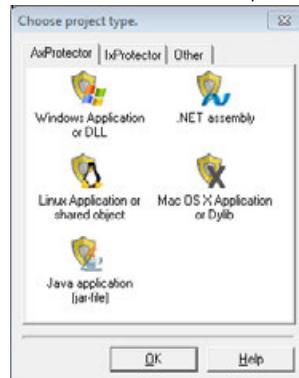
7.1.5 Project type area

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 73 |
| | .NET Assembly 104 |
| | Mac OS X Application or Dylib 130 |
| | Java Application (jar file) 154 |
| | Linux Application or Shared Object 173 |
| IxProtector | |
| | Windows Application or DLL 199 |
| | .NET Assembly 212 |
| | Linux Application or Shared Object 238 |
| | Mac OS X Application or Dylib |
| Other | |

| Icon | Project type |
|---|--|
|  | File encryption <small>253</small> |

7.4 AxProtector Tab

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 <small>73</small> | ✓ | Windows commandline <small>270</small> |

7.4.1.1 File to protect

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

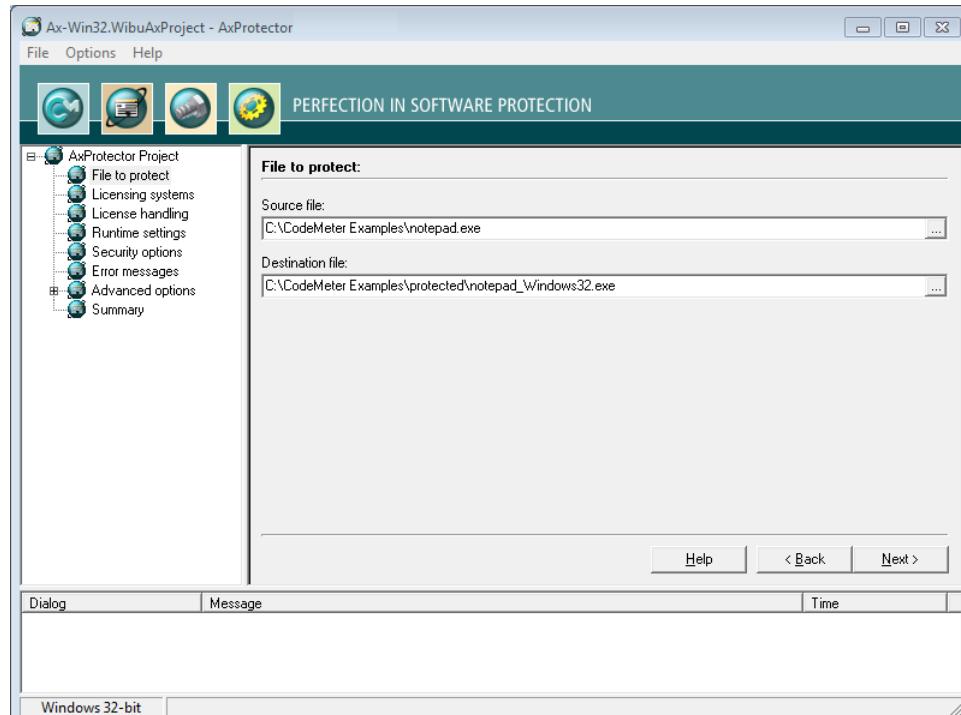


Figure 14: AxProtector - Windows "File to protect"

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. i 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, <i>AxProtector</i> 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 ²⁸⁹ . |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

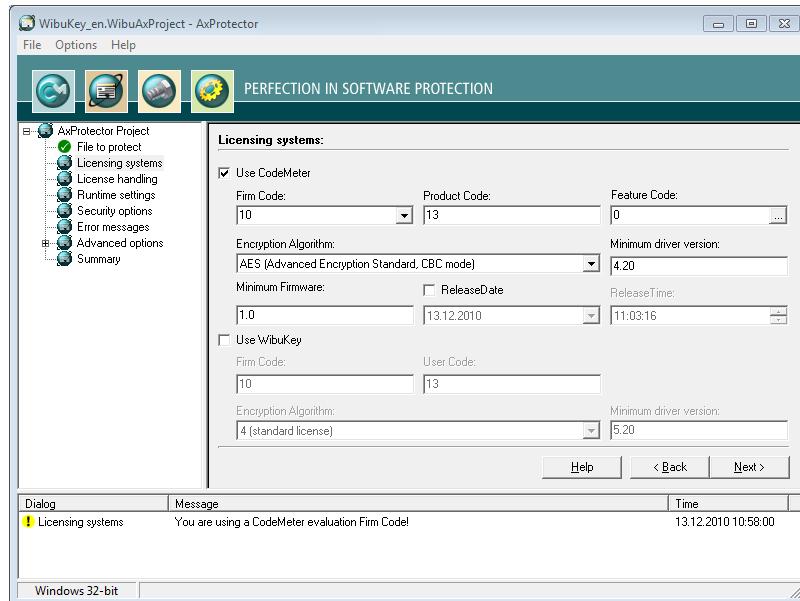


Figure 15: AxProtector - Windows "Licensing Systems"

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.

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|--------------|---|
| Firm Code | <p>Specify the Firm Code to be used for encrypting the software.</p> <p>The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation <i>Firm Code</i> found in the <i>CodeMeter® Software Development Kit (SDK)</i>. In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code..The test Firm Code for <i>CmActLicense</i> is 5010.</p> <p>As a registered licensor, you will be issued your own unique Firm Code(s).</p> <p>Commandline option see here²⁷¹.</p> |
| Product Code | Enter the Product Code which defines the encryption of a specific product. You can freely |

| Element | Description |
|------------------------|--|
| | <p>choose this identifier, e.g. for a separate module of a software application, or for a single application. Commandline option see here²⁷¹.</p> |
| Feature Code | <p>Enter the Feature Code which defines, for example, the encryption of different software versions.</p> |
| | <p> 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.</p> <p>Using the "..." button you may enter the feature map value in hexadecimal, decimal or binary format.</p>  <p>Figure 16: AxProtector - Windows Feature Map Input Commandline option see here²⁷².</p> |
| Encryption Algorithm | <p>Select the algorithm to encrypt your software. Currently, CodeMeter® solely supports AES (Advanced Encryption Standard).</p> |
| Minimum Driver Version | <p>Enter the minimum driver version required for the installed CodeMeter License Servers. 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.</p> |
| | <p> 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.</p> |
| Release Date | <p>Starting with Firmware version 1.18 CodeMeter® supports the Product Item Option Maintenance Period⁴⁵.</p> |
| Minimum Firmware | <p>Specify the minimum firmware version required. In order to use the Product Item Option Maintenance Period you require the firmware version 1.18. Commandline option see here²⁷².</p> |

WibuKey

For setting WibuKey options, see the separate "WibuKey Developer Guide".

7.4.1.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network or both. Moreover, you can define the license allocation (access) mode.

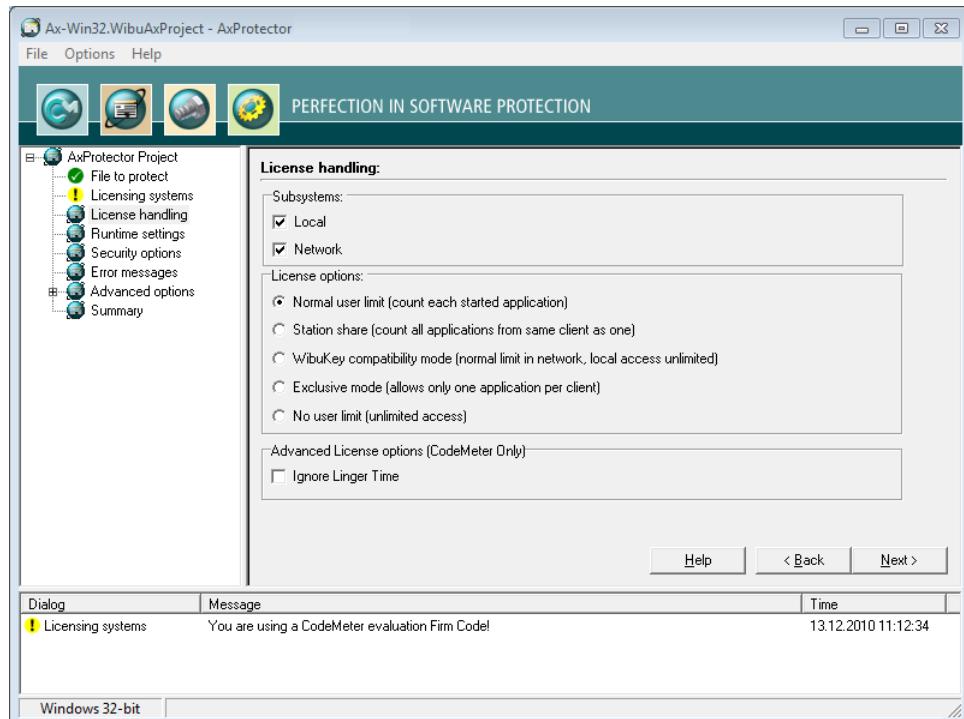


Figure 17: AxProtector - Windows "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 <i>CodeMeter License Server</i> runs and is activated as network server. i On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

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](#)²⁷³).

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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. WibuKey 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. |

Linger Time

| Element | Description |
|--------------------|--|
| Ignore Linger Time | Activate this option to ignore a programmed LingerTime. This license option allows to define an allocation time of the license after the license of a protected application has been released or the protected application has been closed. |

7.4.1.4 Runtime Settings

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

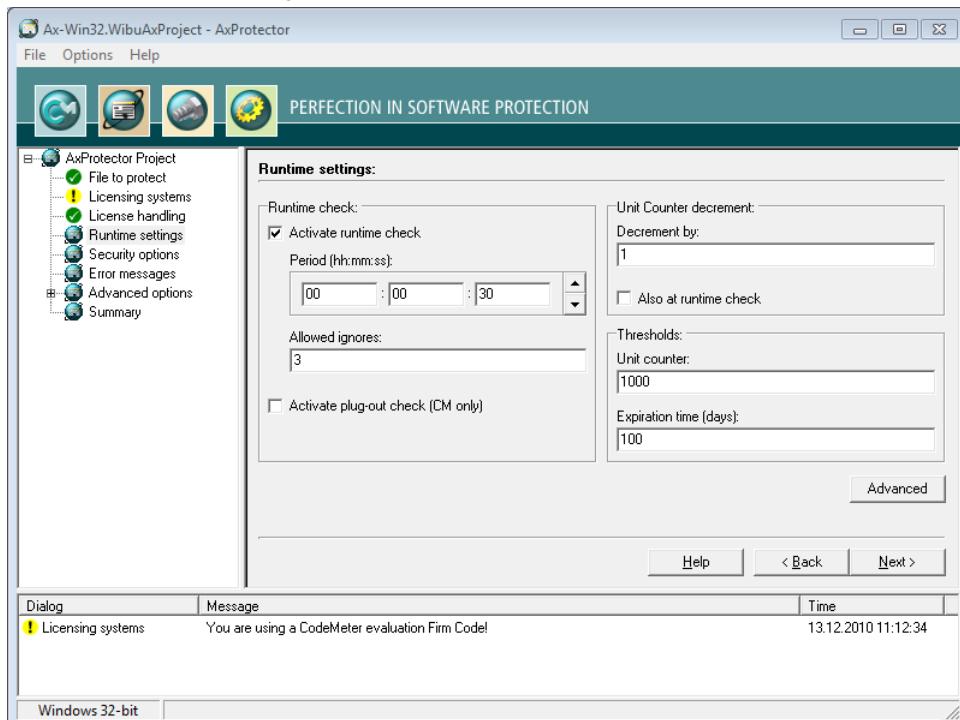


Figure 18: AxProtector - Windows "Runtime Settings"

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 ²⁷⁷ . |
| 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 <i>CmContainer</i> 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 | This option closes the protected application when the <i>CmDongle</i> is removed while the application is running. Immediately, an error message is issued. This option is valid for <i>CmDongle</i> only. |

| Element | Description |
|-----------------|--|
| (only CmDongle) | Commandline option see here ²⁷⁵ . |

Unit Counter Decrement

Decrementing an Unit Counter can serve to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see [here](#)²⁸⁴).

| 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 ⁷⁹ " 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 ⁸⁹ . |
|--|---|

| Element | Description |
|------------------------|---|
| Unit Counter | If the defined threshold falls short, a warning message is issued. Commandline option see here ²⁸⁶ . |
| 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 ²⁸⁵ . |

7.4.1.4.1 Advanced Runtime Settings

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

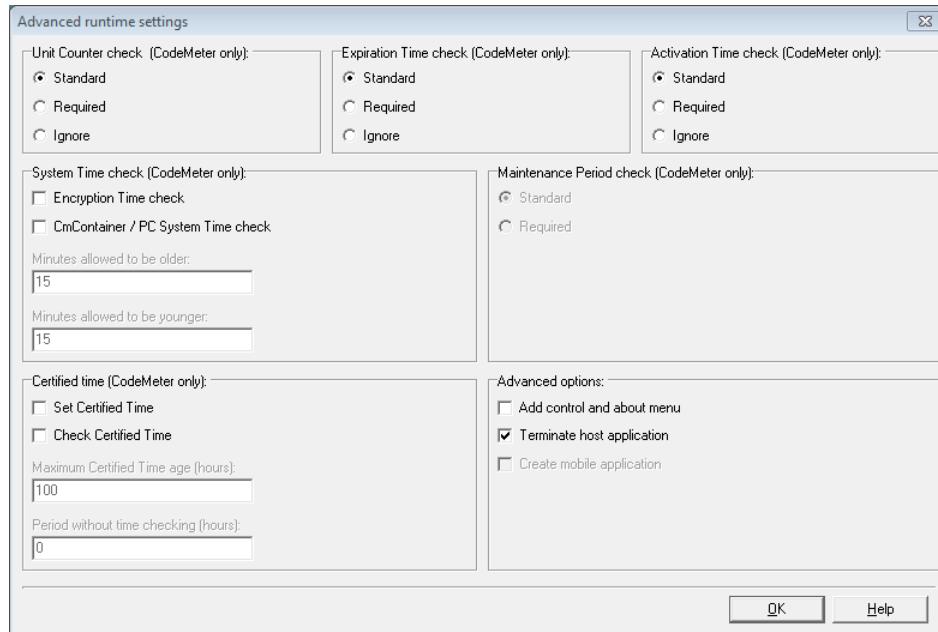


Figure 19: AxProtector - Windows "Advanced Runtime Settings"

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](#)²⁸⁴).

| Element | Description |
|----------|---|
| Standard | Decrement 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 |

| Element | Description |
|---------|---|
| | 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](#)²⁸⁴).

| 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](#)²⁸³).

| 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 ³⁹⁴ 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](#)²⁸⁴).

| |
|---|
|  The option is available only, if you activated the checkbox Release Date on the page "Licensing systems" ⁷⁶ . |
|---|

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](#)²⁷⁸).

 For information on the fail safe and manipulation safe processes referring to *Activation and Expiration Time* see [here](#)²⁹⁴ ..

| Element | Description |
|--------------------------------------|---|
| Set Certified Time | This option attempts to update the <i>Certified Time</i> in a <i>CmDongle</i> . The <i>Certified Time</i> is requested from the Time Server.  This option requires a connection to the Internet. |
| Check Certified Time | This option checks to see if the <i>Certified Time</i> 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 <i>Certified Time</i> . |
| Period without time checking (hours) | Specifies the period (in hours) when no check of the <i>Certified Time</i> certificate is performed. If this period is not reached, a check is not performed. If the <i>Certified Time</i> certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the <i>Certified Time</i> 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 <i>Certified Time</i> 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](#)²⁷⁴).

| 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 <i>CmContainer</i> System Time is newer than the encryption time.  Requires at least <i>CodeMeter® 4.10</i> . |
| CmContainer / PC System Time check | If activated, these options define a time corridor in which a difference between <i>CmContainer</i> 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 <i>CmContainer</i> System Time. |
| Minutes to be allowed younger | States in minutes how much PC Time is allowed to be younger than the <i>CmContainer</i> 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 ²⁷⁷). |
| 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 ²⁸⁶). |
| Create mobile application | [not yet implemented] |

7.4.1.5 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, 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.

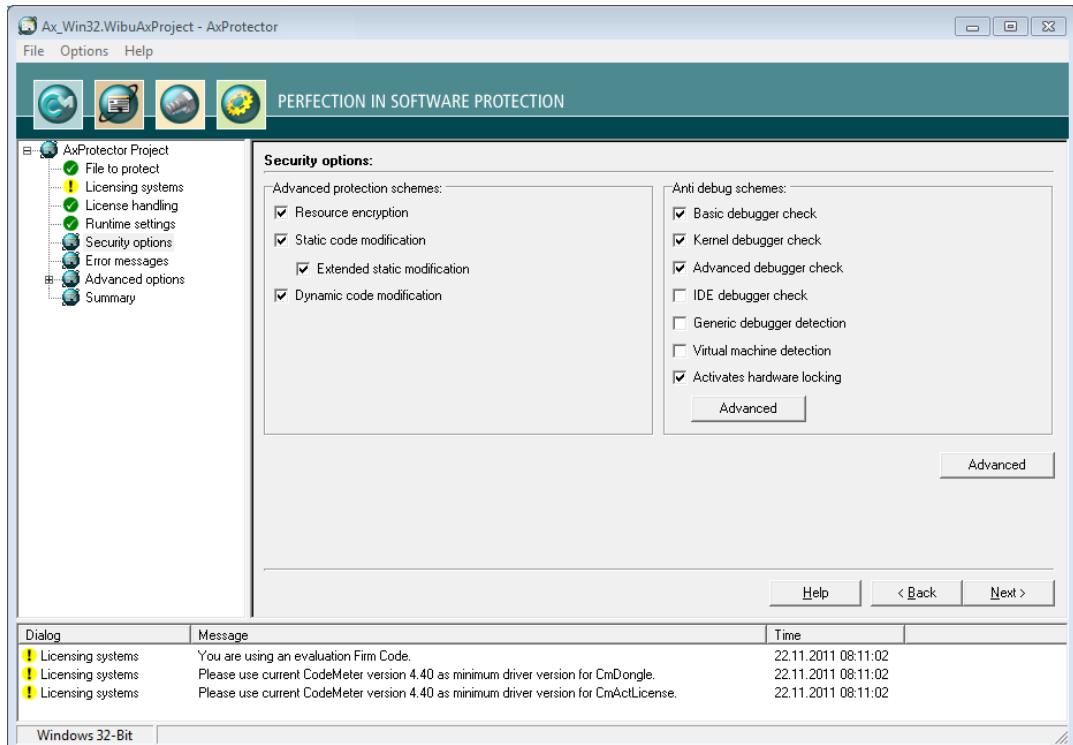


Figure 20: AxProtector - Windows "Security Options"

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](#) 274).

| 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 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. |
| Dynamic Code Modification | The source code of the application to be protected is modified dynamically <u>at runtime</u> of the application. |



The options "Static Code Modification" and "Extended Static Modification" conflict with an activated option "[Activate Automatic File Encryption](#)"⁹¹ 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](#)²⁷⁵).

| 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 <i>CmContainer</i> 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. |
| 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. |



This button is activated only for *CodeMeter*.

Locking Scenario

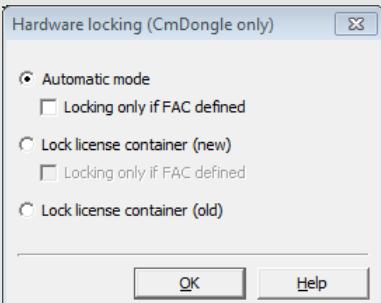
Description

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. 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.

| Element | Description |
|--|--|
| Locking Scenario | Description |
| | <p>The owner / end-user of the locked Firm Items must contact the software vendor for unlocking codes. This can be done by remote programming.</p>  |
| <p align="center">Figure 21: AxProtector - Windows "Security Options - Hardware Locking"</p> | |
| The following settings are available: | |
| Option | Description |
| 'Automatic Mode' activated and "Locking only if FAC defined" not activated (Standard) | <p>If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1.</p> <p>If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>Due to compatibility reasons this corresponds to the default setting.</p> |
| 'Automatic Mode' activated and "Locking only if FAC defined" activated | <p>If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1.</p> <p>If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked.</p> |
| 'Lock License Container (new)' activated and "Locking only if FAC defined" not activated | <p>If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher.</p> |
| 'Lock License Container (new)' and "Locking only if FAC defined" activated | <p>If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked.</p> |
| 'Lock License Container (old)' activated | <p>For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1.</p> |

7.4.1.5.1 Advanced Security Options

This input window lets you define further settings.



Figure 22: AxProtector - Windows "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

| Element | Description |
|------------------------------------|--|
| Add virus check | Adds a virus check to the protected application by using a check sum (commandline option see here ²⁷⁸). |
| Link API statically to Application | The <i>CodeMeter Core API</i> is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here ²⁷⁹). |
| Size of encrypted Code (in %) | Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here ²⁷⁸). |

7.4.1.6 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.

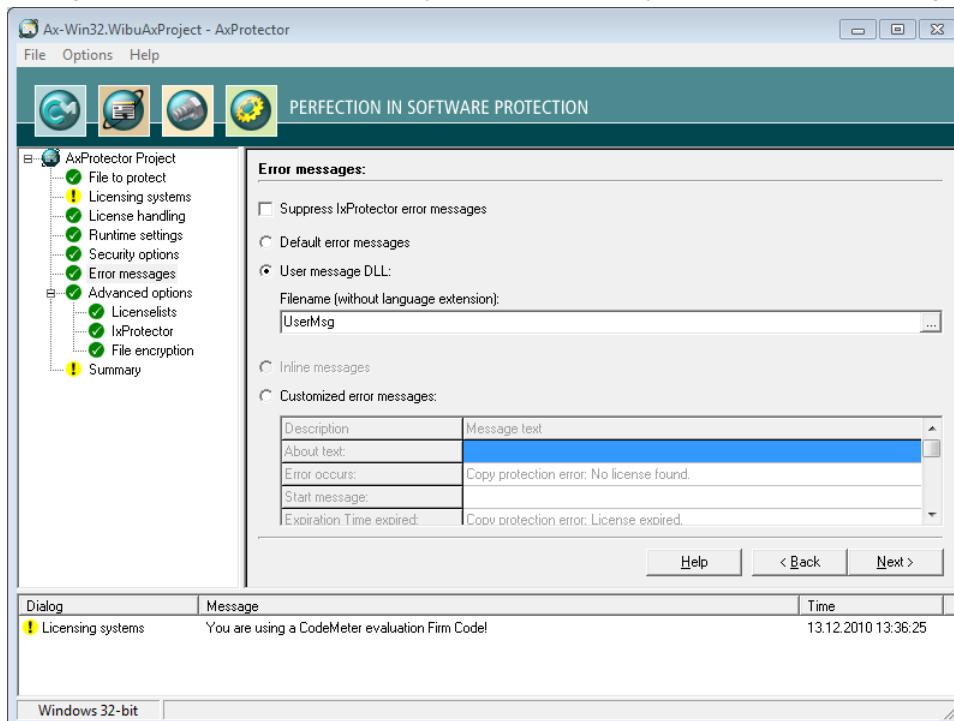
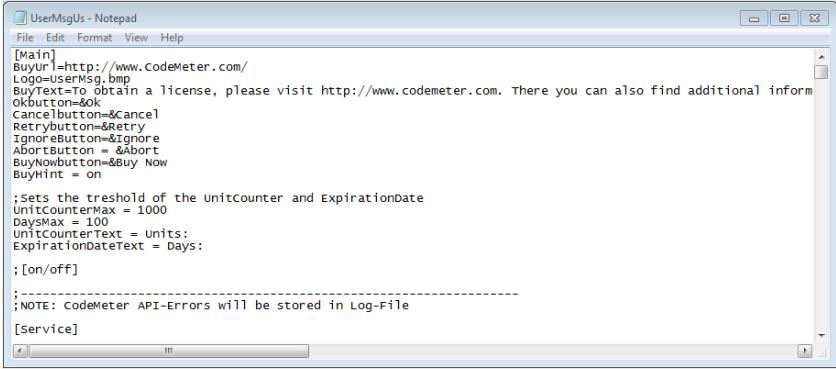


Figure 23: AxProtector - Windows "Error Messages"

Error Messages

| Element | Description |
|-------------------------------------|--|
| Suppress IxProtector Error Messages | The output of IxProtector error messages is suppressed (commandline option see hier ²⁸¹). <div style="border: 1px solid #ccc; padding: 5px; margin-left: 20px;">  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. </div> |
| Default Error Messages | All errors occurring at the runtime of a protected application display default error messages (commandline option see hier ²⁸⁷). |
| 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 hier ²⁸⁸). |

| Element | Description |
|---------------------------|---|
| | <p>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.</p>  <pre data-bbox="293 370 1103 699"> [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 threshold 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] </pre> |
| | <p>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.</p> |
| Inline Messages | <p>Links for .NET projects, with an inline assembly which can also be configured by *.ini files.</p> <p> This option is available for the encryption of .NET applications only.</p> |
| Customized Error Messages | <p>Activate this option to enter customized error messages displayed in the message boxes below.</p> |

7.4.1.7 Advanced Options

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

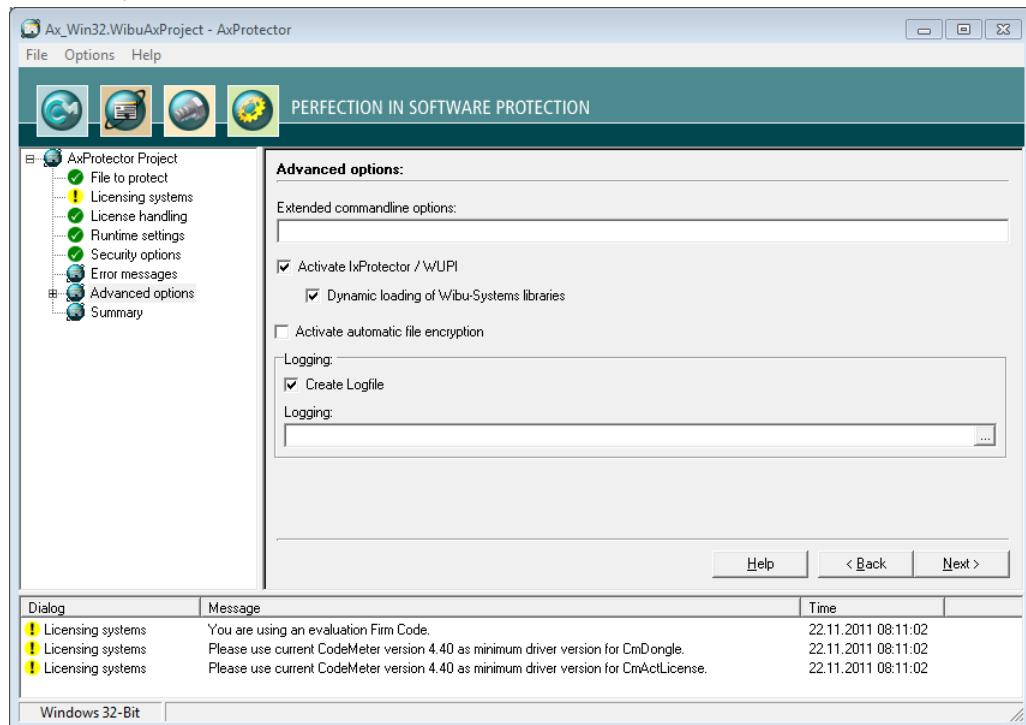


Figure 25: AxProtector - Windows "Advanced Options"

| Element | Description |
|---|--|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the <i>AxProtector</i> 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 <i>IxProtector</i> via the Software Protection-API ²⁹⁶ . (command option see here ²⁹¹). |
| 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 ²⁹¹). |
| Activate Automatic File Encryption | Activate this checkbox to trigger the automatic decryption of files by the <i>AxProtector</i> engine (command option see here ²⁷⁴). |

| Element | Description |
|----------------|---|
| | 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. |

7.4.1.7.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *lxProtector* via the [Software Protection-API \(WUPI\)](#)²²⁶.

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](#)²²⁷.

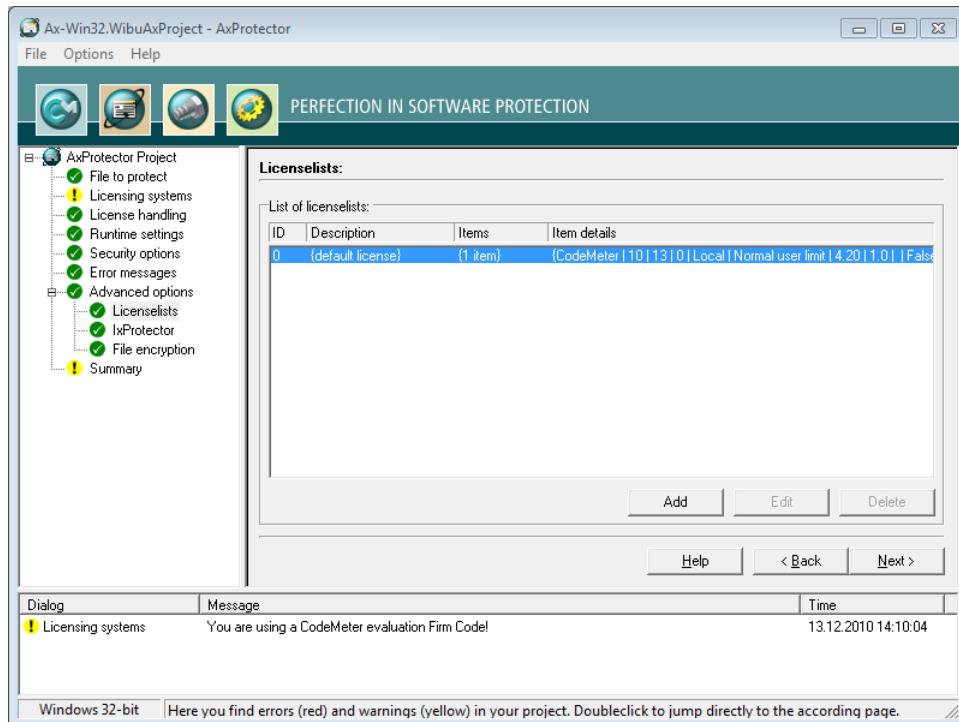
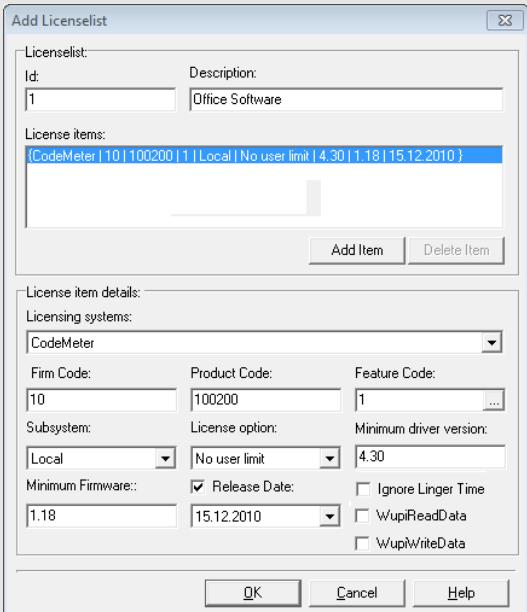
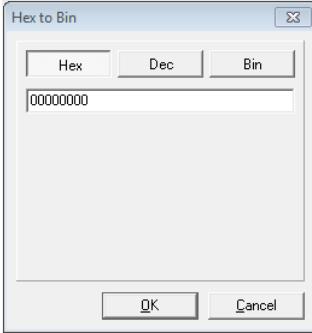


Figure 26: AxProtector - Windows "License Lists"

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 | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p>i 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.</p> |
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p> |

| Element | Description |
|-------------------|--|
| |  <p>The screenshot shows the 'Add License List' dialog box. At the top, there's a section for 'License list' with fields for 'Id' (set to 1) and 'Description' (set to 'Office Software'). Below this is a list titled 'License items' containing a single item: '[CodeMeter 10 100200 1 Local No user limit 4.30 1.18 15.12.2010]'. There are 'Add Item' and 'Delete Item' buttons at the bottom of this list. Below the list, there's a section for 'License item details' with fields for 'Licensing systems' (set to 'CodeMeter'), 'Firm Code' (set to '10'), 'Product Code' (set to '100200'), 'Feature Code' (set to '1'), 'Subsystem' (set to 'Local'), 'License option' (set to 'No user limit'), 'Minimum driver version' (set to '4.30'), 'Minimum Firmware' (set to '1.18'), 'Release Date' (set to '15.12.2010'), and checkboxes for 'Ignore Linger Time', 'WupiReadData', and 'WupiWriteData'. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons.</p> |
| | Figure 27: AxProtector - Windows "Add License Lists" |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). |
| 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. Using the "... button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. |
| |  <p>The screenshot shows the 'Hex to Bin' dialog box. It has tabs for 'Hex', 'Dec', and 'Bin', with 'Hex' selected. A text input field contains the value '00000000'. At the bottom are 'OK' and 'Cancel' buttons.</p> |

| Element | Description |
|------------------------|--|
| 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: <ul style="list-style-type: none"> • 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 <i>CodeMeter</i> ® 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: <ol style="list-style-type: none"> 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 finished (more information in the <i>CodeMeter</i> Developer Guide). |
| WupiReadData | Activate this option to read data from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data into a <i>CmContainer</i> 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.

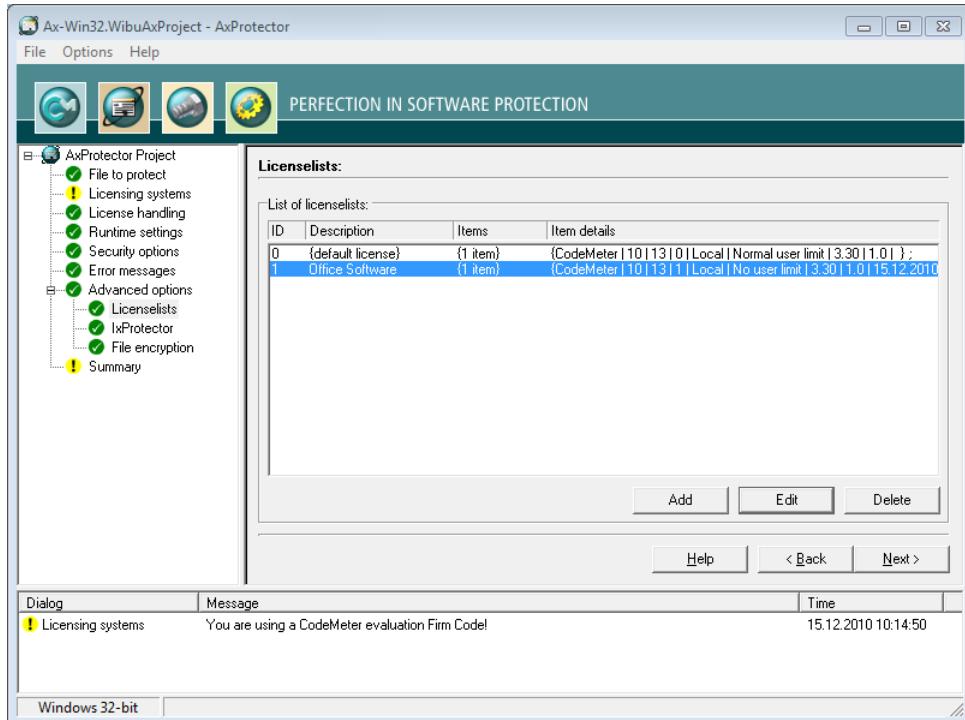


Figure 28: AxProtector - Windows "Completed License List"

7.4.1.7.2 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.

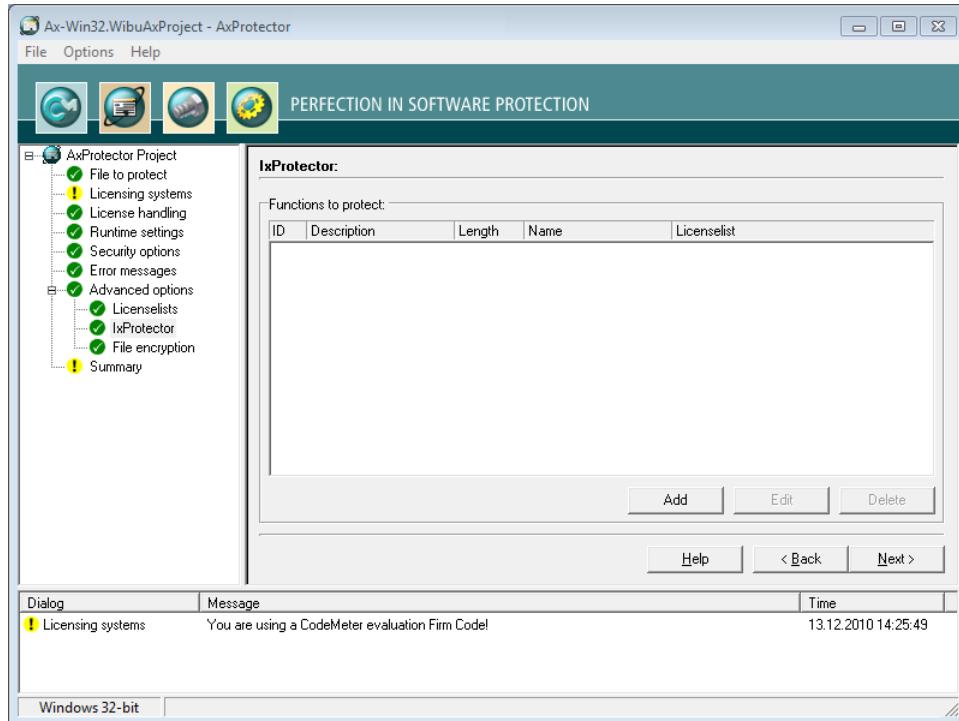
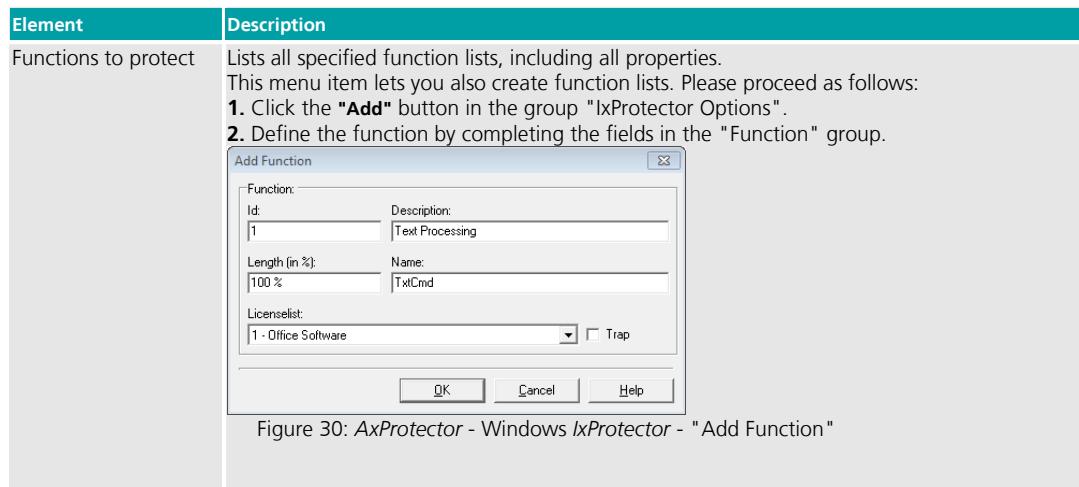


Figure 29: AxProtector - Windows IxProtector - "Function List"



| Element | Description | |
|--------------|-------------|--|
| | Element | Description |
| Id | | <p>Uniquely identifies the function.</p> <p> This Id corresponds to the identification you use when calling the WUPI commands WupiDecryptCode ²⁹⁸ and WupiEncryptCode ²⁹⁸.</p> |
| Description | | <p>Enter a description of the function with text.</p> |
| Length | | <p>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.</p> <p> If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.</p> |
| Name | | <p>Specify the name of the function to be encrypted.</p> <p>The function name must exactly match the name used in the export list of the linked map file.</p> <p>Please note the correct spelling (case sensitive, underline, etc.). Microsoft Dependency Walker shows dependencies between 32-or 64-bit Windows PE files. A tree view shows all linked modules and imported and exported functions are displayed in tables. Dependency Walker is part of Windows XP SP2 Support Tools and also part of Microsoft Visual Studio including version 8.0 (Visual Studio 2008, i.e. version 9.0 no longer provides Dependency Walker).</p> |
| License List | | <p>Selects an existing license to which the function is assigned. Then this license list is used for the encryption of the function.</p> |
| Trap | | <p>Activates the trap function for the function. Command line option see here ²⁹⁵.</p> |

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

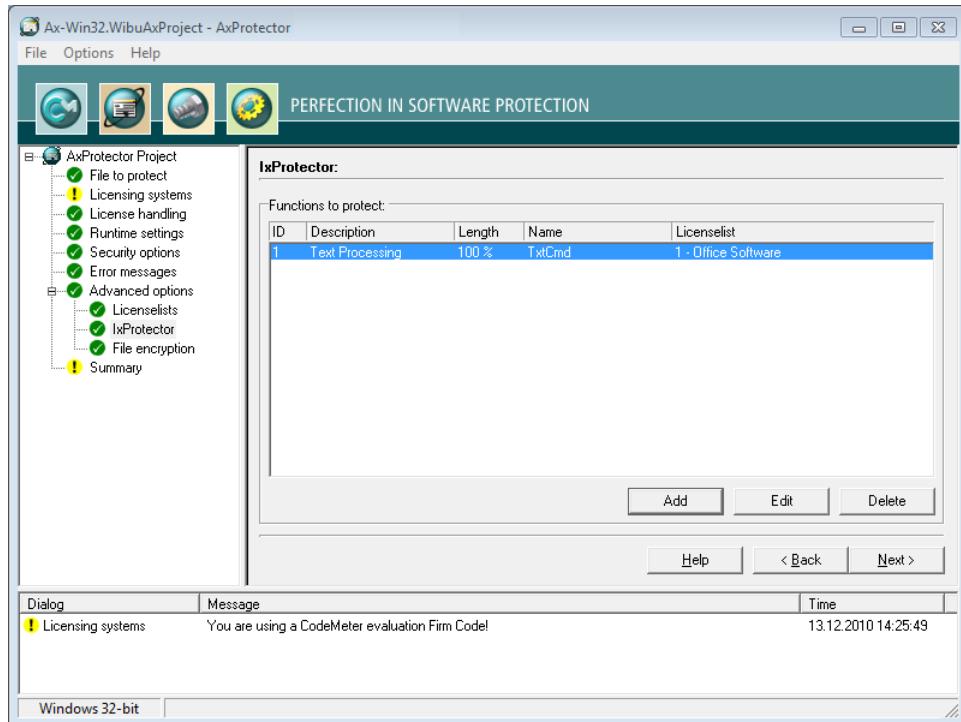


Figure 31: AxProtector - Windows IxProtector - "Completed Function List"

7.4.1.7.3 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.

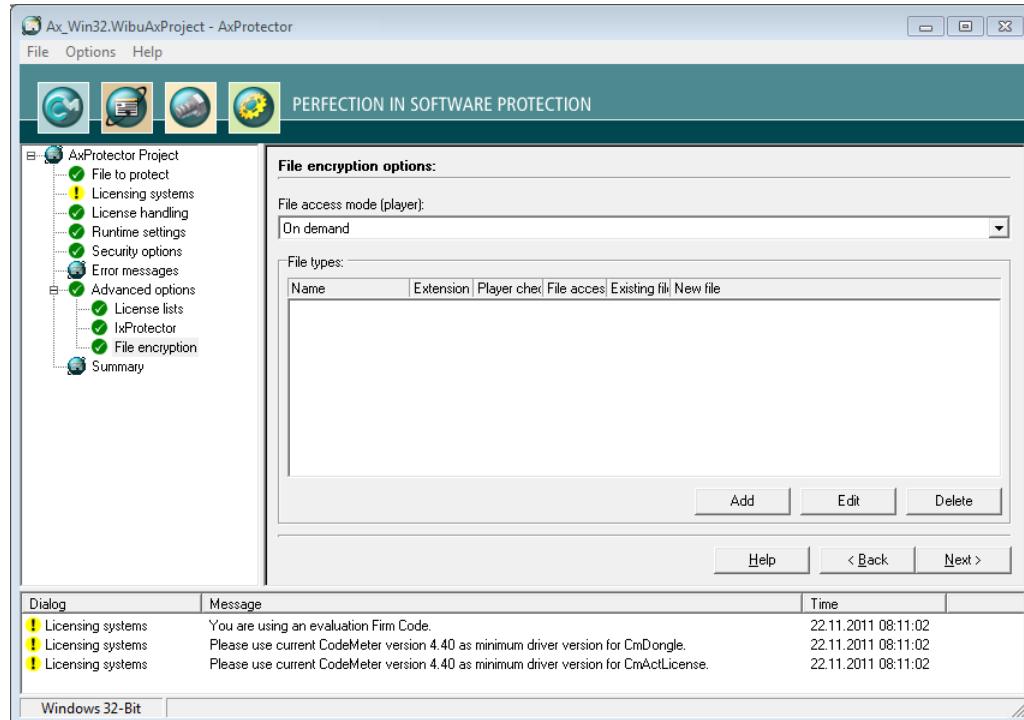


Figure 32: AxProtector - Windows "File Encryption"

| Element | Description |
|---------------|---|
| Add File Type | <p>1. Click on the "Add" button to add a new file type.</p> |
| | <p>2. Enter in the "Name" field a describing descriptive name for the file type. This name has no impact on the encryption.</p> <p>3. Enter in the "Extension" field the file extension of the file type you create, e.g. txt for text files.</p> <p>4. In the "Player Check" dropdown you define whether the license options of the accessing application</p> |

Figure 33: AxProtector - File Encryption "Add File Type"

| Element | Description | | | | | | | | | | |
|---|--|---|--|-----------------|---|---------|--|----------------|--|----|--|
| | <p>(player) are checked when the encryption takes place.</p> <table border="1"> <tr> <td>License list</td> <td>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. </td></tr> <tr> <td>No player check</td> <td>No check of the accessing application is performed.</td></tr> </table> | 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. | | | | | | |
| 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. | <p>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.</p> <table border="1"> <tr> <td></td> <td> <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> </td></tr> <tr> <td>On demand</td> <td> <p>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.</p> <p> 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.</p> </td></tr> <tr> <td>At once</td> <td> <p>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.</p> <p> 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.</p> </td></tr> <tr> <td>Huge file mode</td> <td> <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> 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.</p> </td></tr> <tr> <td>6.</td><td> <p>In the group "Write Options" define the settings on how changes are saved.</p> <p>Existing File In this group you define the settings on how changes to an existing file are saved.</p> </td></tr> </table> |  | <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> | On demand | <p>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.</p> <p> 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.</p> | At once | <p>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.</p> <p> 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.</p> | Huge file mode | <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> 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.</p> | 6. | <p>In the group "Write Options" define the settings on how changes are saved.</p> <p>Existing File In this group you define the settings on how changes to an existing file are saved.</p> |
|  | <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> | | | | | | | | | | |
| On demand | <p>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.</p> <p> 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.</p> | | | | | | | | | | |
| At once | <p>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.</p> <p> 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.</p> | | | | | | | | | | |
| Huge file mode | <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> 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.</p> | | | | | | | | | | |
| 6. | <p>In the group "Write Options" define the settings on how changes are saved.</p> <p>Existing File In this group you define the settings on how changes to an existing file are saved.</p> | | | | | | | | | | |

| Element | Description | |
|--|--------------|--|
| | 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. |

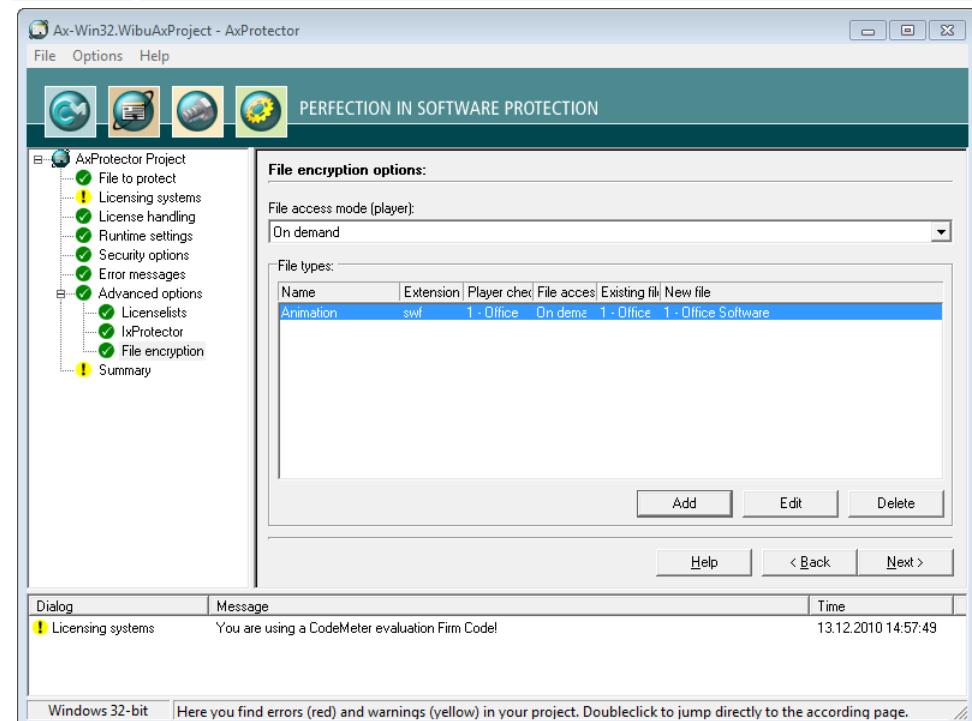


Figure 34: AxProtector - File Encryption "Completed Option list"

7.4.1.8 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](#)²⁶ type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

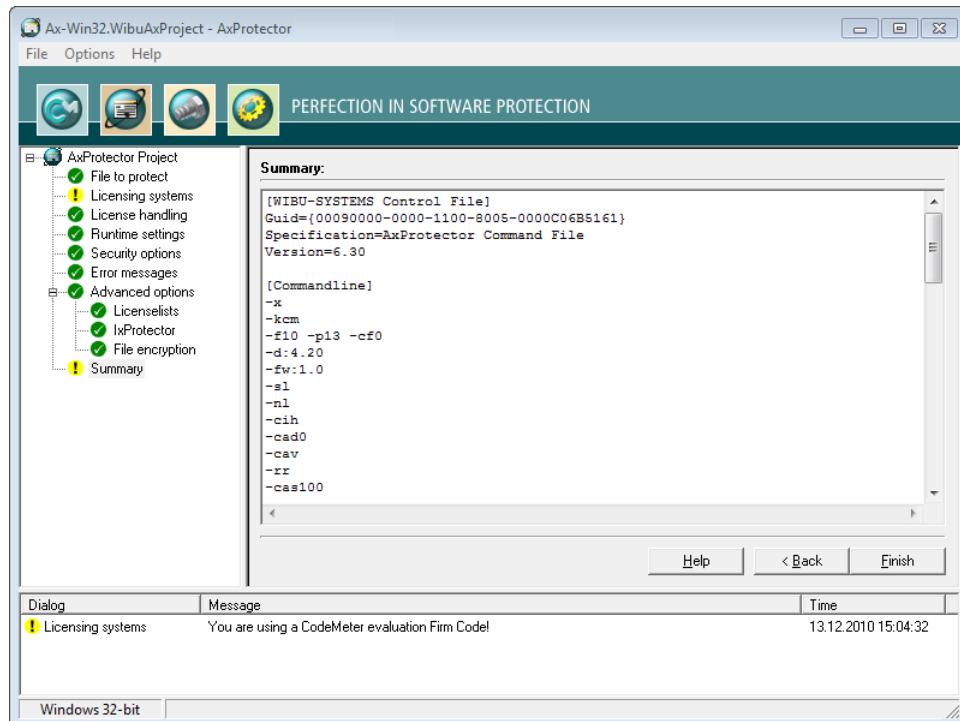


Figure 35: AxProtector - Windows "Summary"

| 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.

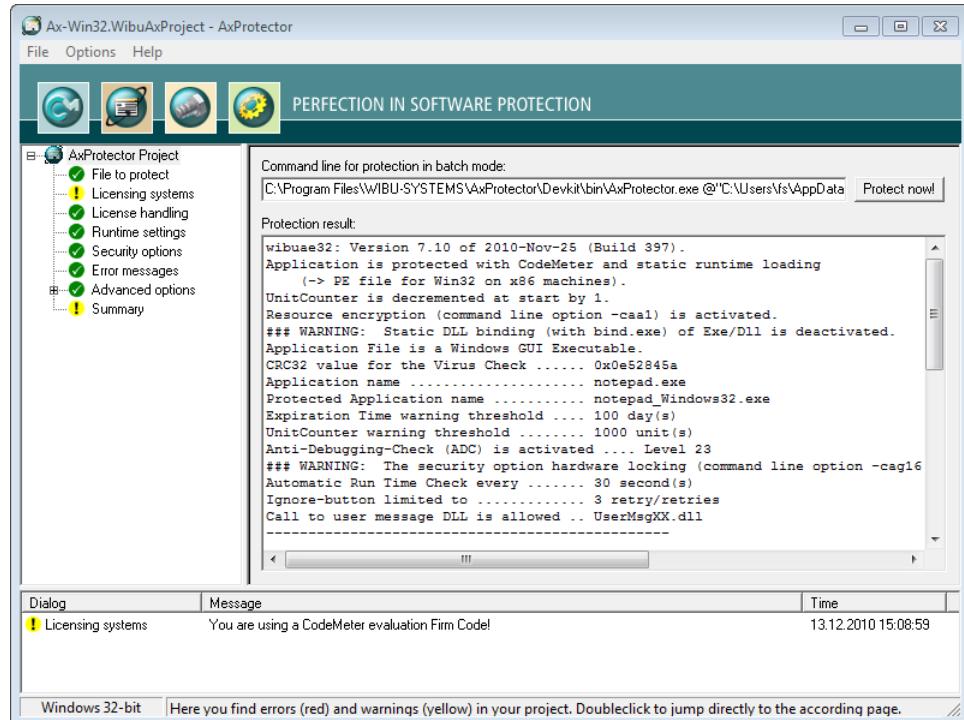


Figure 36: AxProtector - Windows "Encryption Result"

| Element | Description |
|-------------|--|
| Protect now | <p>When you need to repeat the encryption operation, click the "Protect now!" button. Then the AxProtector commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

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.

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 ¹⁰⁶ | ✓ | .NET commandline ²⁷⁰ |

i 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.

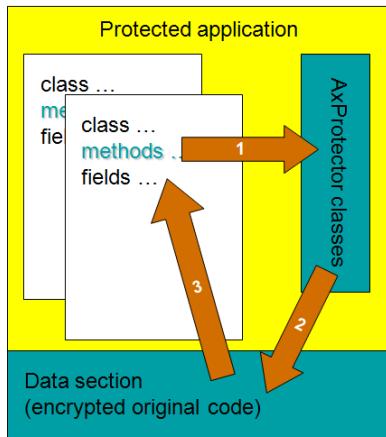


Figure 37: .NET encryption

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.

7.4.2.1 File to protect

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

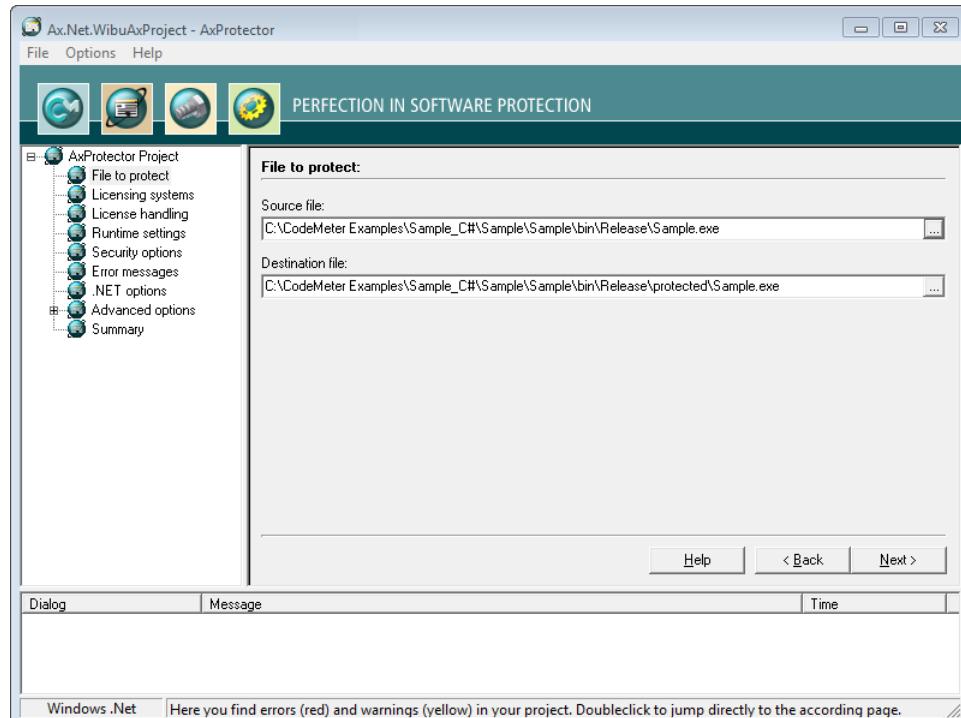


Figure 38: *AxProtector .NET - "File to Protect"*

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 | After you selected the source file, <i>AxProtector</i> automatically creates a secondary folder [...] \pro- |

| Element | Description |
|---------|--|
| File | ected\...]. 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 ²⁸⁹ . |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

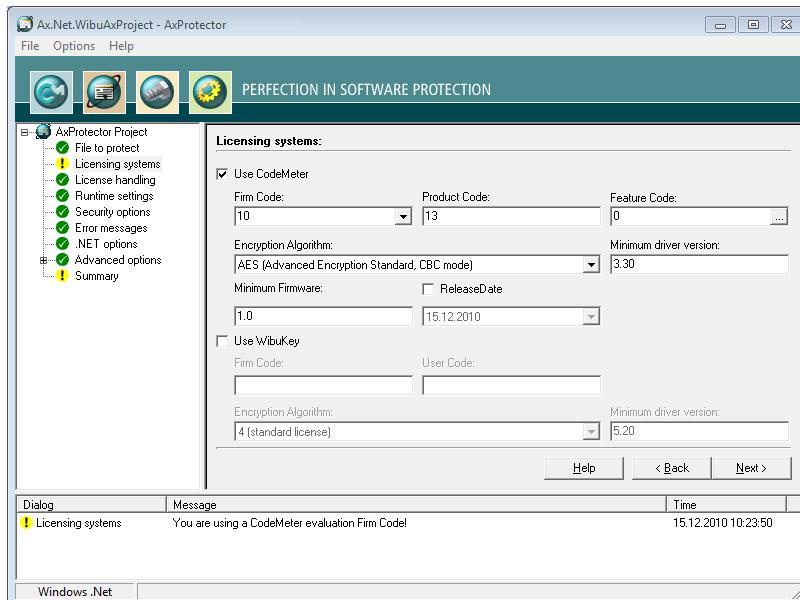
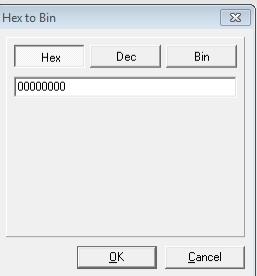


Figure 39: AxProtector .NET - "Licensing Systems"

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

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|-----------|---|
| Firm Code | Specify the Firm Code to be used for encrypting the software. |

| Element | Description |
|------------------------|---|
| | <p> The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation <i>Firm Code</i> found in the <i>CodeMeter® Software Development Kit (SDK)</i>. In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code..The test Firm Code for <i>CmActLicense</i> is 5010. As a registered licensor, you will be issued your own unique Firm Code(s).</p> |
| Product Code | <p>Commandline option see here²⁷¹.</p> <p>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.</p> <p>Commandline option see here²⁷¹.</p> |
| Feature Code | <p>Enter the Feature Code which defines, for example, the encryption of different software versions.</p> <p> 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.</p> <p>Using the "... " button you may enter the feature map value in hexadecimal, decimal or binary format.</p>  <p>Figure 40: <i>AxProtector</i> - Windows Feature Map Input</p> <p>Commandline option see here²⁷².</p> |
| Encryption Algorithm | <p>Select the algorithm to encrypt your software. Currently, <i>CodeMeter®</i> solely supports AES (Advanced Encryption Standard).</p> |
| Minimum Driver Version | <p>Enter the minimum driver version required for the installed <i>CodeMeter License Servers</i>.</p> |
| | <p>If setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that <i>AxProtector</i> automatically handles sessions of the protected software, and each session is allocated one of the available licenses.</p> <p> Setting the driver version is also required if, 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.</p> |
| Release Date | <p>Commandline option see here²⁷².</p> <p>Starting with Firmware version 1.18 <i>CodeMeter®</i> supports the Product Item Option Maintenance Period⁴⁵</p> |
| Minimum Firmware | <p>Specify the minimum firmware version required. In order to use the Product Item Option</p> |

| Element | Description |
|---------|---|
| | Maintenance Period you require the firmware version 1.18. Commandline option see here ²⁷² . |

WibuKey

For setting WibuKey options, see the separate "WibuKey Developer Guide".

7.4.2.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network, or both. Moreover, you can define the license allocation (access) mode.

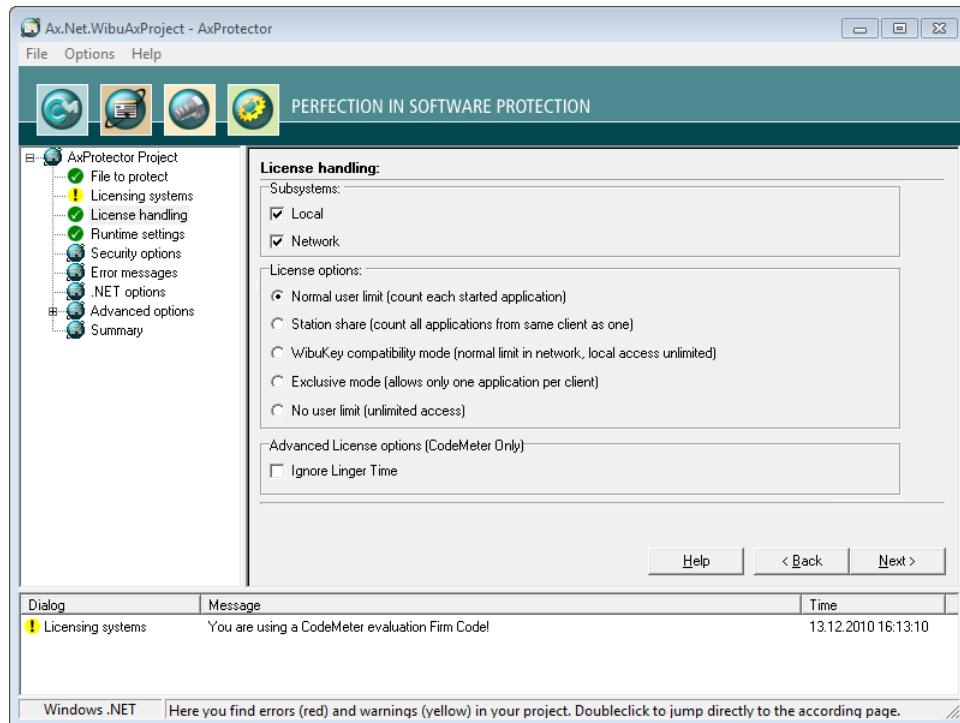


Figure 41: AxProtector .NET - "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 <i>CodeMeter License Server</i> runs and is activated as network server.  On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

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](#)²⁷³).

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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, if 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 <i>WibuKey</i> . <i>WibuKey Systems</i> 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 on a network, and no additional licenses are allocated. Allocated licenses in this mode can be re-used. |

Linger Time

| Element | Description |
|--------------------|--|
| 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 license of protected application has been released or a protected application has been closed. |

7.4.2.4 Runtime Settings

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

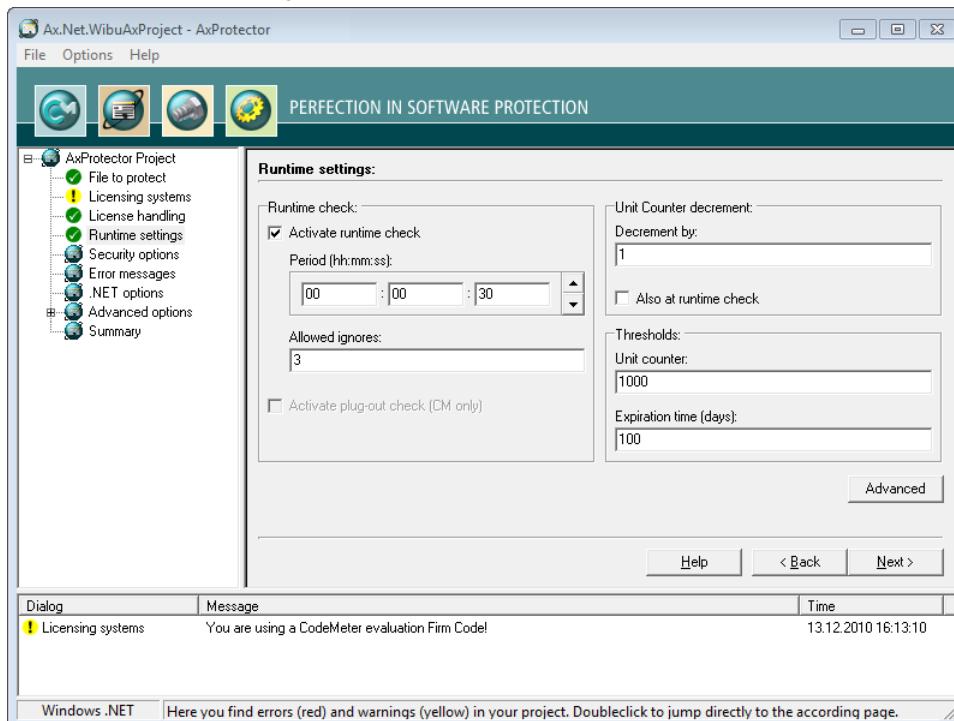


Figure 42: AxProtector .NET - "Runtime Settings"

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 ²⁷⁷ . |
| 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 <i>CmContainer</i> 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 | This option closes the protected application when the <i>CmDongle</i> is removed while the application is running. Immediately, an error message is issued. This option is valid for <i>CmDongle</i> only. |

| Element | Description |
|-----------------|--|
| (only CmDongle) | Commandline option see here ²⁷⁵ . |

Unit Counter Decrement

Decrementing an Unit Counter can serve to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see [here](#)²⁸⁴).

| 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 ¹¹¹ " 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 ¹¹⁹ . |
|--|--|

| Element | Description |
|------------------------|---|
| Unit Counter | If the defined threshold falls short, a warning message is issued. Commandline option see here ²⁸⁶ . |
| 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 ²⁸⁵ . |

7.4.2.4.1 Advanced Runtime Settings

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

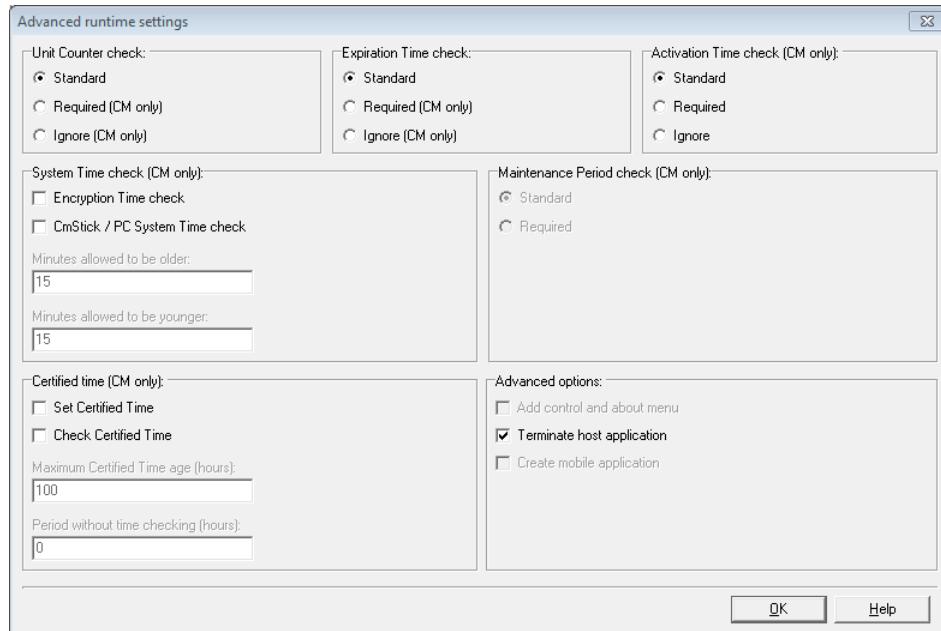


Figure 43: AxProtector .NET - "Advanced Runtime Settings"

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](#)²⁸⁴).

| 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 |

| Element | Description |
|---------|---|
| | 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](#)²⁸⁴).

| 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](#)²⁸³).

| 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 ³⁰⁴ 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](#)²⁸⁴).

| |
|--|
|  The option is available only, if you activated the checkbox Release Date on the page "Licensing systems". |
|--|

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](#)²⁷⁸).

 For information on the fail safe and manipulation safe processes referring to *Activation and Expiration Time* see [here](#)²⁹⁴ ..

| Element | Description |
|--------------------------------------|---|
| Set Certified Time | This option attempts to update the <i>Certified Time</i> in a <i>CmContainer</i> . The <i>certified time</i> is requested from the Time Server.  This option requires a connection to the Internet. |
| Check Certified Time | This option checks to see if the <i>Certified Time</i> 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 <i>Certified Time</i> . |
| Period without time checking (hours) | Specifies the period (in hours) if <u>no</u> check of the <i>Certified Time</i> certificate is taking place. If this period is not reached, a check is not performed. If the <i>Certified Time</i> certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the <i>Certified Time</i> 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 <i>Certified Time</i> 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](#)²⁷⁴).

| 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 <i>CmContainer</i> System Time is newer than the encryption time.  Requires at least <i>CodeMeter® 4.10</i> . |
| CmContainer / PC System Time check | When activated these options define a time corridor in which a difference between <i>CmContainer</i> 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 <i>CmContainer</i> System Time. |
| Minutes to be allowed younger | States in minutes how much PC Time is allowed to be younger than the <i>CmContainer</i> 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 ²⁰⁶). |
| Create mobile application | [not yet implemented] |

7.4.2.5 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, the search intensity for debugger or if a *CmContainer* is locked.

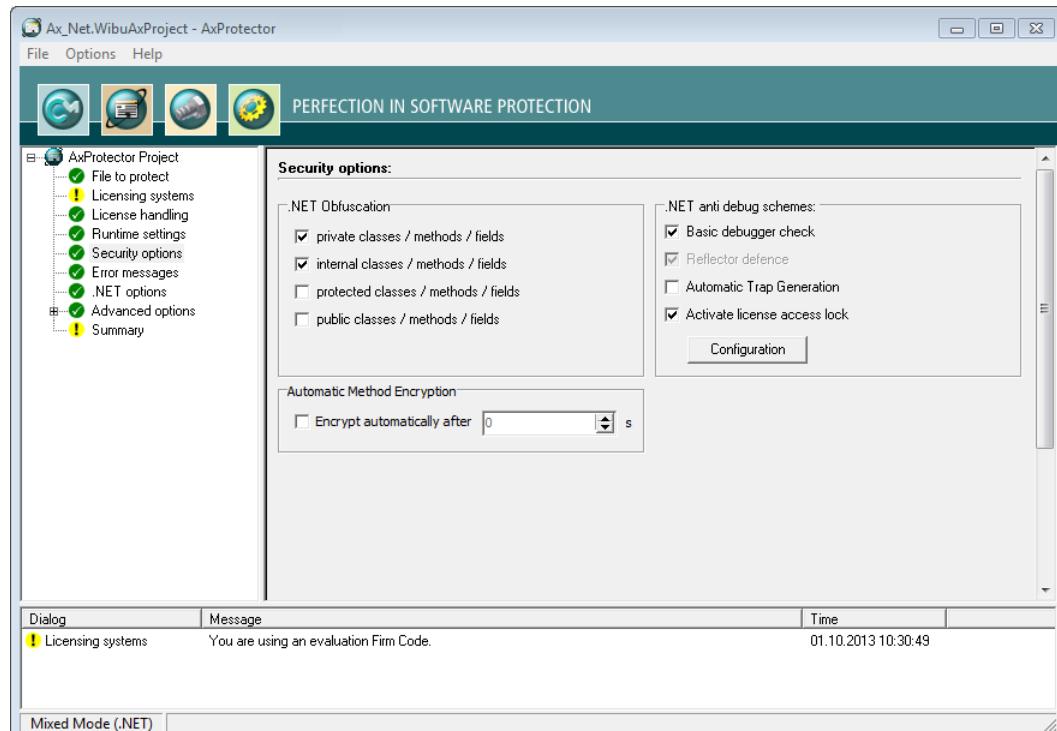


Figure 44: AxProtector .NET - "Security Options"

.NET Obfuscation

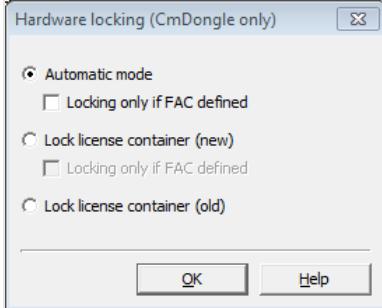
The obfuscation process renames elements to render them meaningless and replaces human-readable information with machine generated information (commandline option see [here](#)²⁰²). Elements comprise classes, methods, and fields.

| Element | Description |
|--------------------------------------|-------------------------------|
| private classes / methods / fields | obfuscates private elements |
| internal classes / methods / fields | obfuscates internal elements |
| protected classes / methods / fields | obfuscates protected elements |
| 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](#)²⁷⁵).

| 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 ²⁸⁹). |
| Activate license access lock | This option locks the license access to the used Firm Item in a <i>CmContainer</i> 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 <i>CodeMeter</i> . | |
| 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. |
| 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 <i>CmContainer</i> . 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. 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 | | | | | | | | | | | | |
| |  <p>Figure 45: AxProtector .NET "Security Options - Hardware Locking"</p> <p>The following settings are available:</p> <table border="1"> <thead> <tr> <th data-bbox="290 667 579 699">Option</th><th data-bbox="579 667 1132 699">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="290 699 579 862">"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)</td><td data-bbox="579 699 1132 862"> If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked. Due to compatibility reasons this corresponds to the default setting. </td></tr> <tr> <td data-bbox="290 862 579 992">"Automatic Mode" activated and "Locking only if FAC defined" activated</td><td data-bbox="579 862 1132 992"> If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. </td></tr> <tr> <td data-bbox="290 992 579 1122">"Lock License Container (new)" activated and "Locking only if FAC defined" not activated</td><td data-bbox="579 992 1132 1122"> If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. </td></tr> <tr> <td data-bbox="290 1122 579 1252">"Lock License Container (new)" and "Locking only if FAC defined" activated</td><td data-bbox="579 1122 1132 1252"> If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. </td></tr> <tr> <td data-bbox="290 1252 579 1317">"Lock License Container (old)" activated</td><td data-bbox="579 1252 1132 1317"> For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. </td></tr> </tbody> </table> | Option | Description | "Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard) | If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked. Due to compatibility reasons this corresponds to the default setting. | "Automatic Mode" activated and "Locking only if FAC defined" activated | If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | "Lock License Container (new)" activated and "Locking only if FAC defined" not activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. | "Lock License Container (new)" and "Locking only if FAC defined" activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | "Lock License Container (old)" activated | For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. |
| Option | Description | | | | | | | | | | | | |
| "Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard) | If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked. Due to compatibility reasons this corresponds to the default setting. | | | | | | | | | | | | |
| "Automatic Mode" activated and "Locking only if FAC defined" activated | If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | | | | | | | | | | | | |
| "Lock License Container (new)" activated and "Locking only if FAC defined" not activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. | | | | | | | | | | | | |
| "Lock License Container (new)" and "Locking only if FAC defined" activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | | | | | | | | | | | | |
| "Lock License Container (old)" activated | For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. | | | | | | | | | | | | |

Automatic Methode Encryption

This group allows re-encrypting methods after n seconds of discardment (commandline option see [here](#) ▶ ²⁸³).

| Element | Description |
|-----------------------------|---|
| Encrypt automatically after | re-encrypts methods after discardment. |
| s | allows to specify the number of seconds after which a discarded method is re-encrypted. |

7.4.2.6 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.

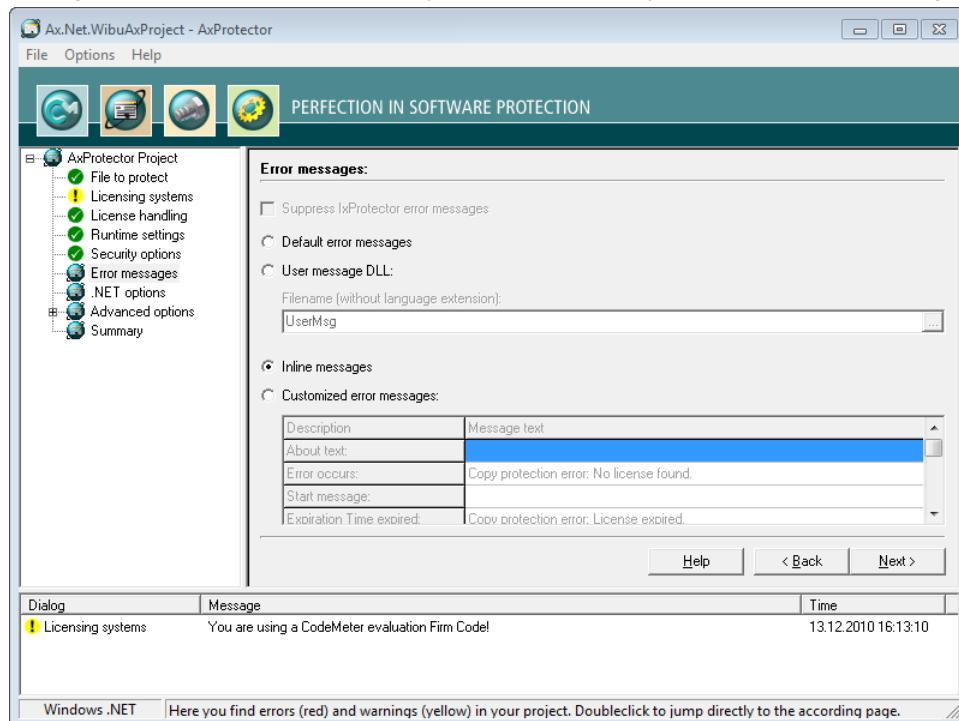
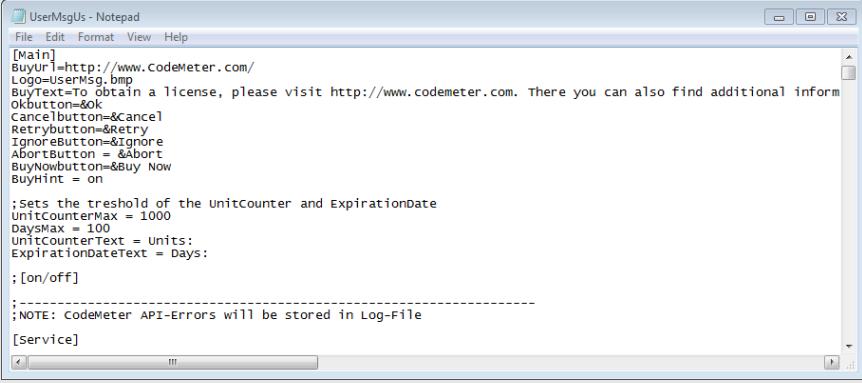


Figure 46: AxProtector .NET "Error Messages"

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 ²⁸⁷). |
| 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 |

| Element | Description |
|----------------------------------|--|
| | <p>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.</p>  <pre data-bbox="280 402 1116 753"> UserMsgUs - Notepad File Edit Format View Help [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 information. Okbutton=ok CancelButton=cancel Retrybutton=&Retry Ignorebutton=&Ignore Abortbutton = &Abort BuyNowbutton=&Buy Now BuyHint = on :sets the threshold 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] </pre> |
| | <p>Figure 47: AxProtector UserMsgUs.ini</p> <p>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.</p> <p>Inline Messages Links for .NET projects, with an inline assembly, can also be configured by *.ini files (commandline option see here).</p> <p> 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.</p> |
| Customized Error Messages | Activate this option to enter customized error messages displayed in the message boxes below. |

7.4.2.7 .NET Options

This page allows you to specify further .NET settings.

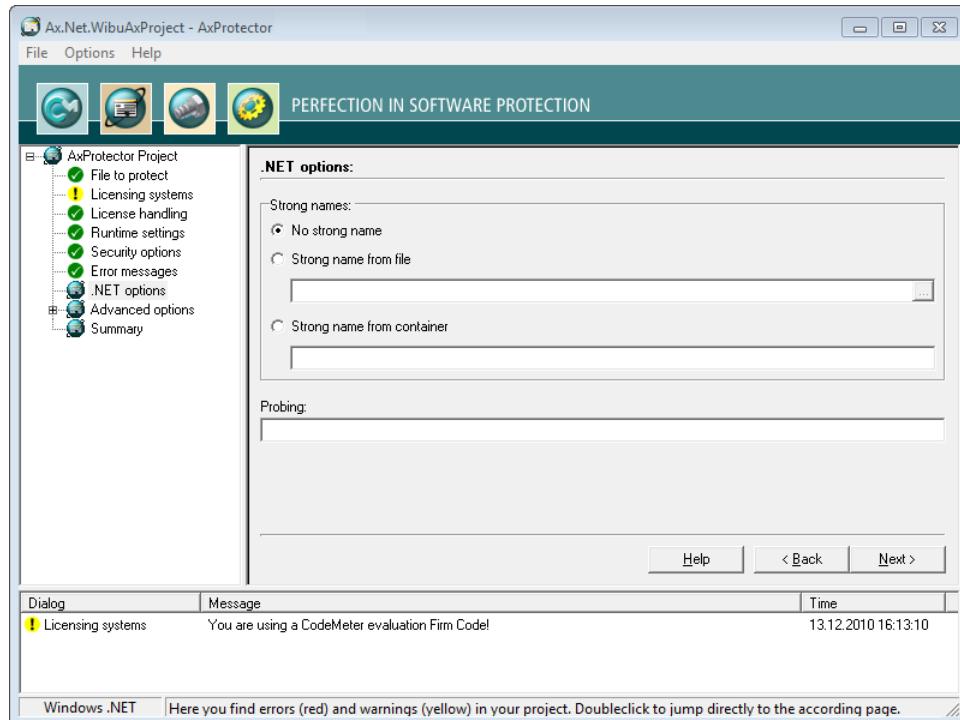


Figure 48: AxProtector .NET - ".NET Options"

.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 ²⁸⁹). |
| Strong Name from Container | Activate this checkbox to use a container file to sign the program class (commandline options see here ²⁸⁹). |
| Probing | <p>This group allows you to specify the location of signed program classes in an <code>app.config</code> file.</p> <p> Specify the path to which the access to the program class refers separated by ";" . Alternatively, specify the respective <code>app.config</code> file.</p> <p>Commandline option see here²⁸⁹.</p> |

7.4.2.8 Advanced Options

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

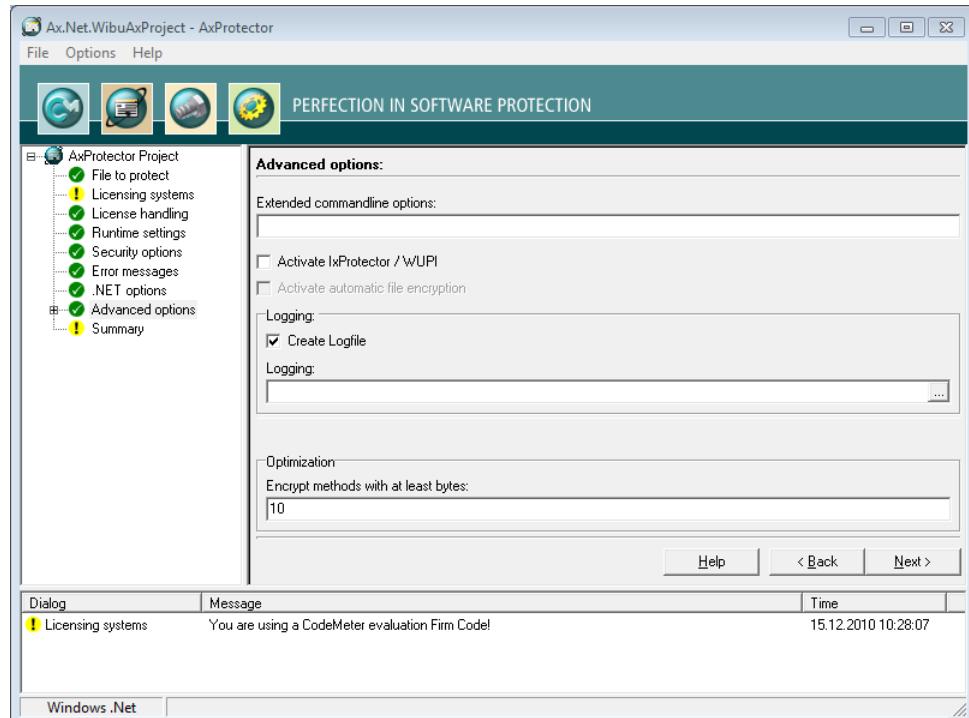


Figure 49: AxProtector .NET - "Advanced Options"

| Element | Description |
|------------------------------|--|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the <i>AxProtector</i> 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 <i>IxProtector</i> via the Software Protection-API ²⁰⁶ . (commandline option see here ²⁰¹). |
| Create Logfile | Activate this checkbox to create file logging for the activities of <i>AxProtector</i> . |
| 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 |

| Element | Description |
|---------|---|
| | 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 ²⁸³ . |

7.4.2.8.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the [Software Protection-API \(WUPI\)](#)²⁹⁶.

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

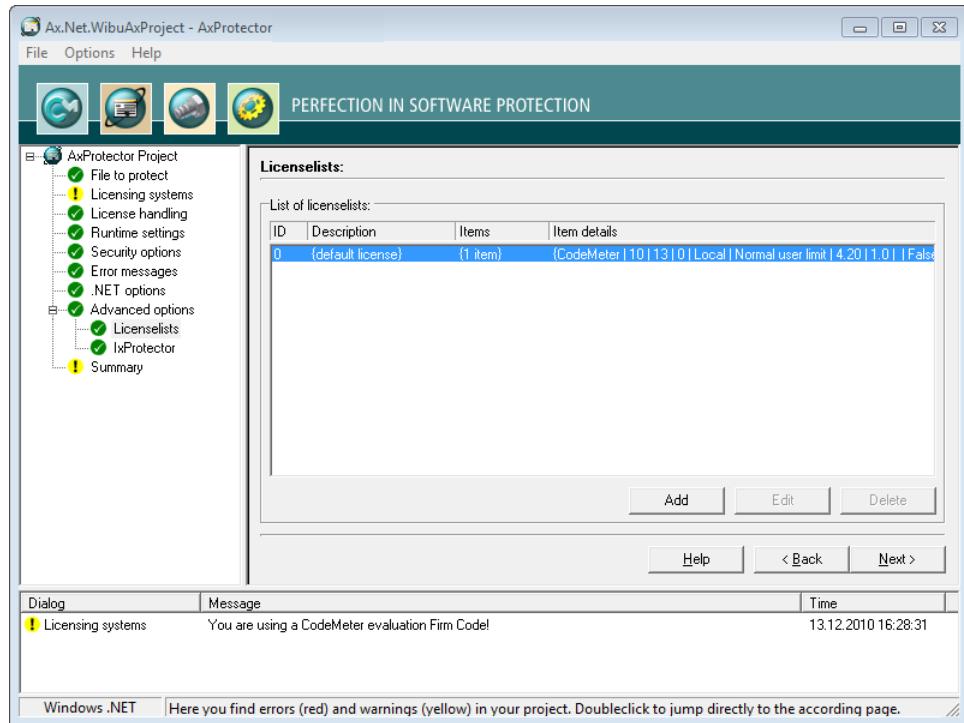
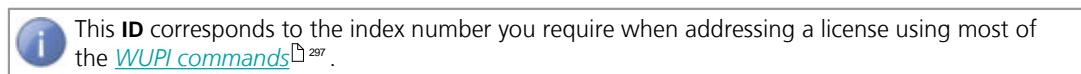
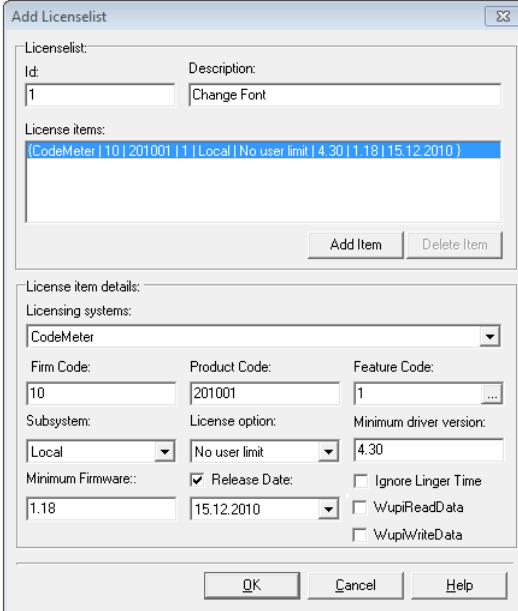


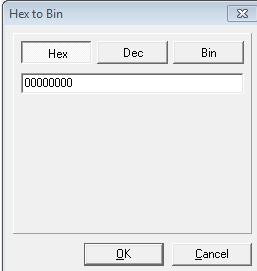
Figure 50: AxProtector .NET - "License Lists"

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 | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p> 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.</p> |
| Description | <p>Here you will describe a license list with text..</p> <p>3. Define the license by completing the fields in the License item details group.</p>  |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> or <i>WibuKey</i>). |
| 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. Using the "... " button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. |

| Element | Description |
|------------------------|---|
| |  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 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). |
| WupiReadData | Activate this option to read data ³⁰⁰ from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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 dis-

played in the license item list.

5. Click the "OK" button. The new license data is added to the license list.

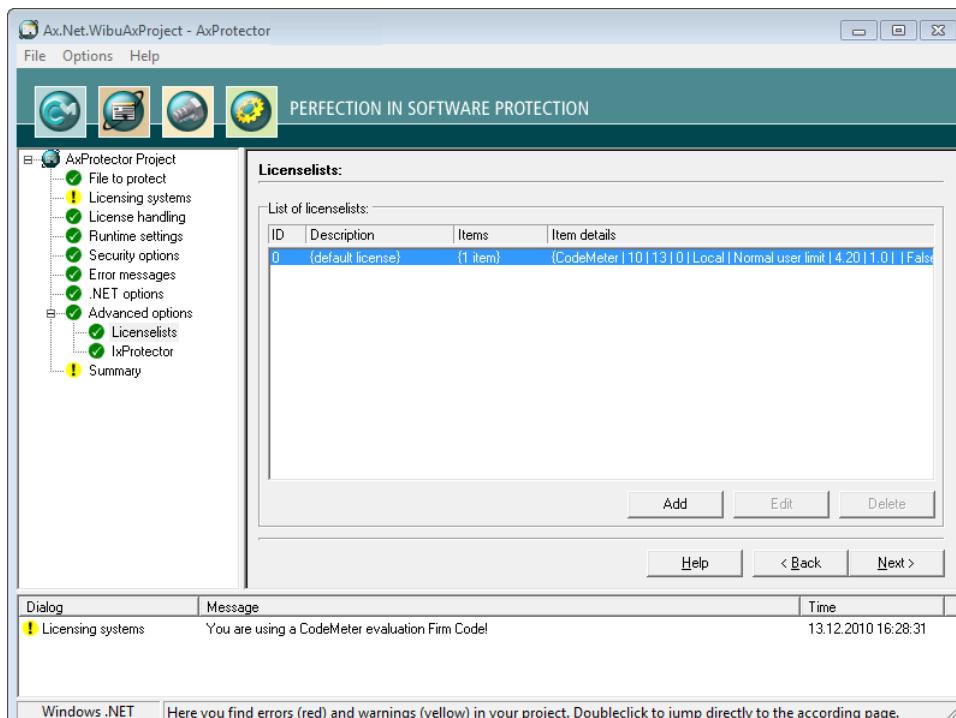


Figure 52: AxProtector .NET - "Completed License Lists"

7.4.2.8.2 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.

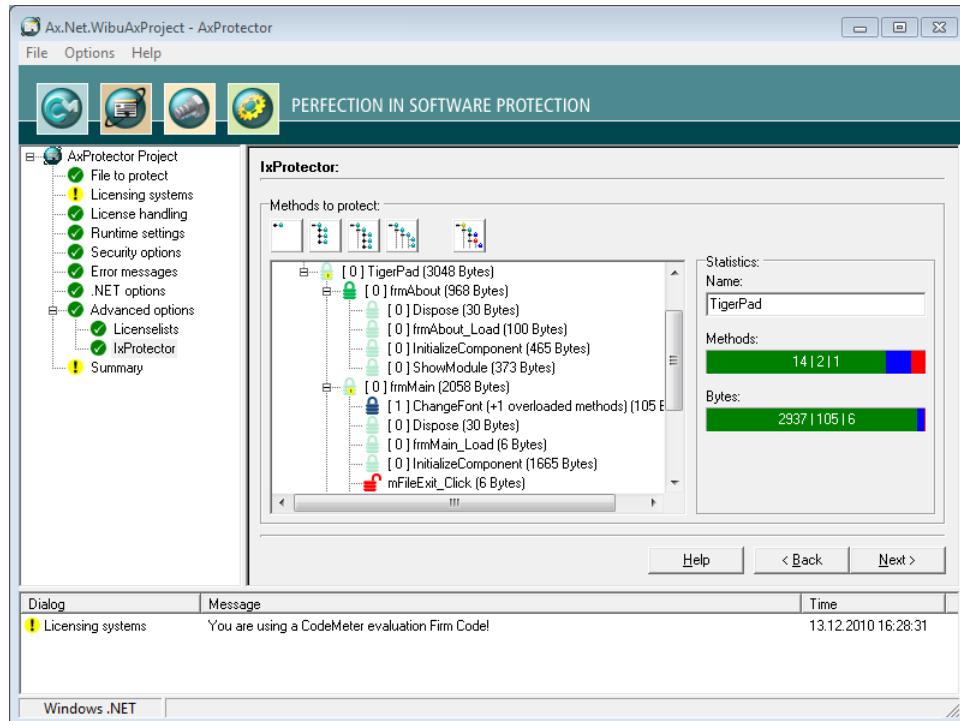


Figure 53: AxProtector .NET - "IxProtector"

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. |

| Element | Description | |
|---------|---|--|
| | Color | Description |
| 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. | |
| | Green | Shows that the method will be encrypted using <i>AxProtector</i> and that the License List ID has a value of 0 (default license) |
| | Blue | Shows that the method will be encrypted using <i>IxProtector</i> and that the License List ID has a value unequal 0. |
| | Red | Shows that the method is 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 <i>AxProtector</i> and that the License List ID has a value of 0 (default license) |
| | Blue | Shows that the method will be encrypted using <i>IxProtector</i> and that the License List ID has a value unequal 0. |
| | Red | Shows that the method is 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 <i>AxProtector</i> (License List ID with a value of 0, i.e. default license). |
| | Encrypts the selected element using <i>IxProtector</i> (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.

7.4.2.9 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](#)²⁶ type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

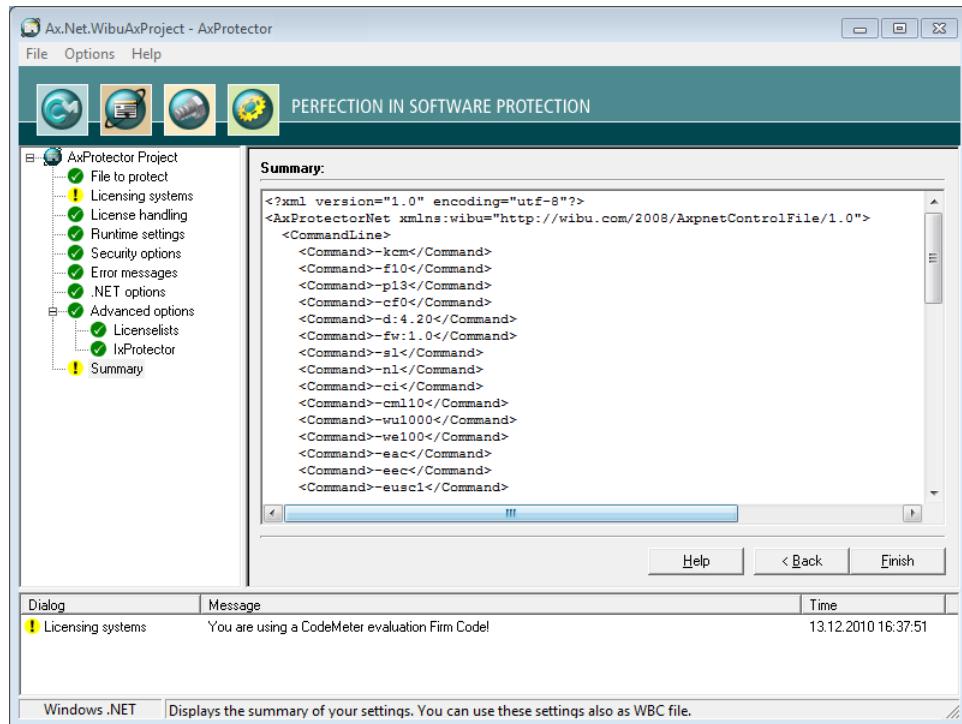


Figure 54: AxProtector .NET "Summary"

| 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.

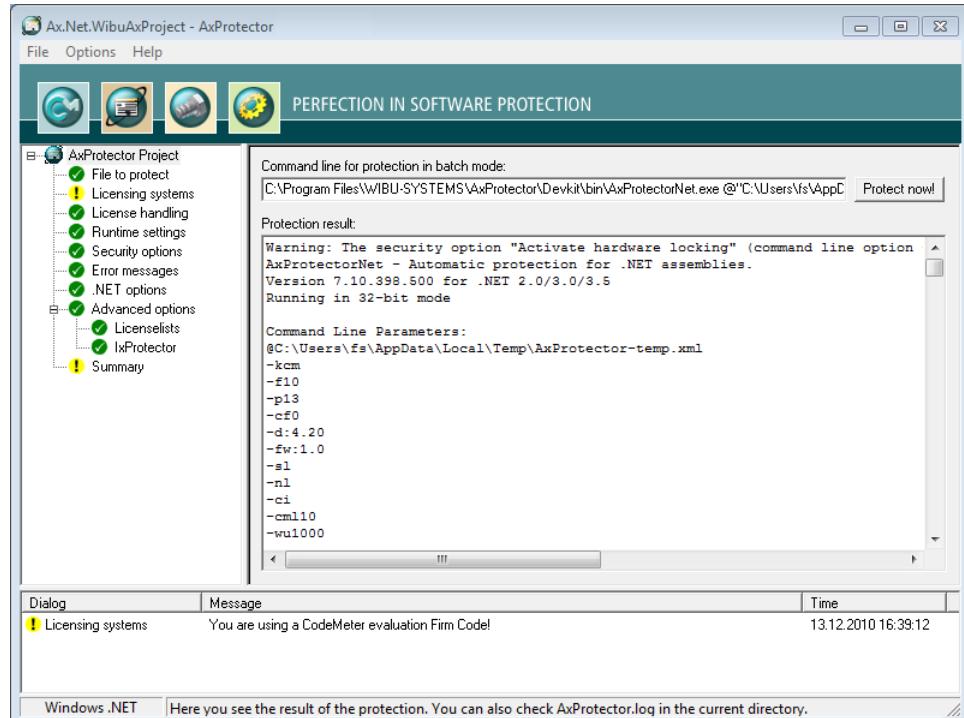


Figure 55: AxProtector .NET - "Encryption Result"

| Element | Description |
|-------------|--|
| Protect now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the AxProtector commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

7.4.3 Mac OS X Application or Dylib

For this project type applications to be encrypted comprise Mac OS X applications starting with Version 10.4. Application created for Mac OS X 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 |
|-------------------------------|--|---|------------------------------------|
| Mac OS X Application or Dylib |  AxProtector Mac ¹³¹ |  | Windows commandline ¹²⁰ |

| Application to be protected | Project type | GUI Windows | Commandline |
|-----------------------------|--------------|-------------|---|
| | | | In a separate commandline for Mac, running on Mac OS X operating systems, you are also able to insert encryption parameter ¹⁵⁴ . |

7.4.3.1 File to protect

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

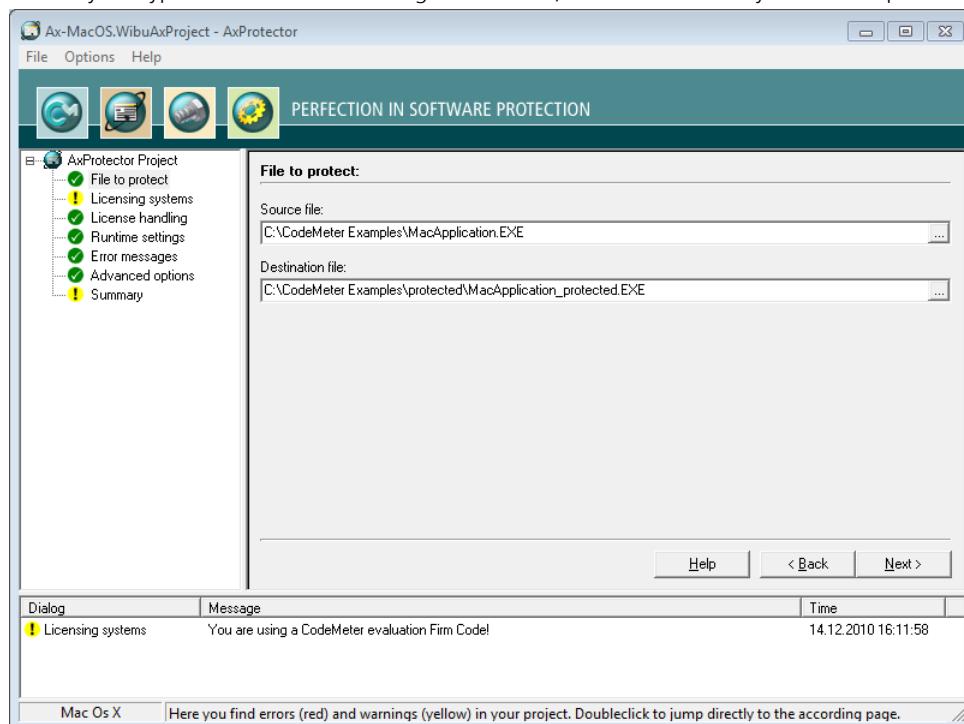


Figure 56: AxProtector - Mac OS X "File to Protect"

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. |

| Element | Description |
|------------------|--|
| | 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 . |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

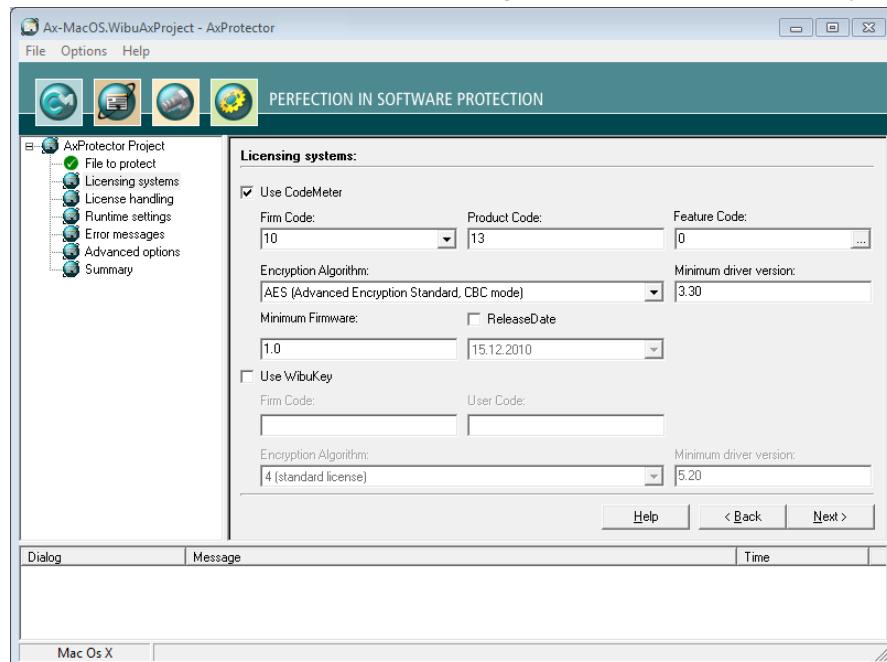
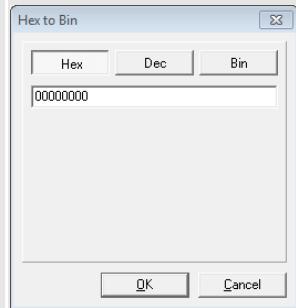


Figure 57: AxProtector - Mac OS X "Licensing Systems"

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.

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|------------------------|--|
| Firm Code | <p>Specify the Firm Code to be used for encrypting the software.</p> <p> The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation Firm Code found in the <i>CodeMeter® Software Development Kit (SDK)</i>. In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code..The test Firm Code for <i>CmActLicense</i> is 5010. As a registered licensor, you will be issued your own unique Firm Code(s).</p> <p>Commandline option see here²⁷¹.</p> |
| Product Code | <p>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.</p> <p>Commandline option see here²⁷¹.</p> |
| Feature Code | <p>Enter the Feature Code which defines, for example, the encryption of different software versions.</p> <p> 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.</p> <p>Using the "..." button you may enter the feature map value in hexadecimal, decimal or binary format.</p>  |
| Encryption Algorithm | <p>Select the algorithm to encrypt your software. Currently, <i>CodeMeter®</i> solely supports AES (Advanced Encryption Standard).</p> <p>Commandline option see here²⁷².</p> |
| Minimum Driver Version | <p>Enter the minimum driver version required for the installed <i>CodeMeter License Servers</i>. When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that <i>AxProtector</i> automatically handles sessions of the protected software, and each session is allocated one of the available licenses.</p> <p> 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.</p> |

| Element | Description |
|------------------|--|
| Release Date | Starting with Firmware version 1.18 CodeMeter® supports the Product Item Option Maintenance Period [45] |
| 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. Commandline option see here [272]. |

WibuKey

For setting WibuKey options, see the separate "WibuKey Developer Guide".

7.4.3.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network, or both. Moreover, you can define the license allocation (access) mode.

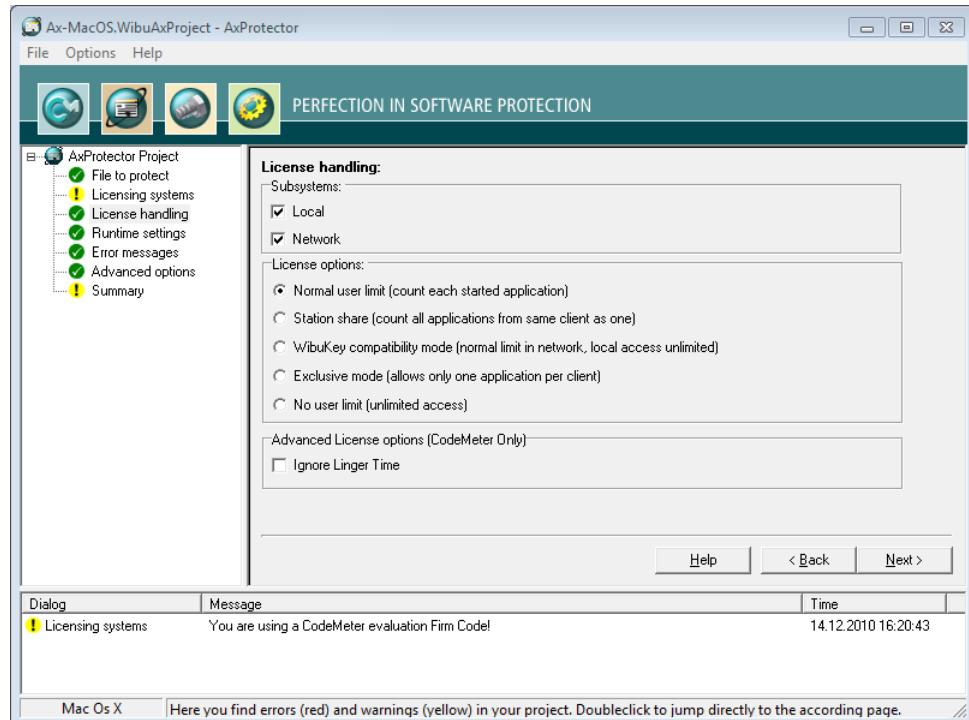


Figure 59: AxProtector - Mac OS X "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 on the network, i.e. only PCs are accessed where <i>CodeMeter License Server</i> runs and is activated as network server.  On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

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](#)²⁷³).

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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 <u>recommends</u> 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. |

Linger Time

| Element | Description |
|--------------------|--|
| 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 license of a protected application has been released or the protected application has been closed (more information in the <i>CodeMeter Developer Guide</i>). |

7.4.3.4 Runtime Settings

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

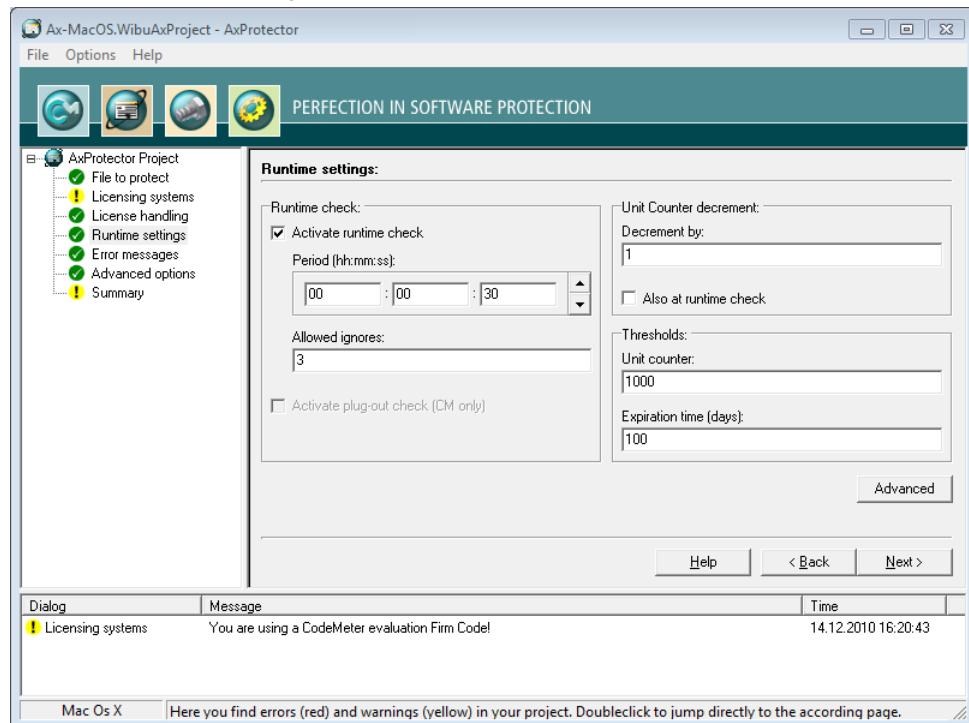


Figure 60: AxProtector - Mac OS X "Runtime Settings"

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 ²⁷⁷ . |
| 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 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  If the connection to a <i>CmContainer</i> 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. </div> |
| Activate Plug-out Check | This option closes the protected application if the <i>CmDongle</i> is removed while the application is running. Immediately, an error message is issued. This option is valid for <i>CmDongle</i> only. |

| Element | Description |
|-----------------|--|
| (only CmDongle) | Commandline option see here ²⁷⁵ . |

Unit Counter Decrement

Decrementing an Unit Counter can serve to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see [here](#)²⁸⁴).

| 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 ¹³⁶ " 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 ¹⁴¹ . |
|---|--|

| Element | Description |
|------------------------|---|
| Unit Counter | If the defined threshold falls short, a warning message is issued. Commandline option see here ²⁸⁶ . |
| 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 ²⁸⁵ . |

7.4.3.4.1 Advanced Runtime Settings

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

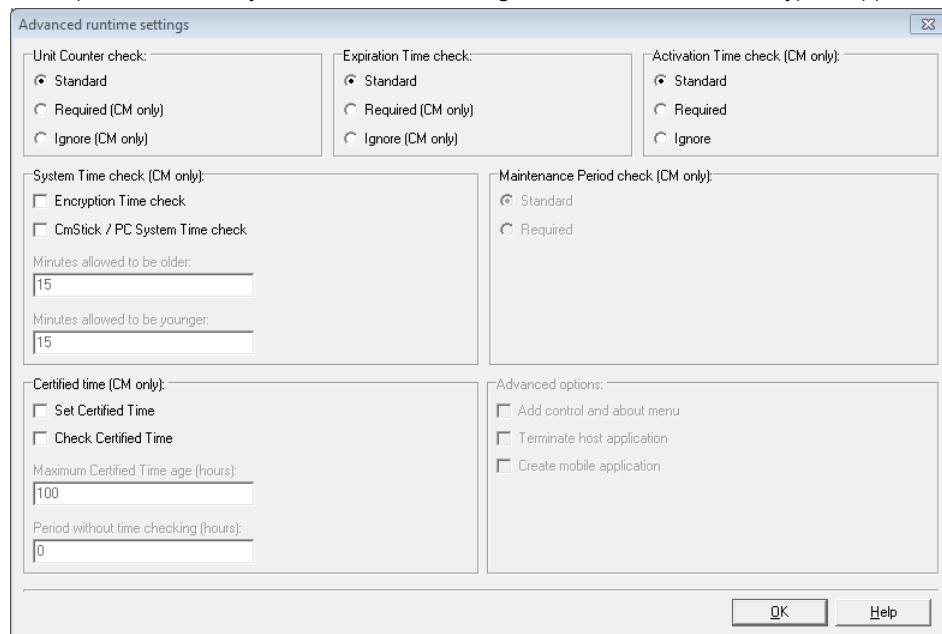


Figure 61: AxProtector - Mac OS X "Advanced Runtime Settings"

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](#)²⁸⁴).

| 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 |

| Element | Description |
|---------|---|
| | 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](#)²²⁴).

| 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](#)²²³).

| 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 ²³⁴ 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](#)²²⁴).

| |
|---|
|  The option is available only, if you activated the checkbox Release Date on the page "Licensing systems" ¹³² . |
|---|

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](#)  ²⁷⁸).

 For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see [here](#)  ³⁹⁴ ..

| Element | Description |
|--------------------------------------|--|
| Set Certified Time | This option attempts to update the <i>Certified Time</i> in a <i>CmDongle</i> . The <i>Certified Time</i> is requested from the Time Server.  This option requires a connection to the Internet. |
| Check Certified Time | This option checks to see if the <i>Certified Time</i> 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 <i>Certified Time</i> . |
| Period without time checking (hours) | Specifies the period (in hours) when <u>no</u> check of the <i>Certified Time</i> certificate is performed. If this period is not reached, a check is not performed. If the <i>Certified Time</i> certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the <i>Certified Time</i> 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 <i>Certified Time</i> 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](#)  ²⁷⁴).

| 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 <i>CmContainer</i> System Time is newer than the encryption time.  Requires at least <i>CodeMeter® 4.10</i> . |
| CmContainer / PC System Time check | When activated these options define a time corridor in which a difference between <i>CmContainer</i> 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 <i>CmContainer</i> System Time. |
| Minutes to be allowed younger | States in minutes how much PC Time is allowed to be younger than the <i>CmContainer</i> System Time. |

7.4.3.5 Error Messages

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

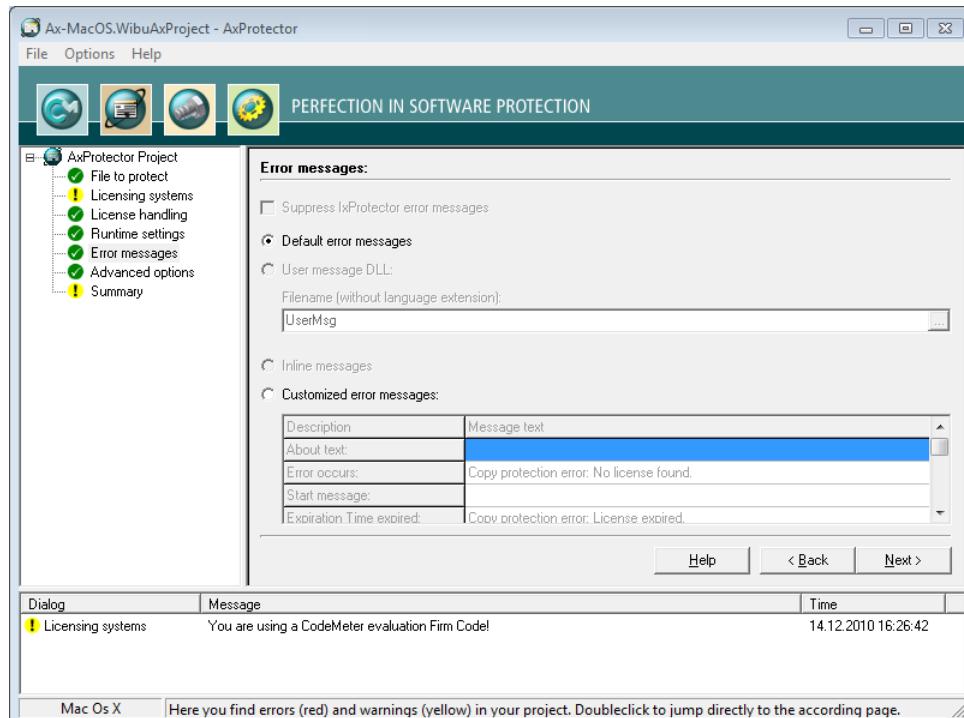


Figure 62: AxProtector - Mac OS X "Error Messages"

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 ²⁸⁷). |
| Customized Error Messages | Activate this option to enter customized error messages displayed in the message boxes below. |

7.4.3.6 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.

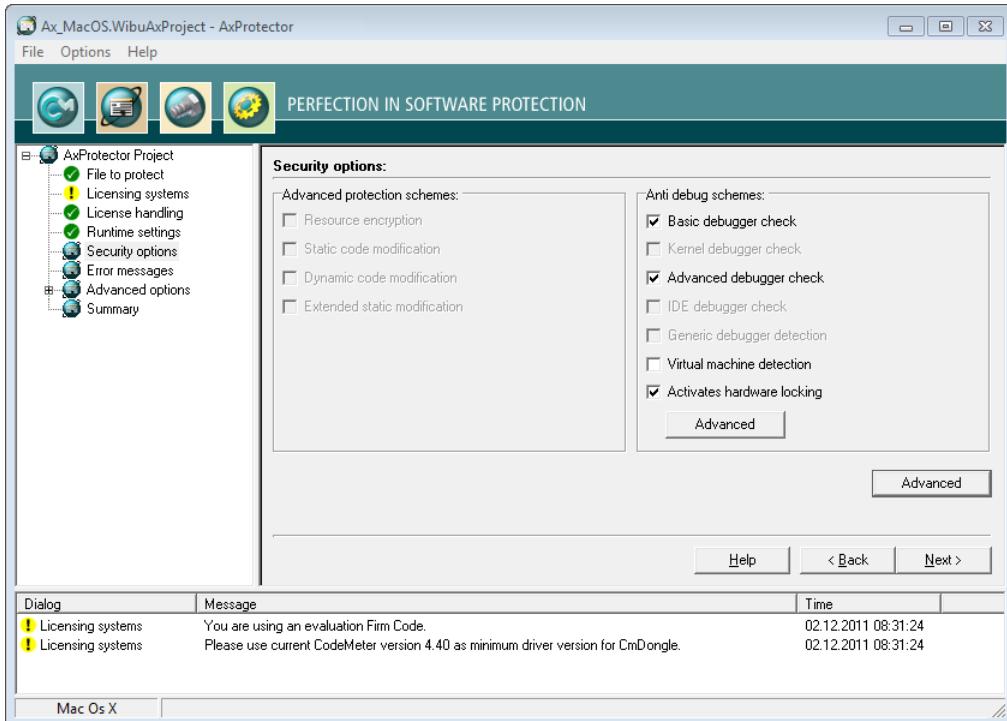
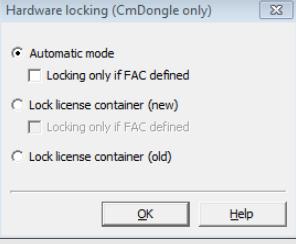
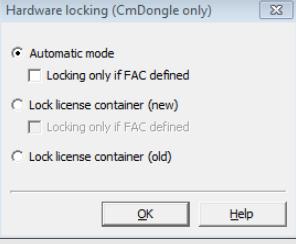
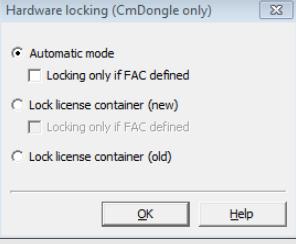


Figure 63: AxProtector - Mac OS "Security Options"

Anti-Debug Schemes

Debugger programs serve an honest role in searching for errors 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](#)²⁷⁵).

| 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 | This option locks the license access to the used Firm Item in a <i>CmContainer</i> as soon as a de- |

| Element | Description | | | | | | |
|---|--|------------------|-------------|---|---|---|---|
| lock | <p>bugger program is detected. If this option is activated, the settings are applied you defined in the dialog to be opened by the "Configuration" button.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  This button is activated only for <i>CodeMeter</i>. </div> | | | | | | |
| Configuration | <p>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.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Locking Scenario</th><th>Description</th></tr> </thead> <tbody> <tr> <td>immediate locking</td><td>is performed starting with Firmware Version 1.14 as soon as a debugger is detected.</td></tr> <tr> <td>prepared locking</td><td> <p>is performed by checking the Firm Access Counter (FAC). The Firm Access Counter locates at the Firm Item level of a <i>CmContainer</i>. This counter allows you to control whether a Firm Item can be used for encryption and decryption operations.</p> <p>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. 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.</p> <p>The owner / end-user of the locked Firm Items must contact the software vendor for unlocking codes. This can be done by remote programming.</p> <div style="text-align: center; margin-top: 10px;">  </div> </td></tr> </tbody> </table> | Locking Scenario | Description | immediate locking | is performed starting with Firmware Version 1.14 as soon as a debugger is detected. | prepared locking | <p>is performed by checking the Firm Access Counter (FAC). The Firm Access Counter locates at the Firm Item level of a <i>CmContainer</i>. This counter allows you to control whether a Firm Item can be used for encryption and decryption operations.</p> <p>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. 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.</p> <p>The owner / end-user of the locked Firm Items must contact the software vendor for unlocking codes. This can be done by remote programming.</p> <div style="text-align: center; margin-top: 10px;">  </div> |
| Locking Scenario | Description | | | | | | |
| immediate locking | is performed starting with Firmware Version 1.14 as soon as a debugger is detected. | | | | | | |
| prepared locking | <p>is performed by checking the Firm Access Counter (FAC). The Firm Access Counter locates at the Firm Item level of a <i>CmContainer</i>. This counter allows you to control whether a Firm Item can be used for encryption and decryption operations.</p> <p>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. 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.</p> <p>The owner / end-user of the locked Firm Items must contact the software vendor for unlocking codes. This can be done by remote programming.</p> <div style="text-align: center; margin-top: 10px;">  </div> | | | | | | |
| | <p>Figure 64: AxProtector - Mac OS "Security Options - Hardware Locking"</p> <p>The following settings are available:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Option</th><th>Description</th></tr> </thead> <tbody> <tr> <td>"Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard)</td><td> <p>If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>Due to compatibility reasons this corresponds to the default setting.</p> </td></tr> <tr> <td>"Automatic Mode" activated and "Locking only if FAC defi-</td><td>If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1.</td></tr> </tbody> </table> | Option | Description | "Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard) | <p>If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>Due to compatibility reasons this corresponds to the default setting.</p> | "Automatic Mode" activated and "Locking only if FAC defi- | If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. |
| Option | Description | | | | | | |
| "Automatic Mode" activated and "Locking only if FAC defined" not activated (Standard) | <p>If the Firmware is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immediately locked.</p> <p>Due to compatibility reasons this corresponds to the default setting.</p> | | | | | | |
| "Automatic Mode" activated and "Locking only if FAC defi- | If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. | | | | | | |

| Element | Description | | | | | | | | | | |
|--|--|--------|-------------|-----------------|--|--|---|--|--|--|---|
| | <table border="1"> <thead> <tr> <th>Option</th><th>Description</th></tr> </thead> <tbody> <tr> <td>"ned" activated</td><td>If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked.</td></tr> <tr> <td>"Lock License Container (new)" activated and "Locking only if FAC defined" not activated</td><td>If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher.</td></tr> <tr> <td>"Lock License Container (new)" and "Locking only if FAC defined" activated</td><td>If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked.</td></tr> <tr> <td>"Lock License Container (old)" activated</td><td>For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1.</td></tr> </tbody> </table> | Option | Description | "ned" activated | If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | "Lock License Container (new)" activated and "Locking only if FAC defined" not activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. | "Lock License Container (new)" and "Locking only if FAC defined" activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | "Lock License Container (old)" activated | For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. |
| Option | Description | | | | | | | | | | |
| "ned" activated | If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | | | | | | | | | | |
| "Lock License Container (new)" activated and "Locking only if FAC defined" not activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. | | | | | | | | | | |
| "Lock License Container (new)" and "Locking only if FAC defined" activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. | | | | | | | | | | |
| "Lock License Container (old)" activated | For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. | | | | | | | | | | |

7.4.3.6.1 Advanced Security Options

This input window lets you define further settings.



Figure 65: AxProtector - Mac OS "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

| Element | Description |
|------------------------------------|--|
| Add virus check | Adds a virus check to the protected application by using a check sum (commandline option see here ²⁷⁸). |
| Link API statically to Application | The <i>CodeMeter Core API</i> is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here ²⁷⁹). |
| Size of encrypted Code (in %) | Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here ²⁷⁸). |

7.4.3.7 Advanced Options

This input window lets you set further encryption options.

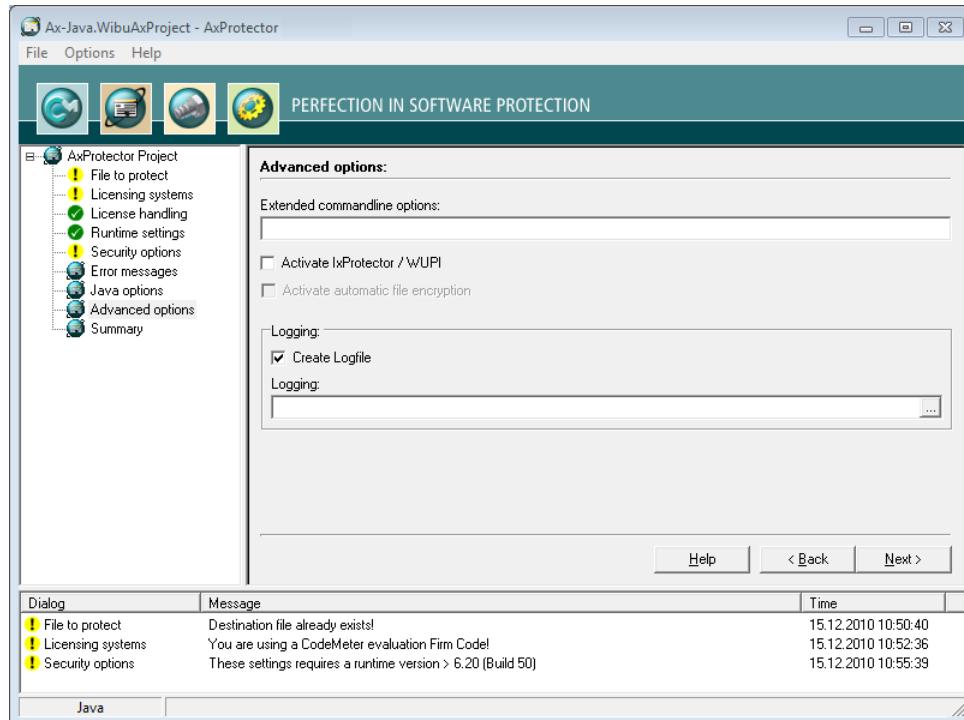


Figure 66: AxProtector - Mac OS X "Advanced Options"

| Element | Description |
|------------------------------|--|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the AxProtector commandline. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i For more information please contact support at Wibu-Systems. </div> |
| 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 ²⁹⁶ . (commandline option see here ²⁹⁷). |
| 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. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

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 \(WUPI\)](#)²⁹⁶.

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

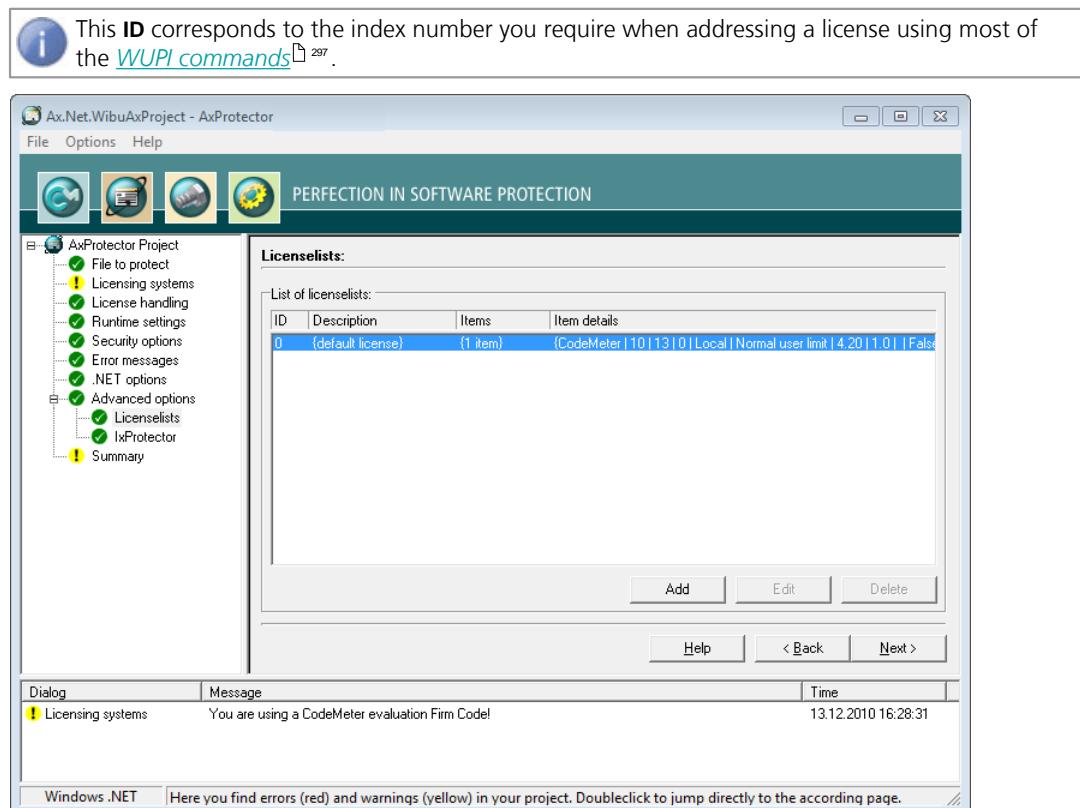
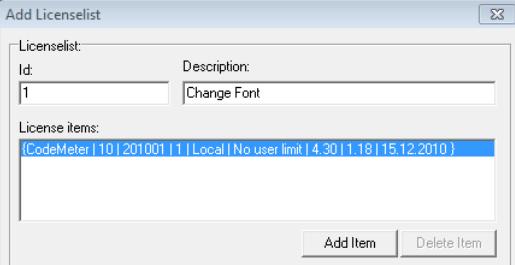


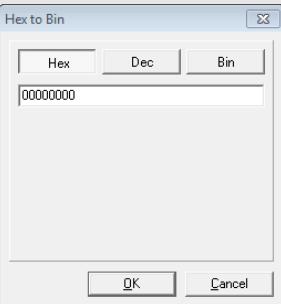
Figure 67: AxProtector Mac OS X - "License Lists"

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 | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p> 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.</p> |

| Element | Description | | | | | | | | | | |
|-------------------|---|----------------------|---------------------|---|-------------|-------------------|----------------------|----------------------|----|--------|---|
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  <p>Licenseslist:</p> <table border="1"> <tr> <td>Id:</td> <td>Description:</td> </tr> <tr> <td>1</td> <td>Change Font</td> </tr> </table> <p>License items:</p> <pre>(CodeMeter 10 201001 1 Local No user limit 4.30 1.18 15.12.2010)</pre> <p>License item details:</p> <p>Licensing systems:</p> <p>CodeMeter</p> <table border="1"> <tr> <td>Firm Code:</td> <td>Product Code:</td> <td>Feature Code:</td> </tr> <tr> <td>10</td> <td>201001</td> <td>1</td> </tr> </table> <p>Subsystem:</p> <p>Local</p> <p>License option:</p> <p>No user limit</p> <p>Minimum driver version:</p> <p>4.30</p> <p>Minimum Firmware:</p> <p>1.18</p> <p>Release Date:</p> <p>15.12.2010</p> <p>Ignore Linger Time: <input checked="" type="checkbox"/></p> <p>WapiReadData: <input type="checkbox"/></p> <p>WapiWriteData: <input type="checkbox"/></p> <p align="center">OK Cancel Help</p> | Id: | Description: | 1 | Change Font | Firm Code: | Product Code: | Feature Code: | 10 | 201001 | 1 |
| Id: | Description: | | | | | | | | | | |
| 1 | Change Font | | | | | | | | | | |
| Firm Code: | Product Code: | Feature Code: | | | | | | | | | |
| 10 | 201001 | 1 | | | | | | | | | |
| | Figure 68: AxProtector Mac OS - "Add License Lists" | | | | | | | | | | |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). | | | | | | | | | | |
| 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. Using the "... " button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. | | | | | | | | | | |

| Element | Description |
|------------------------|--|
| |  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>This license option allows to define an allocation time of the license after the license of a protected application has been released or the protected application has been closed (more Information in the <i>CodeMeter Developer Guide</i>).</p> |
| WupiReadData | Activate this option to read data ³⁰⁰ from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

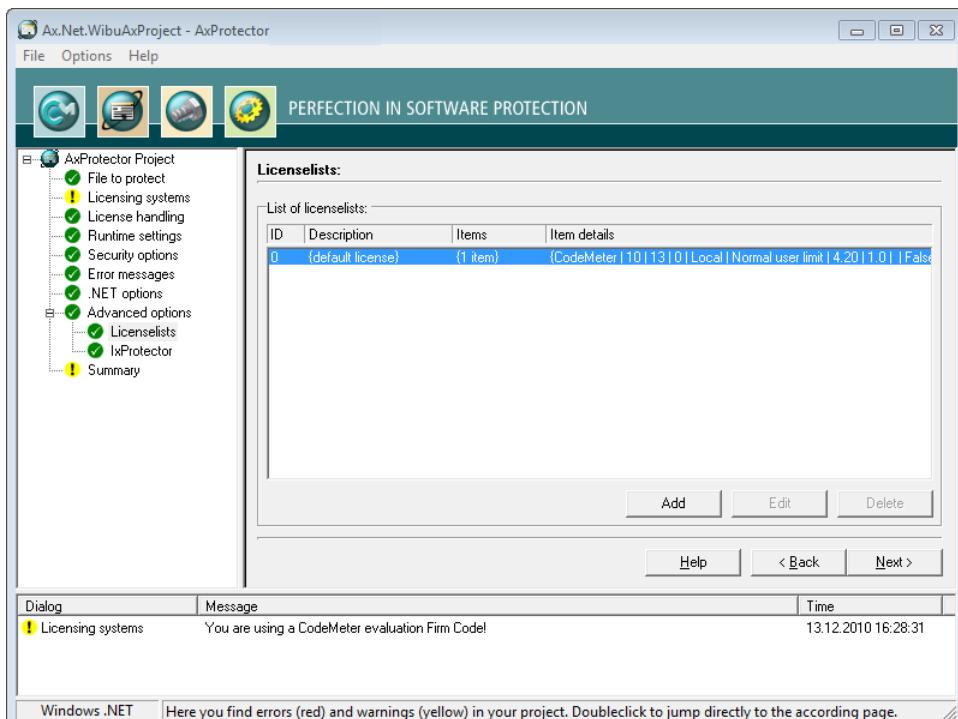


Figure 69: AxProtector .NET - "Completed License Lists"

7.4.3.7.2 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.

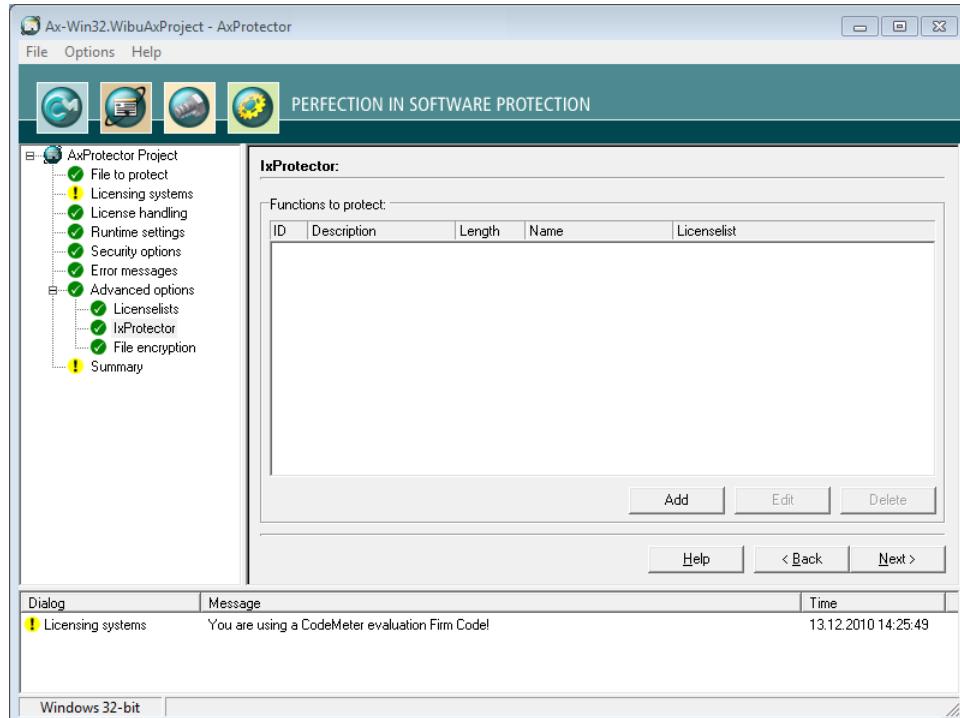
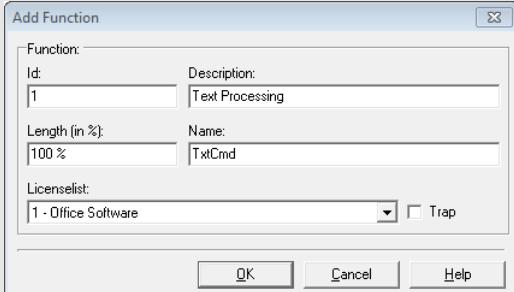


Figure 70: AxProtector - Mac OS X - "Function List"

| Element | Description |
|----------------------|---|
| Functions to protect | <p>Lists all specified function lists, including all properties.</p> <p>This menu item lets you also create function lists. Please proceed as follows:</p> <ol style="list-style-type: none"> 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 71: AxProtector - Mac OS X - "Add Function" | |
| Element | Description |
| Id | <p>Uniquely identifies the function.</p> <p>ⓘ This Id corresponds to the identification you use when calling the WUPI commands WupiDecryptCode²⁹⁸ and WupiEncryptCode²⁹⁹.</p> |
| Description | Enter a description of the function with text. |
| Length | <p>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.</p> <p>ⓘ If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.</p> |
| Name | <p>Specify the name of the function to be encrypted.</p> <p>The function name must exactly match the name used in the export list of the linked map file.</p> <p>Please note the correct spelling (case sensitive, underline, etc.). Microsoft Dependency Walker shows dependencies between 32-or 64-bit Windows PE files. A tree view shows all linked modules and imported and exported functions are displayed in tables. Dependency Walker is part of Windows XP SP2 Support Tools and also part of Microsoft Visual Studio including version 8.0 (Visual Studio 2008, i.e. version 9.0 no longer provides Dependency Walker).</p> |
| 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. Command line option see here ²⁸⁵ . |
| 3. Click the "OK" button. The new functions are added to the function list. | |

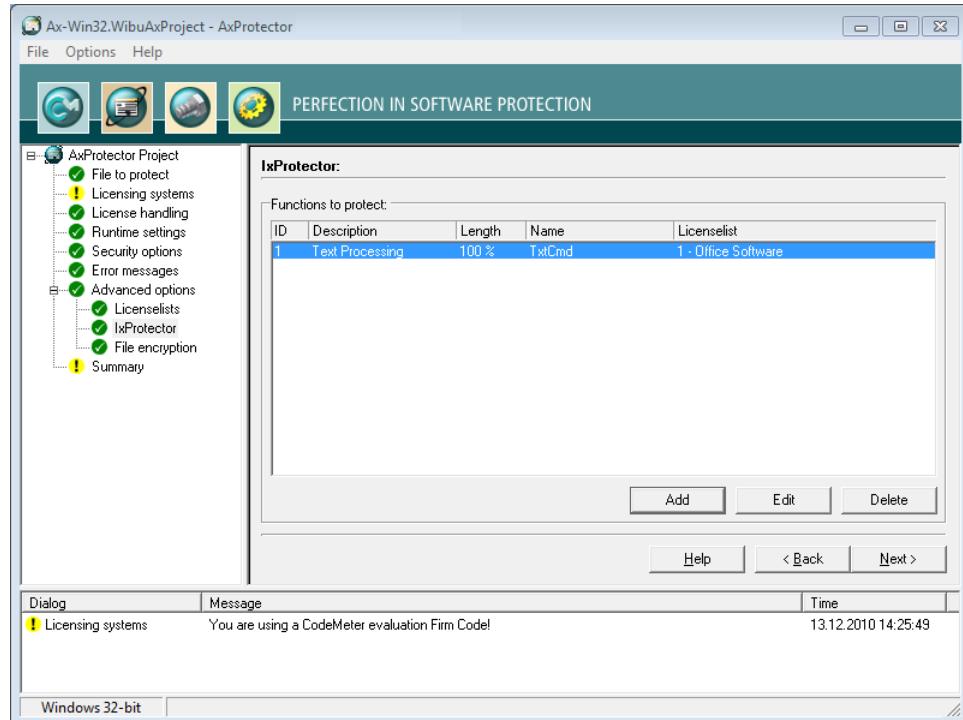


Figure 72: AxProtector - Mac OS X - "Completed Function List"

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](#)^{D₂₉₅} type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

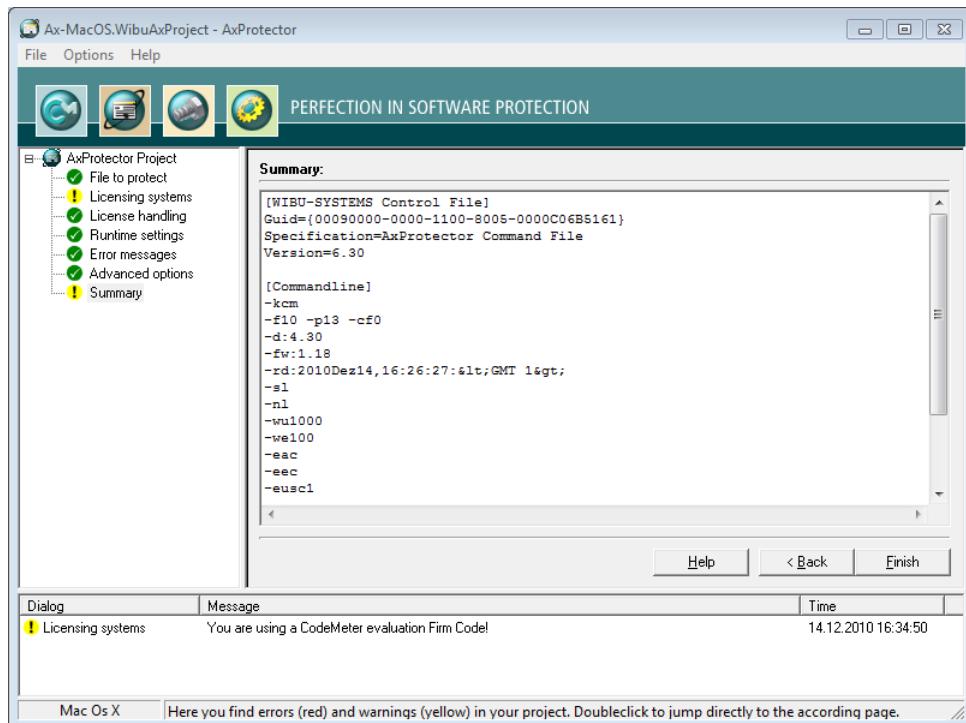


Figure 73: AxProtector - Mac OS X "Summary"

| 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.

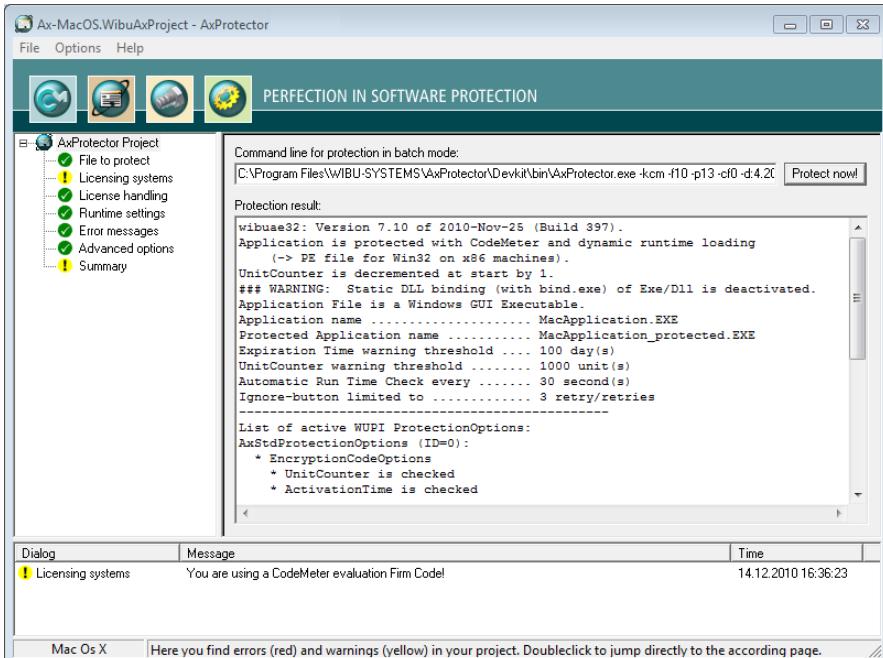


Figure 74: AxProtector - Mac OS X "Encryption Result"

| Element | Description |
|-------------|---|
| Protect now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the <i>AxProtector</i> commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i You are also able to copy the <i>AxProtector</i> 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. </div> |

7.4.4 Java Application (jar file)

Compiled Java code, unlike .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 the *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 after-

ward re-packed in the archive together with some necessary class files of Wibu-Systems.

ClassLoader Modification

On start of the application, at first, the `SystemClassLoader` is loaded: by and by, on demand it re-loads the classes required in the application. It is exactly here where *AxProtector* comes in. On encrypting *AxProtector* also modifies the manifest file resulting in a modified application start. Instead loading the `System ClassLoader` *AxProtector* loads a `ClassLoader` provided by Wibu-Systems using two helper classes (wrapper / starter). This `wibuClassLoader` manages the loading of the encrypted classes, and not encrypted classes are still loaded by the originally Java-included `ClassLoader`.

Decryption in native code

Java itself provides a number of different options to interact into loading processes. Thus, decrypting within Java code is not reasonable, and easily nullified. In *AxProtector Java* the `WibuClassLoader` passes on the encrypted classes to a native library. In this `wibuXPM4J` library the class is now decrypted according to the selected licensing system (`CmDongle` / `CmActLicense` / `WibuKey`), and passed on to the native Java library. The decrypted class then is available in Java without any restrictions.

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 6 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 exclusively works with the original Sun Java, i.e. the usually used variant of Java. Along with the files located in the jar archive, the user requires the native `wibuXPM4J` library mentioned above. It is included for Windows and Mac 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 ¹⁵⁷ | ✓ | Windows commandline ²⁷⁰ In a separate commandline for Java, running on Windows, Mac OS X-, Linux- and Solaris -Betriebssystemen operating systems, you are also able to insert encryption parameter ¹⁷³ . |

7.4.4.1 File to protect

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

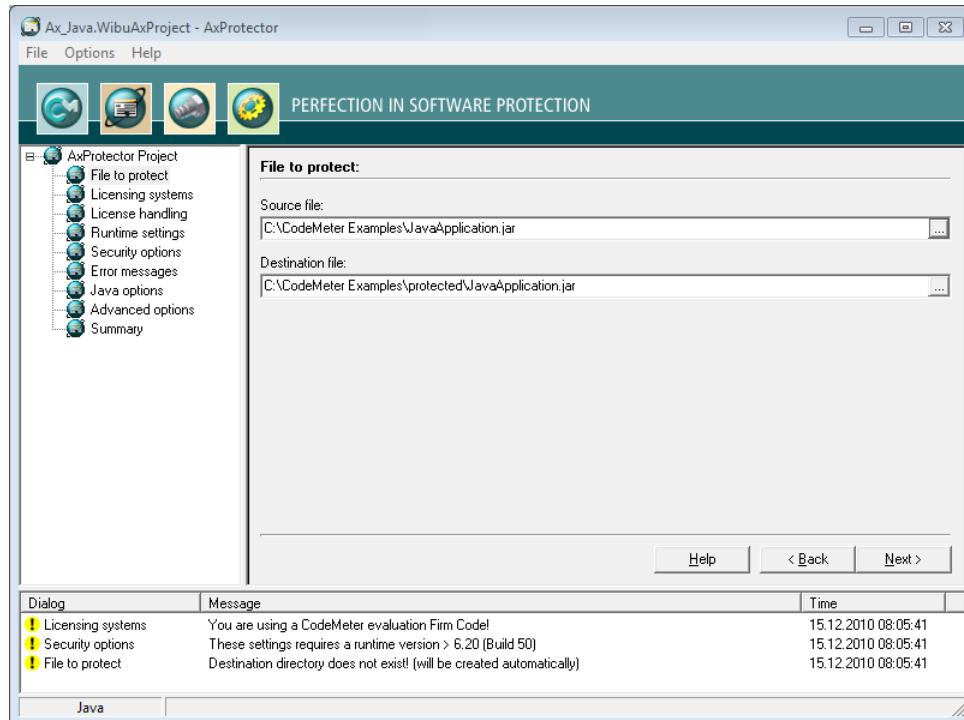


Figure 75: AxProtector - Java "File to Protect"

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 <small>²⁸⁹</small> . |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

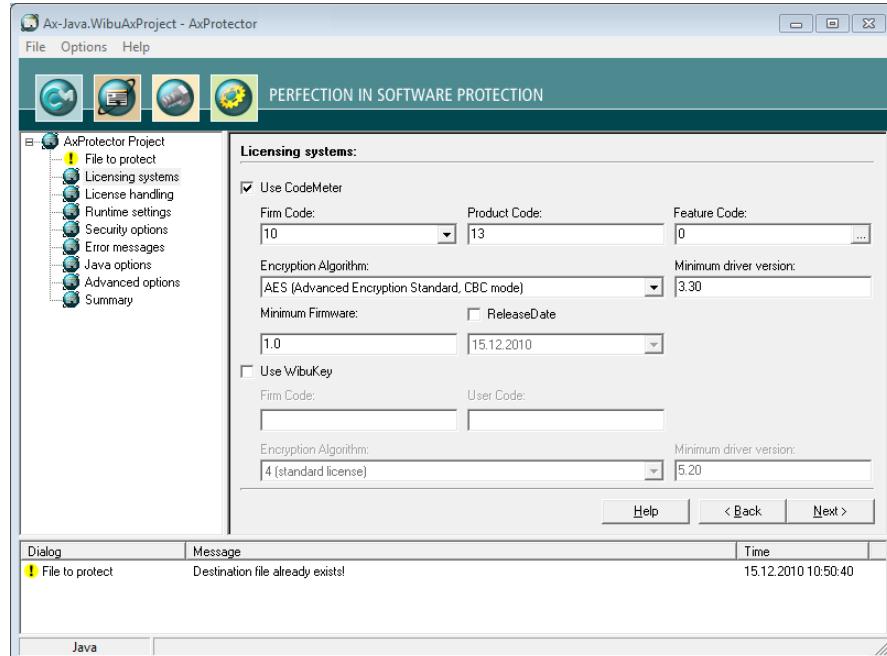
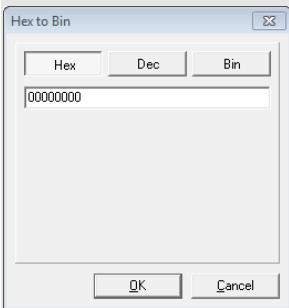


Figure 76: AxProtector - Java "Licensing Systems"

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|--------------|---|
| Firm Code | Specify the Firm Code to be used for encrypting the software. The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation <i>Firm Code</i> found in the <i>CodeMeter® Software Development Kit (SDK)</i> . In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code..The test Firm Code for <i>CmActLicense</i> is 5010. As a registered licensor, you will be issued your own unique Firm Code(s). Commandline option see here ²⁷¹ . |
| 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 ²⁷¹ . |
| Feature Code | Enter the Feature Code which defines, for example, the encryption of different software versi- |

| Element | Description |
|------------------------|--|
| | <p>ons.</p> <p> 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.</p> <p>Using the "..." button you may enter the feature map value in hexadecimal, decimal or binary format.</p>  |
| | <p>Figure 77: AxProtector - Windows Feature Map Input</p> <p>Commandline option see here²⁷².</p> |
| Encryption Algorithm | <p>Select the algorithm to encrypt your software. Currently, <i>CodeMeter®</i> solely supports AES (Advanced Encryption Standard).</p> <p>Commandline option see here²⁷¹.</p> |
| Minimum Driver Version | <p>Enter the minimum driver version required for the installed <i>CodeMeter License Servers</i>. When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that <i>AxProtector</i> automatically handles sessions of the protected software, and each session is allocated one of the available licenses.</p> <p> 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.</p> <p>Commandline option see here²⁷².</p> |
| Release Date | <p>Starting with Firmware version 1.18 <i>CodeMeter®</i> supports the Product Item Option Maintenance Period⁴⁵</p> |
| Minimum Firmware | <p>Specify the minimum firmware version required. In order to use the Product Item Option Maintenance Period you require the firmware version 1.18.</p> <p>Commandline option see here²⁷².</p> |

WibuKey

For setting *WibuKey* options, see the separate "WibuKey Developer Guide".

7.4.4.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network or both. Moreover, you can define the license allocation (access) mode.

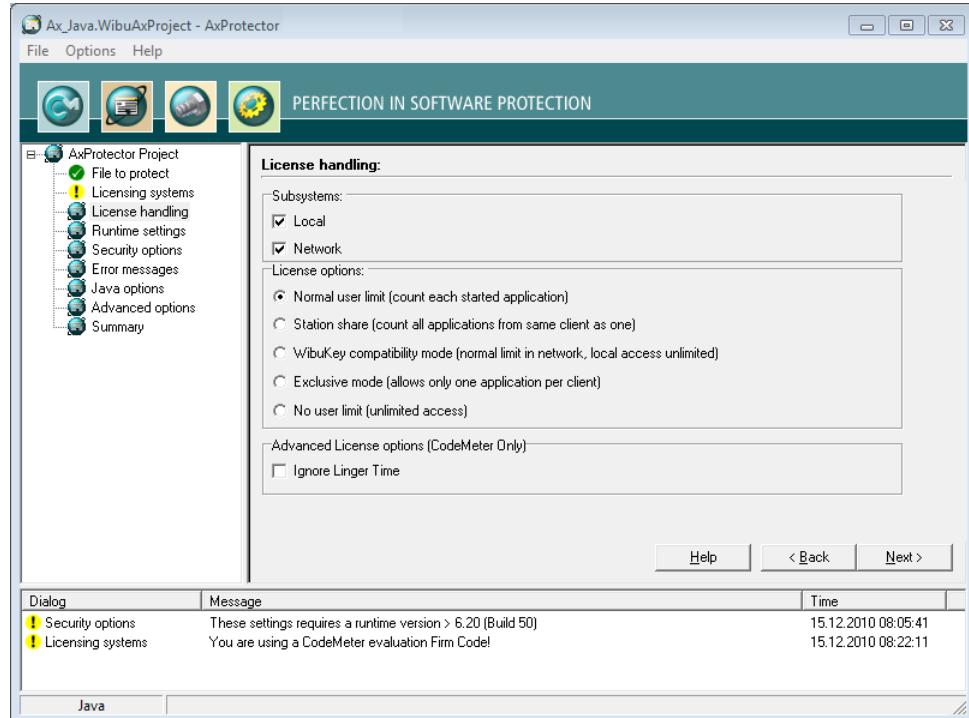


Figure 78: AxProtector - Java "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 on the network, i.e. only PCs are accessed where <i>CodeMeter License Server</i> runs and is activated as network server. On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

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](#)^[273]).

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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 <u>recommends</u> the setting 'normal user limit' and 'station share'. |
| Exclusive Mode | Here a protected application can be started only <u>once</u> 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. |

Linger Time

| Element | Description |
|--------------------|--|
| 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 <i>CodeMeter</i> Developer Guide). |

7.4.4.4 Runtime Settings

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

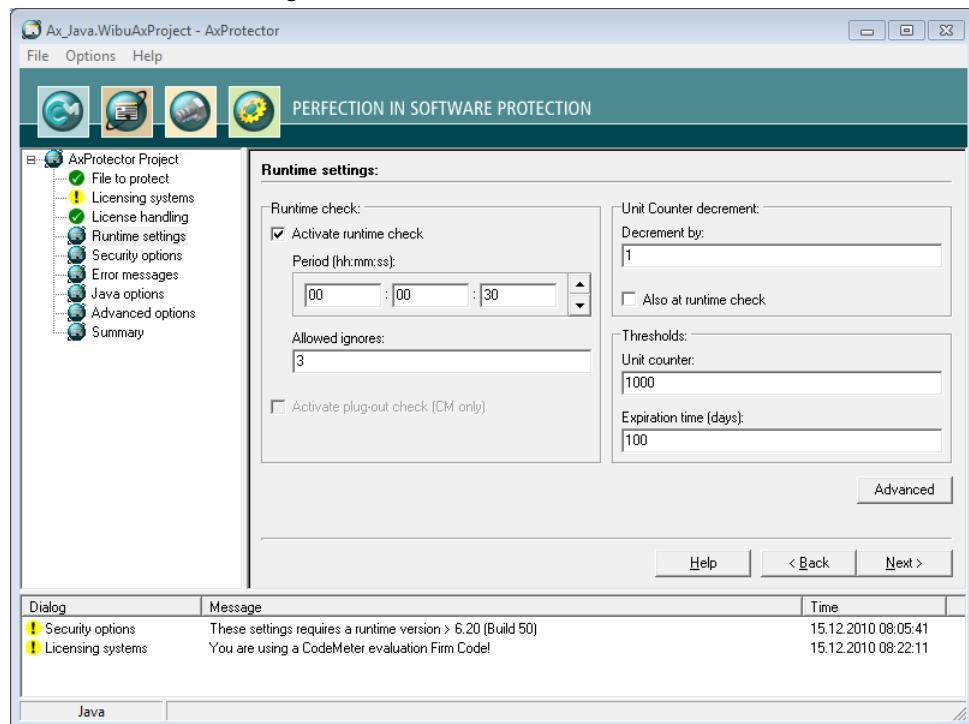


Figure 79: AxProtector - Java "Runtime Settings"

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 . |
| 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 <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;">  If the connection to a <i>CmContainer</i> 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. </div> |

Unit Counter Decrement

Decrementing an Unit Counter can serve to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see [here](#)²⁸⁴).

| 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 ¹⁶² " 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 ¹⁶⁸ . |
| Element | Description |
| Unit Counter | If the defined threshold falls short, a warning message is issued. Commandline option see here ²⁸⁶ . |
| 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 ²⁸⁵ . |

7.4.4.4.1 Advanced Runtime Settings

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

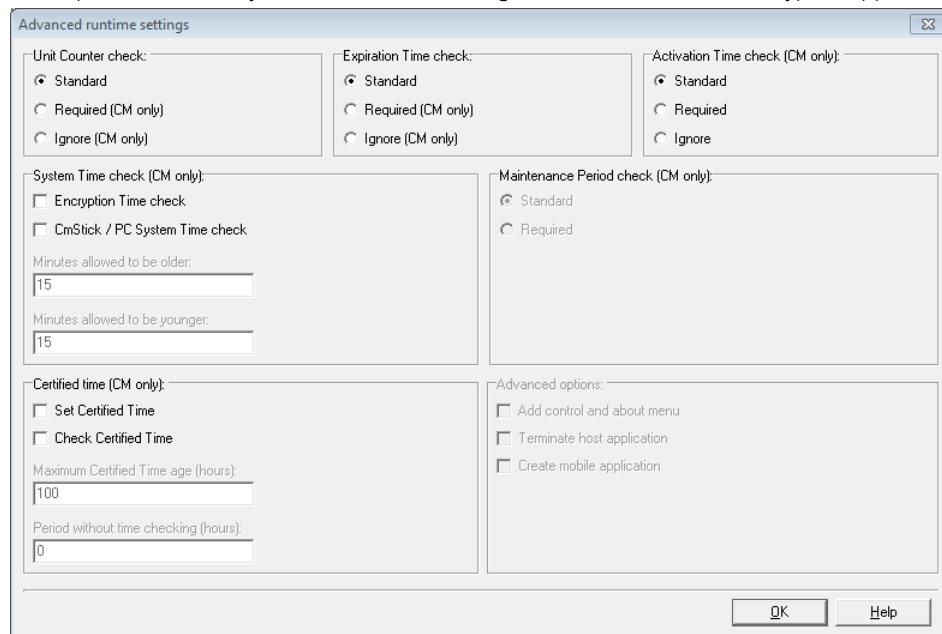


Figure 80: AxProtector - Java "Advanced Runtime Settings"

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](#)²⁸⁴).

| Element | Description |
|----------|--|
| Standard | Decrement 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 |

| Element | Description |
|---------|---|
| | 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](#)²⁸⁴).

| 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](#)²⁸³).

| 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 ²⁸⁴ 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](#)²⁸⁴).

| |
|---|
|  The option is available only, if you activated the checkbox Release Date on the page "Licensing systems" ¹⁵⁸ . |
|---|

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](#) ▶²⁷⁸).

 For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see [here](#) ▶²⁹⁴ ..

| Element | Description |
|--------------------------------------|---|
| Set Certified Time | <p>This option attempts to update the <i>Certified Time</i> in a <i>CmDongle</i>. The <i>certified time</i> is requested from the Time Server.</p> <p> This option requires a connection to the Internet.</p> |
| Check Certified Time | <p>This option checks to see if the <i>Certified Time</i> is older than the 'Maximum Certified Time Age' you defined here. If the 'Maximum Certified Time Age' is exceeded, the application will not start.</p> |
| Maximum Certified Time Age (hours) | <p>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 <i>Certified Time</i>.</p> |
| Period without time checking (hours) | <p>Specifies the period (in hours) when <u>no</u> check of the <i>Certified Time</i> certificate is taking place.</p> <p>If this period is not reached, a check is not performed. If the <i>Certified Time</i> certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the <i>Certified Time</i> 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 <i>Certified Time</i> certificate required.</p> |

System Time

In this area you define settings for additional protection preventing license manipulation by faked PC Time setting (commandline option see [here](#) ▶²⁷⁴).

| Element | Description |
|------------------------------------|---|
| Encryption Time check | <p>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 <i>CmContainer</i> System Time is newer than the encryption time.</p> <p> Requires at least <i>CodeMeter® 4.10</i>.</p> |
| CmContainer / PC System Time check | <p>When activated these options define a time corridor in which a difference between <i>CmContainer</i> 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.</p> |
| Minutes to be allowed older | <p>States in minutes how much the PC Time is allowed to be older than the <i>CmContainer</i> System Time.</p> |
| Minutes to be allowed younger | <p>States in minutes how much PC Time is allowed to be younger than the <i>CmContainer</i> System Time.</p> |

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 (commandline options see [here](#)^[27]).

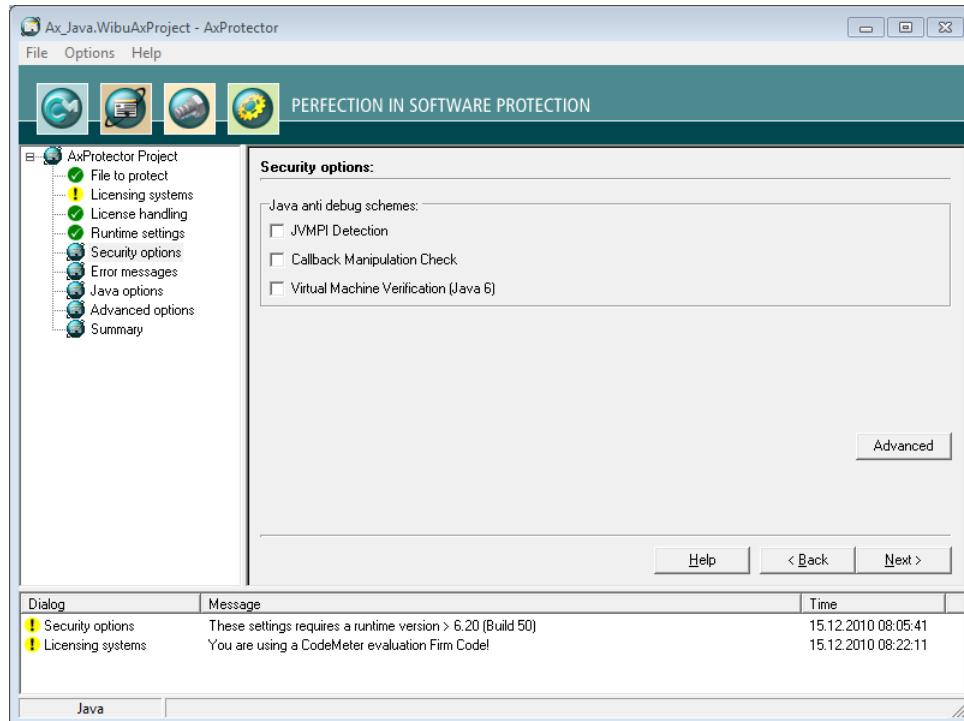


Figure 81: AxProtector - Java "Security Options"

| Element | Description |
|------------------------------|---|
| JVMPI Detection | Activating this checkbox starts the detection of the Java Virtual Machine Profiler Interface (JVMPI). Using JVMPI the Java Virtual Machine is manipulable sending messages to the native code. In particular, the event <code>JVMPI_EVENT_CLASS_LOAD_HOOK</code> 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. |
| VM Verification (Java 6 + 7) | Activating this checkbox checks for the correct Java Virtual Machine runtime environment for Java 6 and 7. |

7.4.4.6 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.

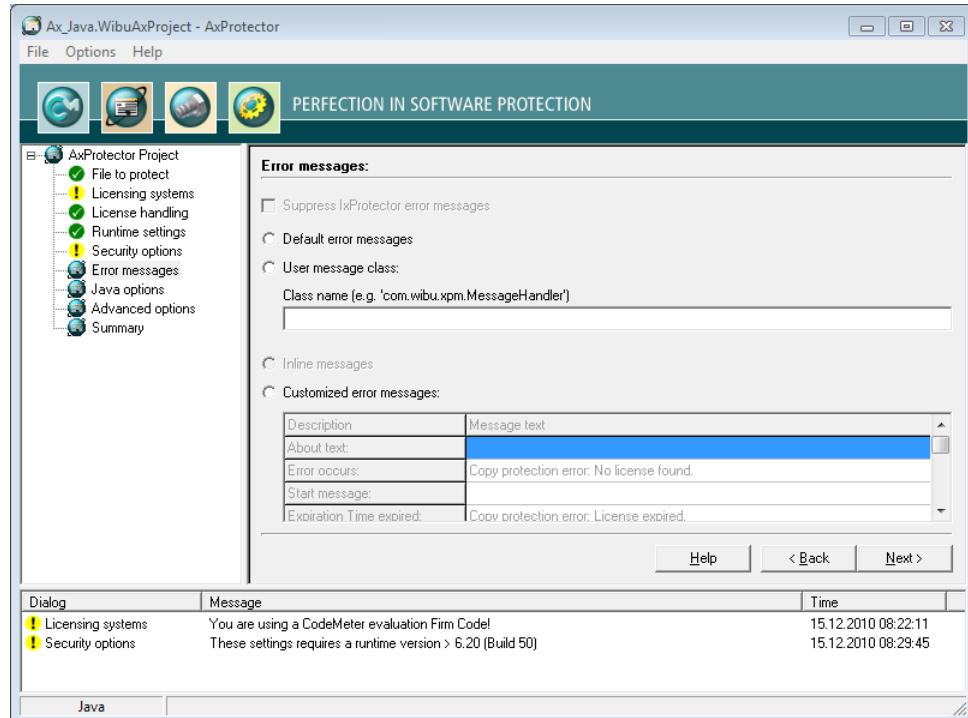


Figure 82: AxProtector - Java "Error Messages"

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 [287]). |
| 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. |

7.4.4.7 Java Options

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

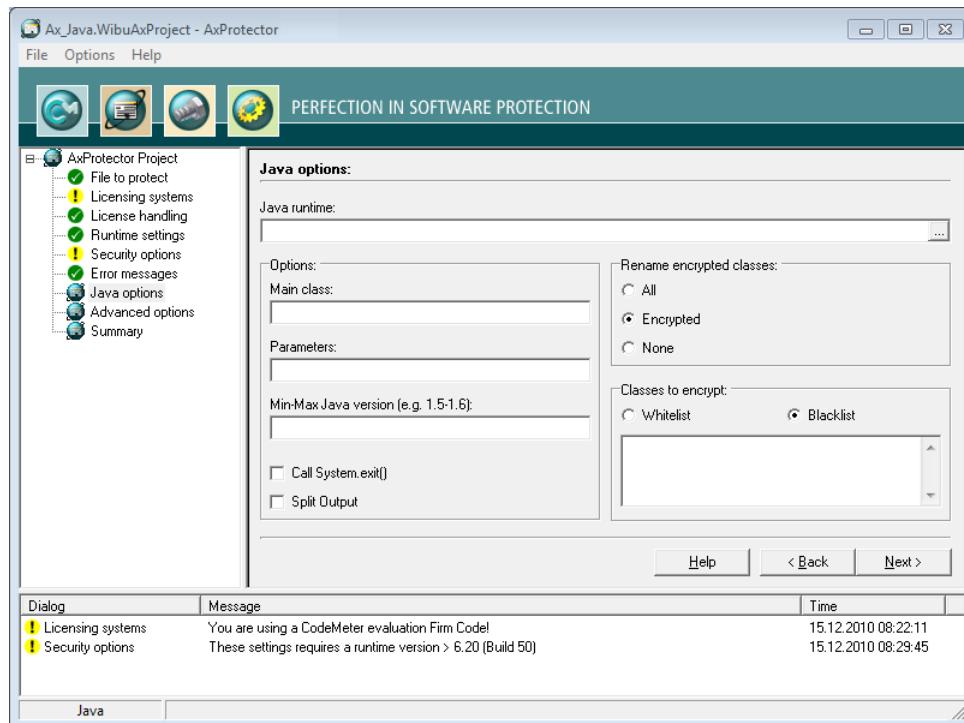


Figure 83: AxProtector - Java "Java Settings" Java Runtime (java.exe)

| Element | Description |
|-------------------------|--|
| Java Runtime (java.exe) | Using the "... " button specify the <code>java.exe</code> file of the installed runtime environment. |
| Main class | Enter here the name of the Java main class (commandline option see here ²⁹⁰). |
| Parameter | Define here the parameters for calling the Java main class (commandline option see here ²⁹⁰). |
| Minimum Java Version | Enter here the required minimum Java version (commandline option see here ²⁹⁰). <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  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. </div> |
| Call System.exit() | Activate this option to exit the application by the call of <code>System.exit()</code> after return to the Java main class. |

| Element | Description | | | | | | | | |
|--------------------------|---|---------|-------------|-----------|--|-----------|---|------|--|
| |  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 ²⁹²). | | | | | | | | |
| Split Output | Activate this option to save runtime classes to the separate <code>WibuXpm4Jruntime.jar</code> file (commandline option see here ²⁹¹).  Swapping the Wibu <code>ClassLoader</code> to a separate file increases performance of the protected application. Then even in the case of multiple encrypted classes, the Wibu <code>ClassLoader</code> will be only one-time loaded. | | | | | | | | |
| Rename encrypted classes | This group allows you to determine the classes which classes will be renamed, and loaded into the Wibu <code>ClassLoader</code> (commandline option see here ²⁹⁰).  For all class-related settings, the classes are renamed, and follow the pattern: <code><MyClass>.class.wibu</code> . | | | | | | | | |
| | <table border="1" data-bbox="284 618 1126 756"> <thead> <tr> <th data-bbox="284 618 425 643">Element</th><th data-bbox="425 618 1126 643">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="284 651 425 675">All</td><td data-bbox="425 651 1126 675">Activate this option to rename all existing classes.</td></tr> <tr> <td data-bbox="284 683 425 708">Encrypted</td><td data-bbox="425 683 1126 708">Activate this option to rename encrypted classes only.</td></tr> <tr> <td data-bbox="284 716 425 740">None</td><td data-bbox="425 716 1126 740">Activate this option to rename no classes.</td></tr> </tbody> </table>  When you rename encrypted classes only, only these classes are loaded by the Wibu <code>ClassLoader</code> . 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. | 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. |
| 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. | | | | | | | | |
| Classes to encrypt | This allows you to assign white or black list to classes (commandline option see here ²⁹¹). <table border="1" data-bbox="284 927 1126 1105"> <thead> <tr> <th data-bbox="284 927 425 951">Element</th><th data-bbox="425 927 1126 951">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="284 959 425 1040">Whitelist</td><td data-bbox="425 959 1126 1040">All classes referred to in the whitelist will be encrypted. This whitelist is saved to the jar-archive as an unencrypted text file <code>com/wibu/xpm/encrypted</code>.</td></tr> <tr> <td data-bbox="284 1049 425 1073">Blacklist</td><td data-bbox="425 1049 1126 1073">All classes referred to in the blacklist will not be encrypted.</td></tr> </tbody> </table>  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 <code>com.wibu.xpm.MessageHandler</code> . | 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 <code>com/wibu/xpm/encrypted</code> . | Blacklist | All classes referred to in the blacklist will not be encrypted. | | |
| 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 <code>com/wibu/xpm/encrypted</code> . | | | | | | | | |
| Blacklist | All classes referred to in the blacklist will not be encrypted. | | | | | | | | |

7.4.4.8 Advanced Options

This input window lets you set further encryption options.

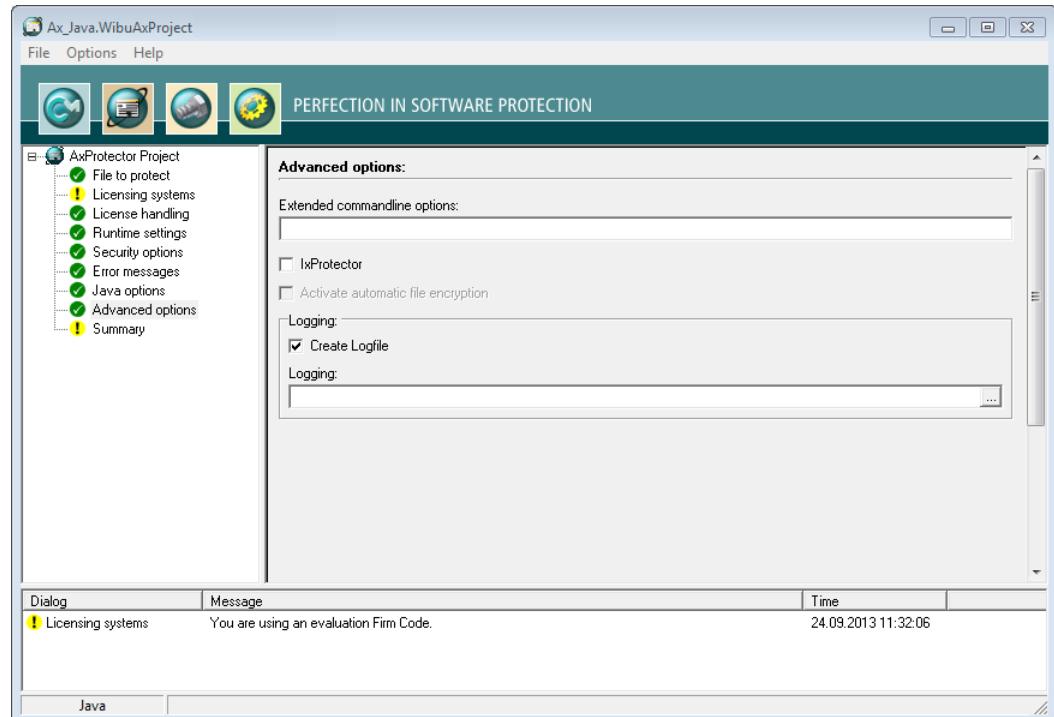


Figure 84: AxProtector - Java "Advanced Options"

| Element | Description |
|------------------------------|---|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the AxProtector cmdline.  For more information please contact support at Wibu-Systems. |
| Create Logfile | Activate this checkbox to create file logging for the activities of AxProtector. |
| 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 ²⁹⁰ . (commandline option see here ²⁹¹). |
| 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. |

7.4.4.9 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](#)²⁹⁵ type *AxProtector.exe @*.wbc*.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

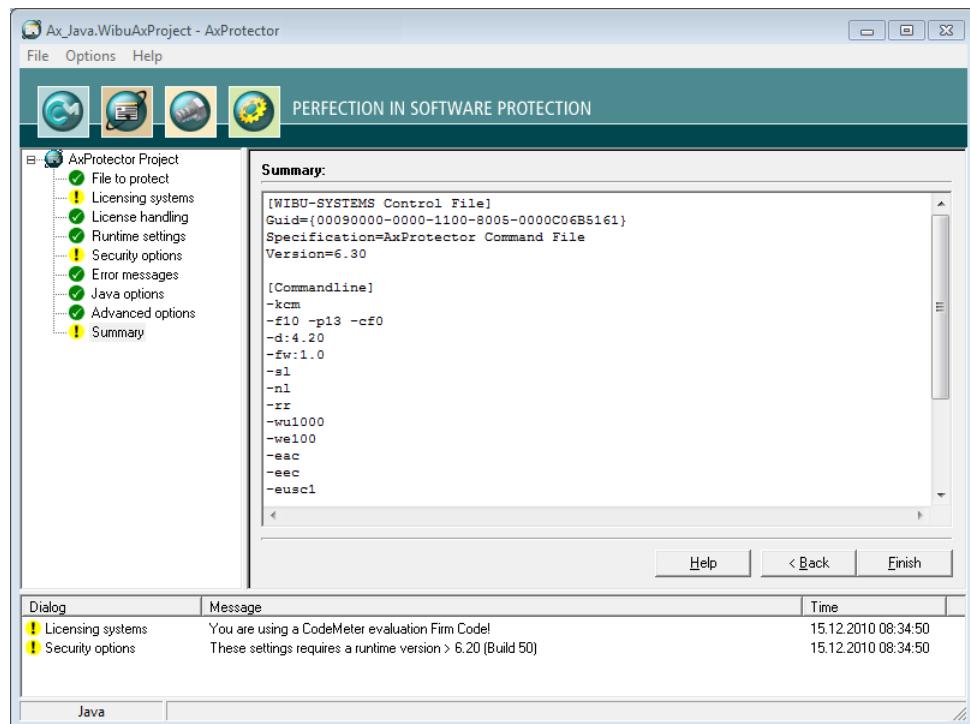


Figure 85: AxProtector - Java "Summary"

| Element | Description |
|---------|--|
| Finish | Starts the encryption using <i>AxProtector</i> 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.

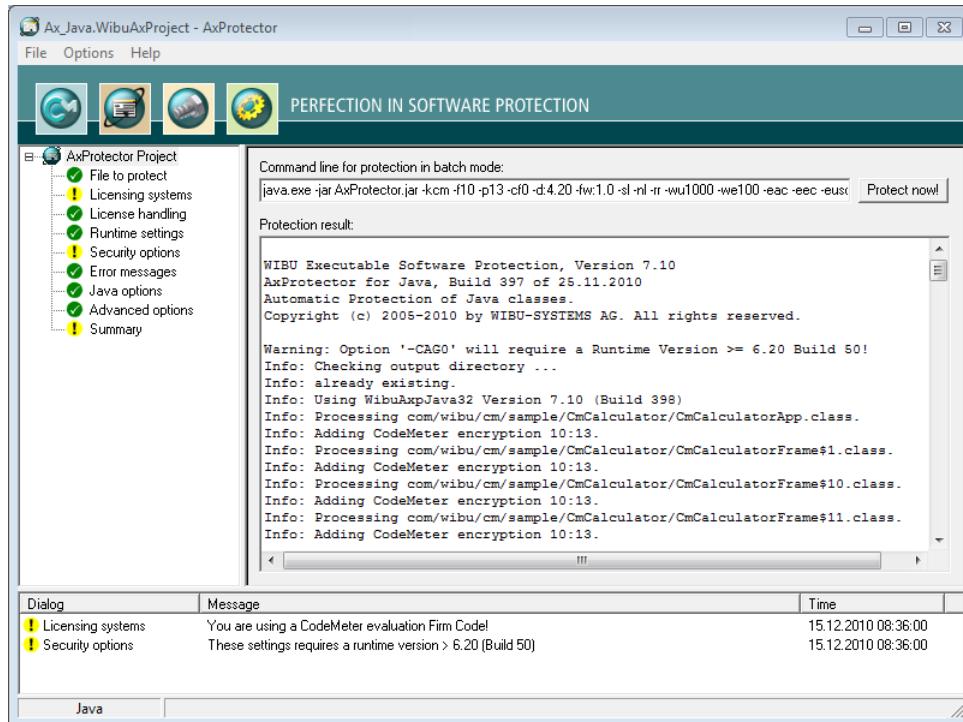


Figure 86: AxProtector - Java "Encryption Result"

| Element | Description |
|-------------|---|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now!" button. Then the AxProtector commandline is executed in batch mode.</p> <p>i 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.</p> |

7.4.5 Linux Application or Shared Object

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 ¹⁷⁴ | ✓ | Windows commandline ²⁷⁰ In a separate commandline for Linux, running on Linux operating systems, you are also able to insert encryption parameter ¹⁹⁸ . |

7.4.5.1 File to protect

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

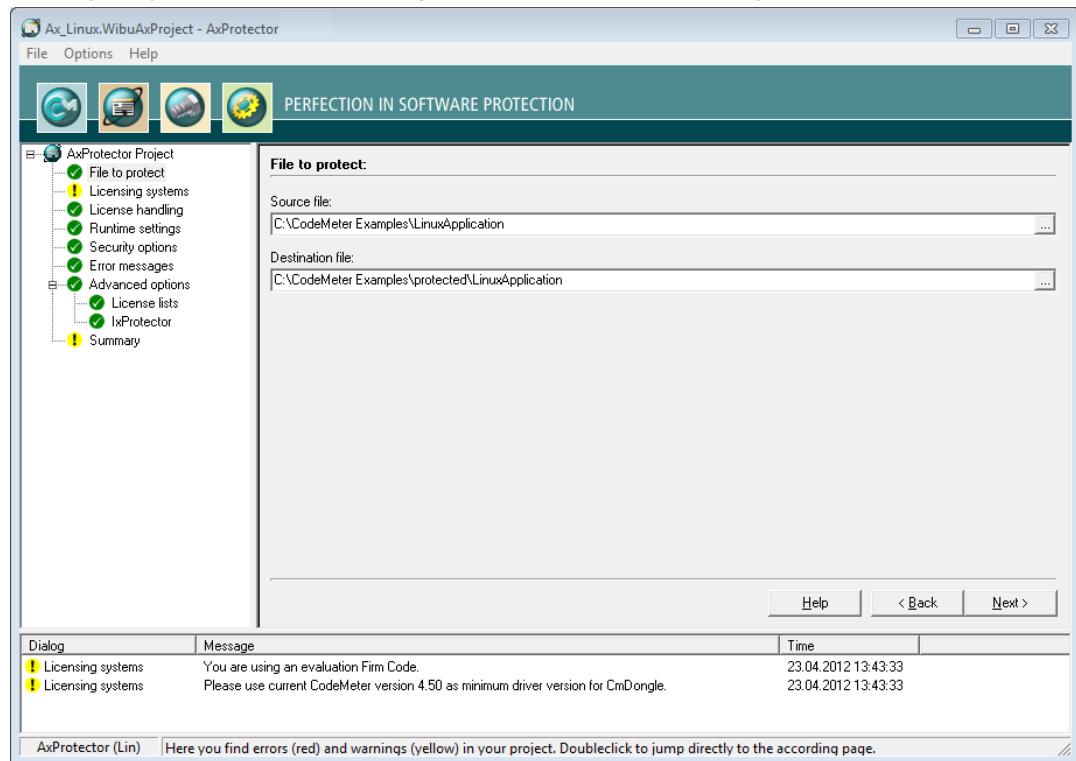


Figure 87: AxProtector - Linux "File to Protect"

File to Protect

| Element | Description |
|-------------|--|
| Source File | Click on the "... button and select the file to protect using the system dialog "Open". Alternatively, |

| Element | Description |
|------------------|---|
| | <p>manually specify the path and name of the file in this field.</p> <p> As alternative to the "..." button, you may also directly drag & drop the source file from Windows Explorer into the source file field.</p> |
| Destination File | <p>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.</p> <p>Commandline option see here²⁸⁹.</p> |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

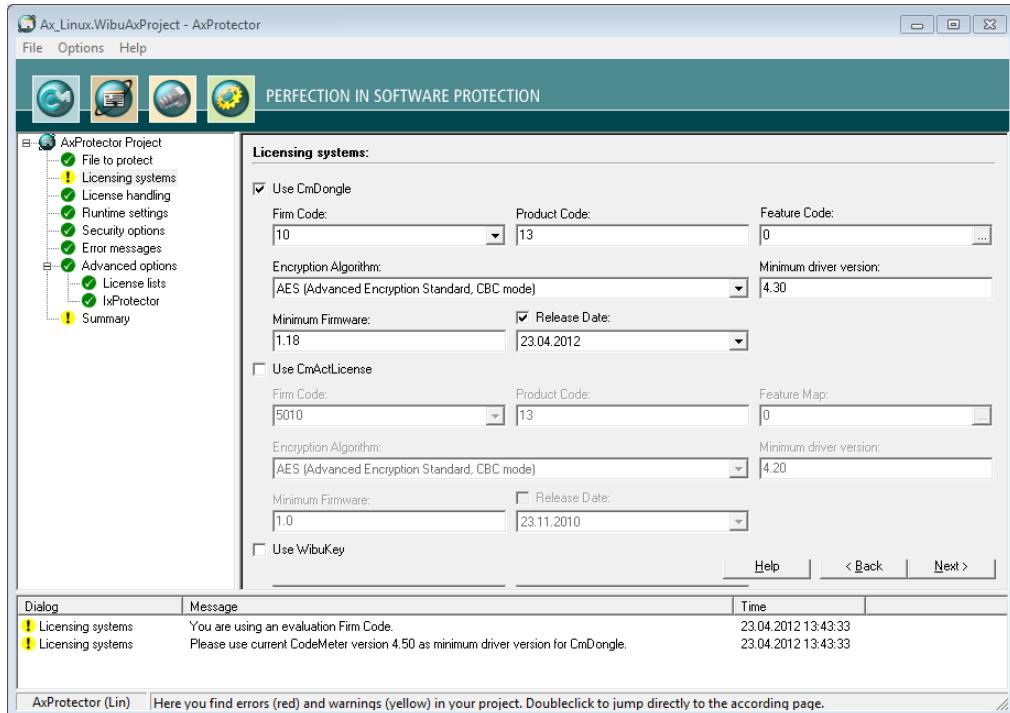
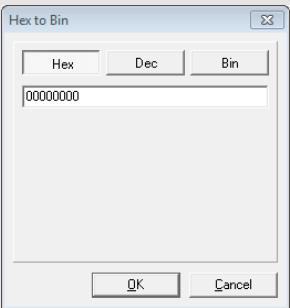


Figure 88: AxProtector - Linux "Licensing Systems"

 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.

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|--|--|
| Firm Code | <p>Specify the Firm Code to be used for encrypting the software.</p> <p> The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation <i>Firm Code</i> found in the <i>CodeMeter® Software Development Kit (SDK)</i>. In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code..The test Firm Code for <i>CmActLicense</i> is 5010. As a registered licensor, you will be issued your own unique Firm Code(s).</p> <p>Commandline option see here²⁷¹.</p> |
| Product Code | <p>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.</p> <p>Commandline option see here²⁷¹.</p> |
| Feature Code | <p>Enter the Feature Code which defines, for example, the encryption of different software versions.</p> <p> 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.</p> <p>Using the "..." button you may enter the feature map value in hexadecimal, decimal or binary format.</p> |
|  | Figure 89: <i>AxProtector</i> - Feature Map Input |
| Encryption Algorithm | <p>Select the algorithm to encrypt your software. Currently, <i>CodeMeter®</i> solely supports AES (Advanced Encryption Standard).</p> <p>Commandline option see here²⁷².</p> |
| Minimum Driver Version | <p>Enter the minimum driver version required for the installed <i>CodeMeter License Servers</i>. When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that <i>AxProtector</i> automatically handles sessions of the protected</p> |

| Element | Description |
|------------------|---|
| | <p>software, and each session is allocated one of the available licenses.</p> <p> 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.</p> <p>Commandline option see here²⁷².</p> |
| Release Date | Starting with Firmware version 1.18 CodeMeter® supports the Product Item Option Maintenance Period ⁴⁵ |
| 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. Commandline option see here ²⁷² . |

WibuKey

For setting WibuKey options, see the separate "WibuKey Developer Guide".

7.4.5.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network, or both. Moreover, you can define the license allocation (access) mode.

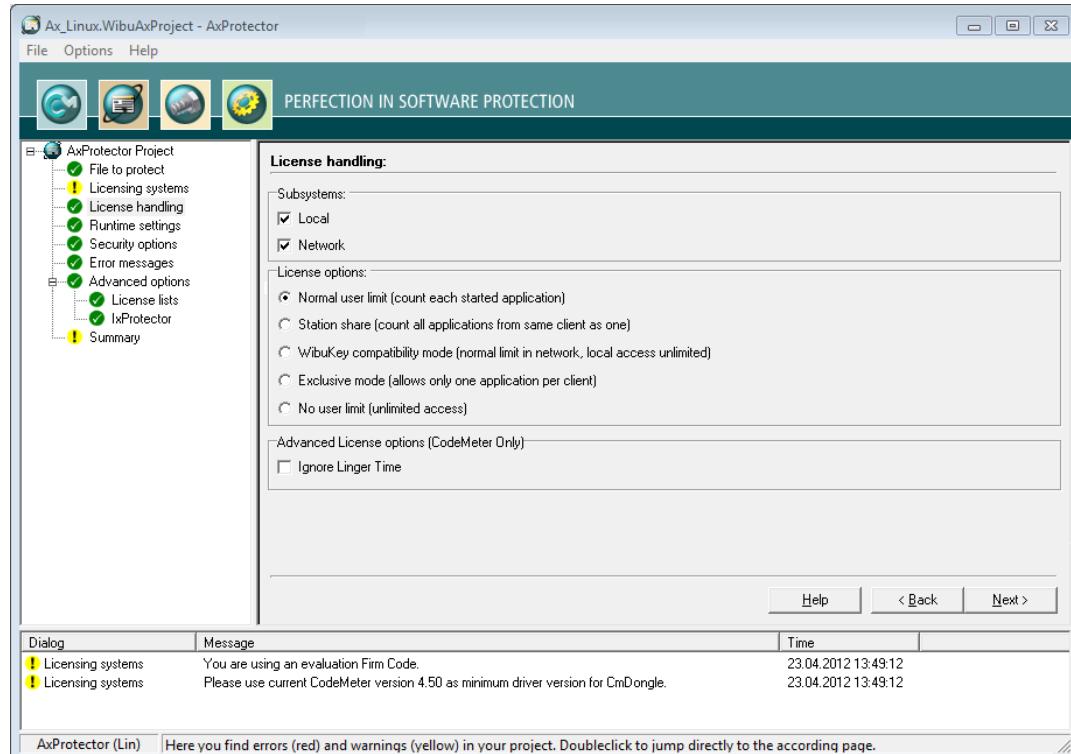


Figure 90: AxProtector - Linux "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 on the network, i.e. only PCs are accessed where <i>CodeMeter License Server</i> runs and is activated as network server. i On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

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](#)²⁷³).

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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. |

Linger Time

| Element | Description |
|--------------------|--|
| 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 license of a protected application has been released or the protected application has been closed (more Information in the <i>CodeMeter Developer Guide</i>). |

7.4.5.4 Runtime Settings

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

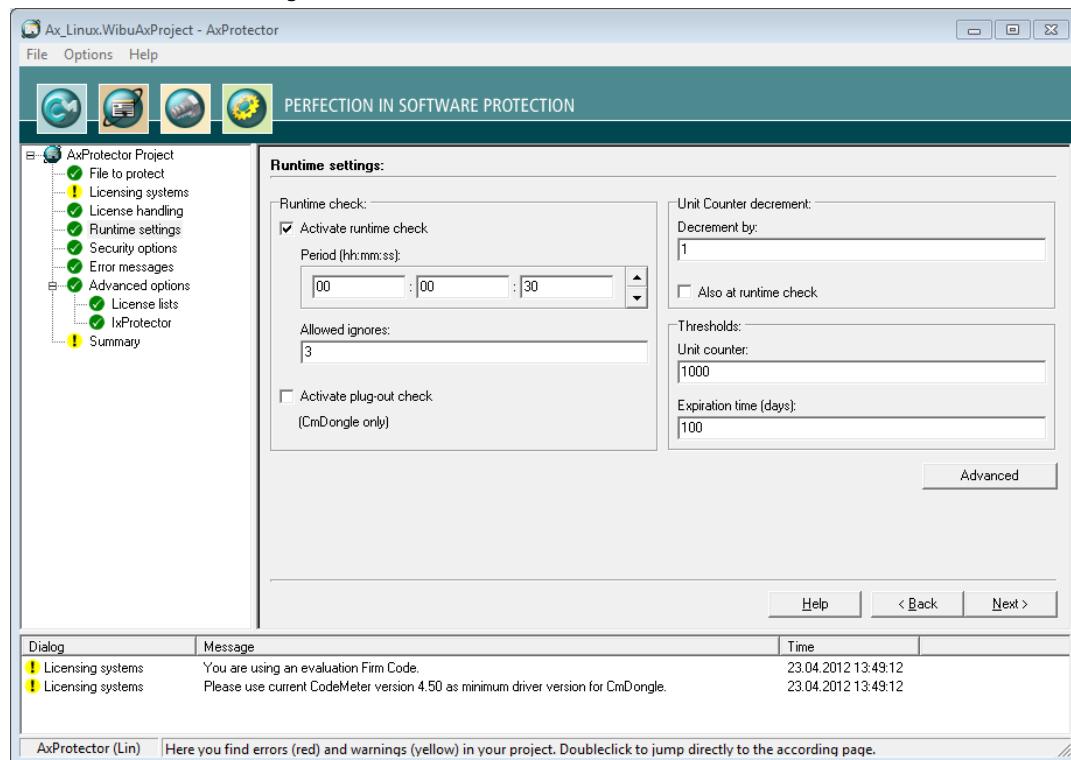


Figure 91: AxProtector - Mac OS X "Runtime Settings"

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 . |
| 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 <i>CmContainer</i> 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. |

| Element | Description |
|--|--|
| Activate Plug-out Check (only CmDongle) | This option closes the protected application if the <i>CmDongle</i> is removed while the application is running. Immediately, an error message is issued. This option is valid for <i>CmDongle</i> only. Commandline option see here ²⁷⁵ . |

Unit Counter Decrement

Decrementing an Unit Counter can serve to establish the validity of licenses in a *CmContainer*. This group allows you to define this behavior (commandline option see [here](#)²⁸⁴).

| 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 ¹⁴¹ . |
|---|--|

| Element | Description |
|------------------------|---|
| Unit Counter | If the defined threshold falls short, a warning message is issued. Commandline option see here ²⁸⁶ . |
| 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 ²⁸⁵ . |

7.4.5.4.1 Advanced Runtime Settings

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

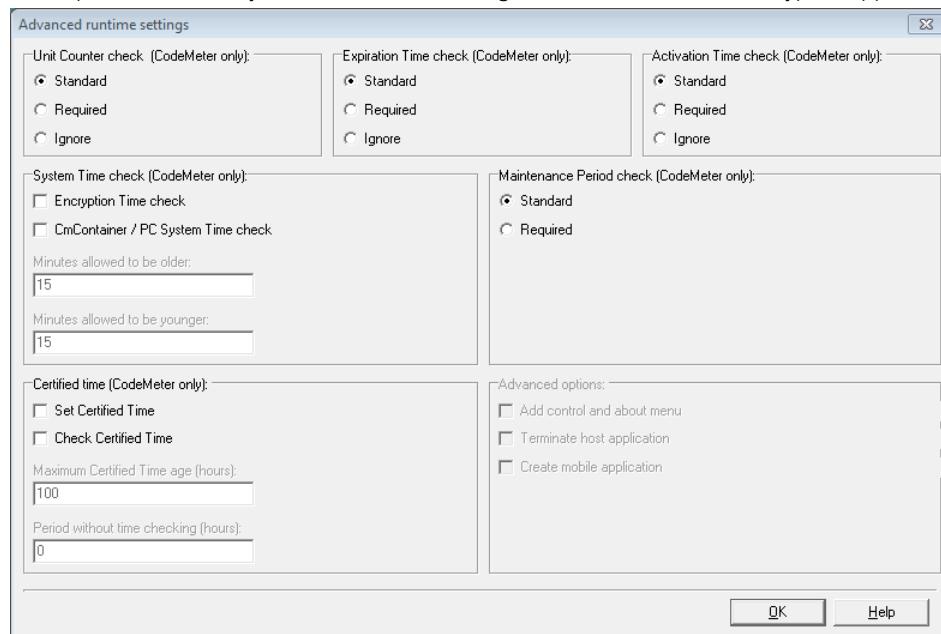


Figure 92: AxProtector - Linux "Advanced Runtime Settings"

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](#)²⁸⁴).

| Element | Description |
|----------|---|
| Standard | Decrement 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 |

| Element | Description |
|---------|---|
| | 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](#)²²⁴).

| 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](#)²²³).

| 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 ²³⁴ 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](#)²²⁴).

| |
|---|
|  The option is available only, if you activated the checkbox Release Date on the page "Licensing systems" ¹⁷⁷ . |
|---|

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](#)  ²⁷⁸).

 For information on the fail safe and manipulation safe processes referring to *Activation* and *Expiration Time* see [here](#)  ²⁹⁴ ..

| Element | Description |
|--------------------------------------|--|
| Set Certified Time | This option attempts to update the <i>Certified Time</i> in a <i>CmDongle</i> . The <i>Certified Time</i> is requested from the Time Server.  This option requires a connection to the Internet. |
| Check Certified Time | This option checks to see if the <i>Certified Time</i> 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 <i>Certified Time</i> . |
| Period without time checking (hours) | Specifies the period (in hours) when <u>no</u> check of the <i>Certified Time</i> certificate is performed. If this period is not reached, a check is not performed. If the <i>Certified Time</i> certificate is located between this period and the 'Maximum Certified Time Age', an attempt to update the <i>Certified Time</i> 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 <i>Certified Time</i> 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](#)  ²⁷⁴).

| 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 <i>CmContainer</i> System Time is newer than the encryption time.  Requires at least <i>CodeMeter® 4.10</i> . |
| CmContainer / PC System Time check | When activated these options define a time corridor in which a difference between <i>CmContainer</i> 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 <i>CmContainer</i> System Time. |
| Minutes to be allowed younger | States in minutes how much PC Time is allowed to be younger than the <i>CmContainer</i> System Time. |

7.4.5.5 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.

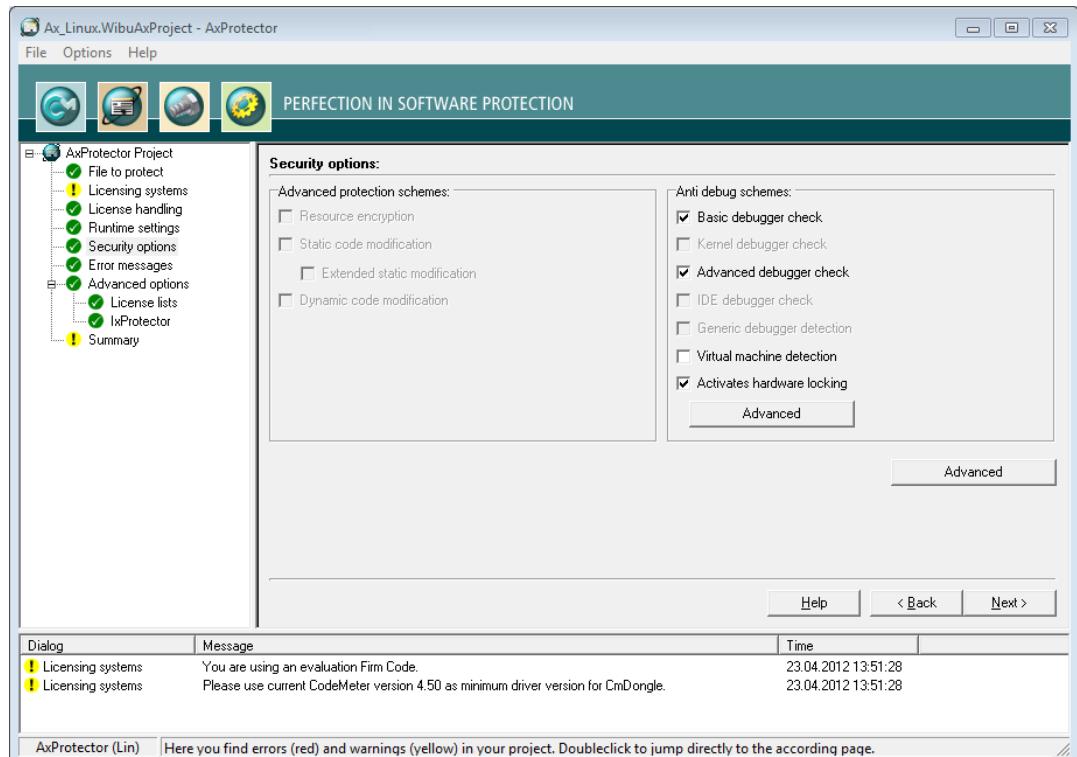
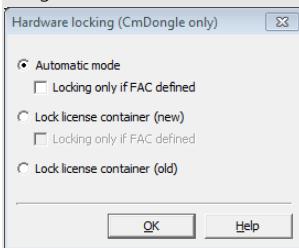


Figure 93: AxProtector - Linux "Security Options"

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](#)²⁷⁵).

| 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 | Checks in an advanced search for debugger programs which may run parallel to your application. |

| Element | Description | | | | | | |
|---|---|------------------|-------------|--------------------------|---|-------------------------|---|
| Check | cation, 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 <i>CmContainer</i> 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 <i>CodeMeter</i> . | | | | | | |
| 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. <table border="1"> <thead> <tr> <th>Locking Scenario</th><th>Description</th></tr> </thead> <tbody> <tr> <td>immediate locking</td><td>is performed starting with Firmware Version 1.14 as soon as a debugger is detected.</td></tr> <tr> <td>prepared locking</td><td>is performed by checking the Firm Access Counter (FAC). The Firm Access Counter locates at the Firm Item level of a <i>CmContainer</i>. 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. 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.</td></tr> </tbody> </table> | 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 <i>CmContainer</i> . 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. 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. |
| 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 <i>CmContainer</i> . 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. 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 94: AxProtector - Mac OS "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 is smaller than 1.14, in the scenario prepared locking the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, the Firm Items is immedi- | | | | | | |

| Element | Description | |
|---------|--|---|
| | Option | Description |
| | | tely locked. Due to compatibility reasons this corresponds to the default setting. |
| | "Automatic Mode" activated and "Locking only if FAC defined" activated | If the Firmware is smaller than 1.14, the FAC is decremented by the value of 1. If the Firmware 1.14 and higher, then at the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. |
| | "Lock License Container (new)" activated and "Locking only if FAC defined" not activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. Seen from a security view this is the recommended option. This, however, requires that all <i>CmContainer</i> in the field must have a Firmware Version 1.14 an higher. |
| | "Lock License Container (new)" and "Locking only if FAC defined" activated | If the Firmware 1.14 and higher, the Firm Items is immediately locked. At the same time it is checked whether a prepared locking is programmed. If a locking is prepared, the Firm Items is locked. |
| | "Lock License Container (old)" activated | For all Firmware versions. If a prepared locking programs the FAC is decremented by a value of 1. |

7.4.5.5.1 Advanced Security Options

This input window lets you define further settings.



Figure 95: AxProtector - Linux "Advanced Security Options"

Advanced settings

This area allows for setting additional options.

| Element | Description |
|------------------------------------|--|
| Add virus check | Adds a virus check to the protected application by using a check sum (commandline option see here ²⁷⁸). |
| Link API statically to Application | The <i>CodeMeter Core API</i> is statically linked to the protected application. This option increases security but also increases the sizes of the executable file (commandline option see here |

| Element | Description |
|-------------------------------|--|
| Size of encrypted Code (in %) | Specifies the portion of the code to be encrypted stated as percentage number (commandline option see here ²⁷⁸). |

7.4.5.6 Error Messages

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

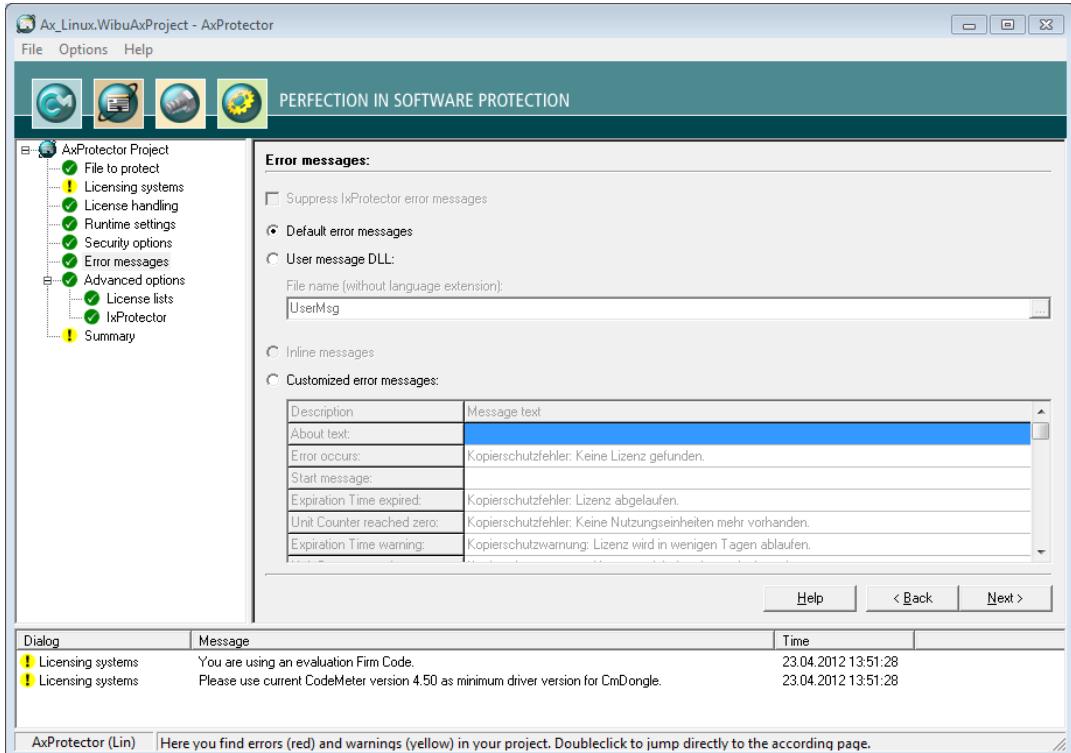


Figure 96: AxProtector - Linux "Error Messages"

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 ²⁸⁷). |
| Customized Error Messages | Activate this option to enter customized error messages displayed in the message boxes below. |

7.4.5.7 Advanced Options

This input window lets you set further encryption options.

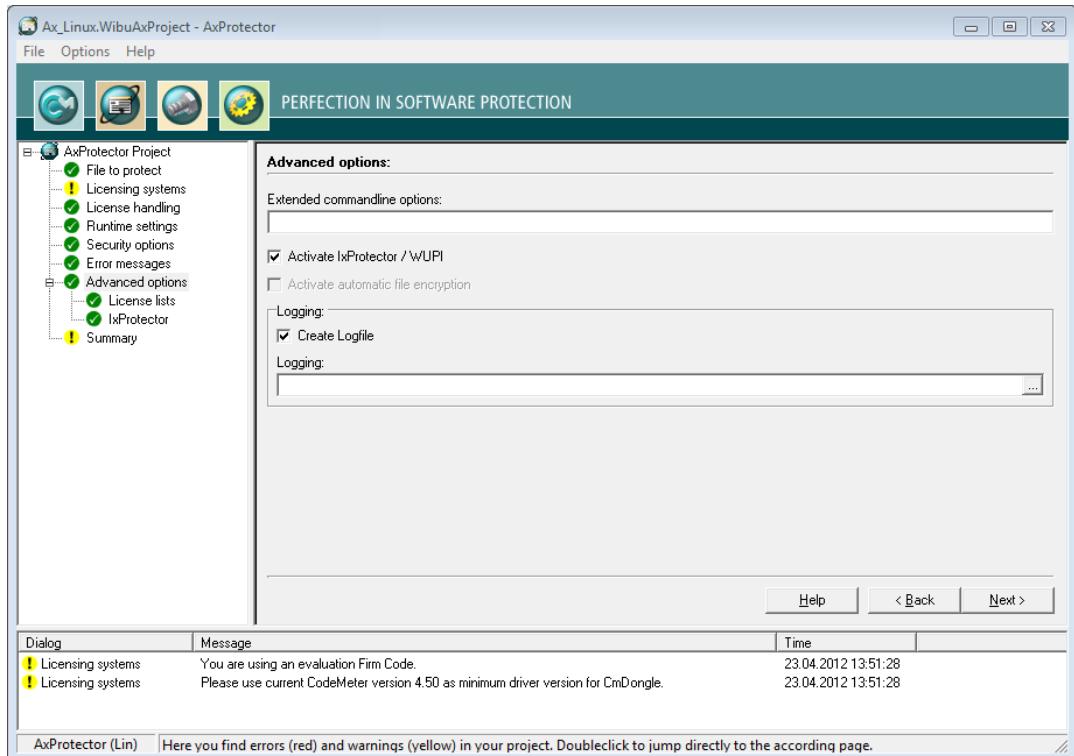


Figure 97: AxProtector - Linux "Advanced Options"

| Element | Description |
|------------------------------|--|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the AxProtector commandline. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i For more information please contact support at Wibu-Systems. </div> |
| 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 ²⁰⁶ . (commandline option see here ²⁰¹). |
| Create Logfile Logging | Activate this checkbox to create file logging for the activities of AxProtector. Specify the path and file name of this log file. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

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 \(WUPI\)](#)²⁹⁶.

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

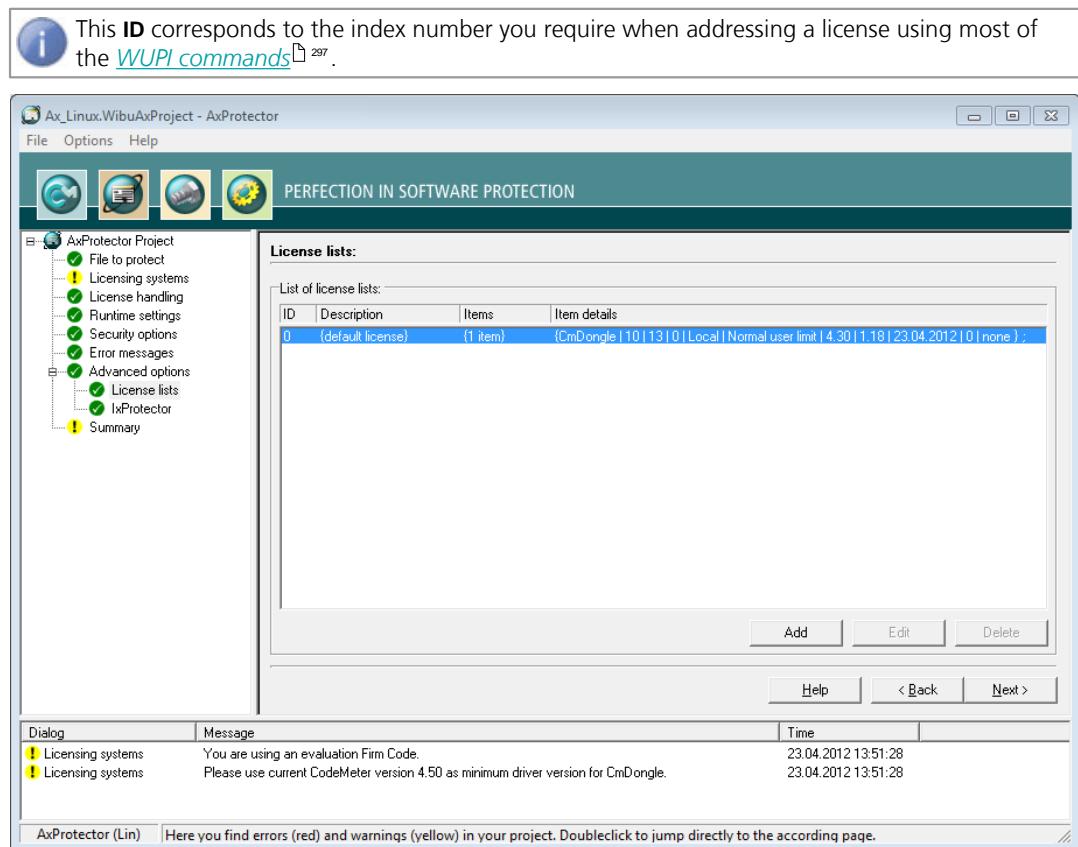
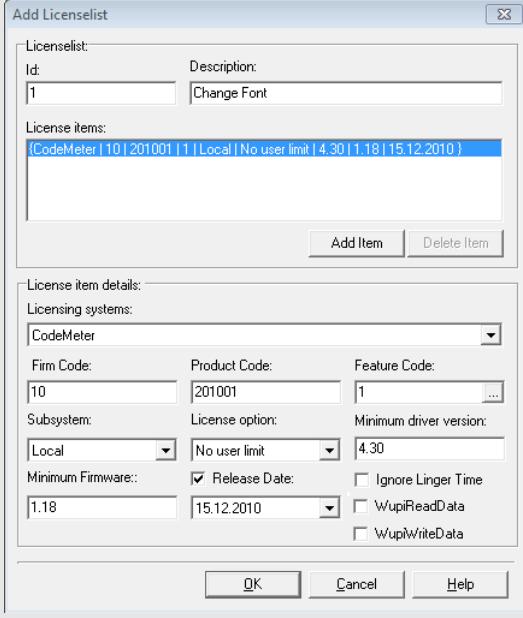


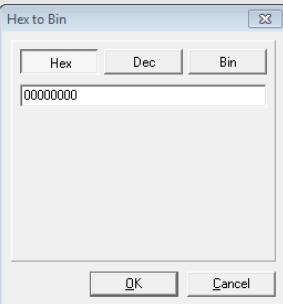
Figure 98: AxProtector Linux - "License Lists"

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. |

| Element | Description |
|-------------------|--|
| |  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 . |
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  |
| | Figure 99: AxProtector - Linux - "Add License Lists" |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). |
| 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. Using the "..." button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. |

| Element | Description |
|------------------------|--|
| |  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>This license option allows to define an allocation time of the license after the license of a protected application has been released or the protected application has been closed (more Information in the <i>CodeMeter Developer Guide</i>).</p> |
| WupiReadData | Activate this option to read data ³⁰⁰ from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

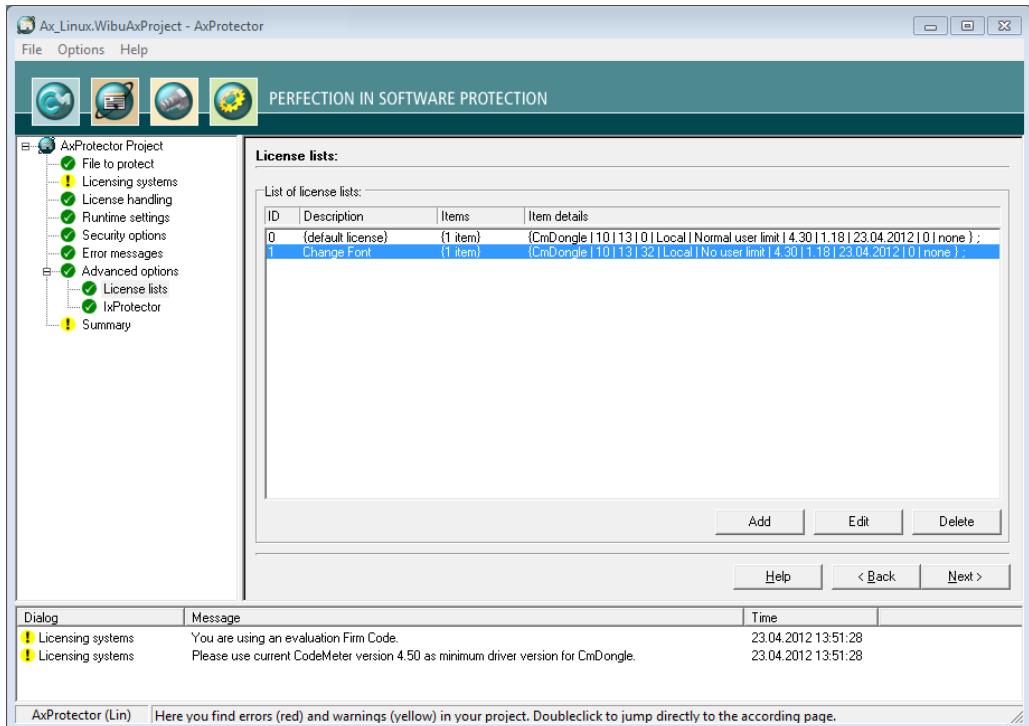


Figure 100: AxProtector - Linux - "Completed License Lists"

7.4.5.7.2 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.

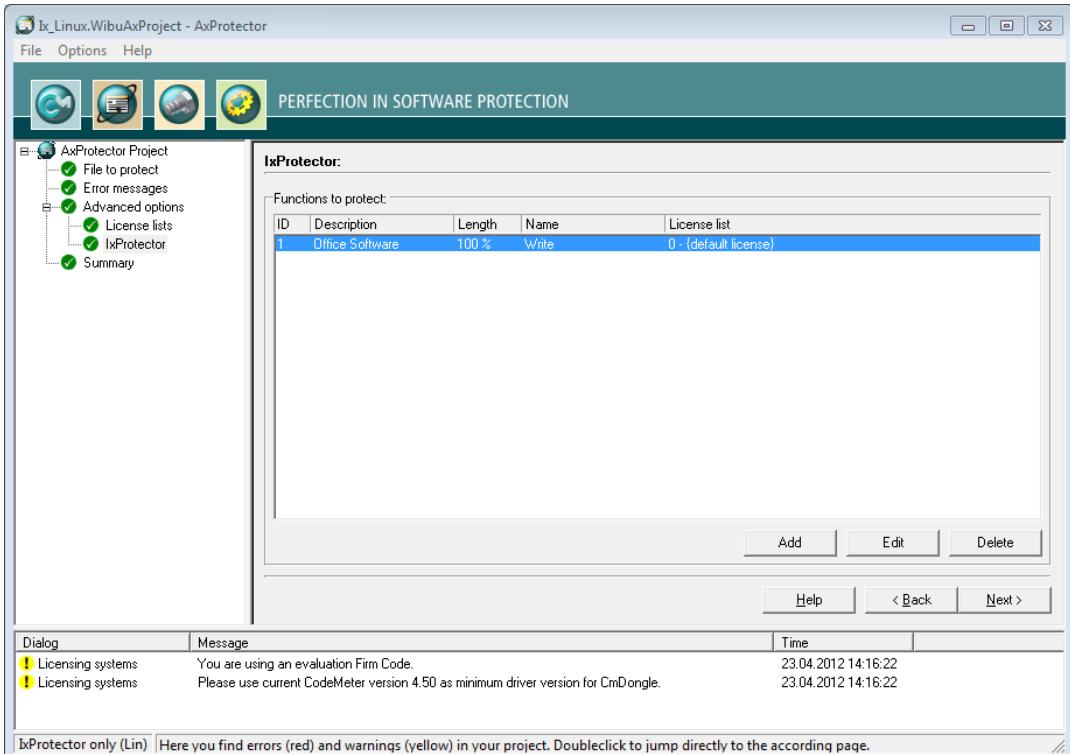
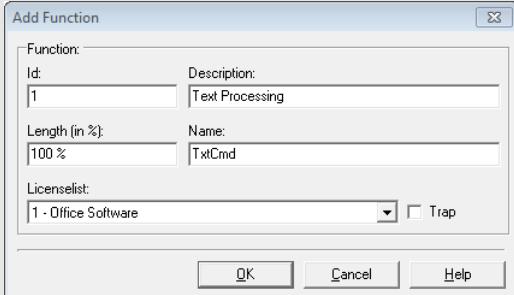


Figure 101: AxProtector - Linux - "Function List"

| Element | Description |
|----------------------|---|
| Functions to protect | <p>Lists all specified function lists, including all properties.</p> <p>This menu item lets you also create function lists. Please proceed as follows:</p> <ol style="list-style-type: none"> 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 102: AxProtector - Linux - "Add Function" | |
| Element | Description |
| Id | <p>Uniquely identifies the function.</p> <p>ⓘ This Id corresponds to the identification you use when calling the WUPI commands WupiDecryptCode²⁹⁸ and WupiEncryptCode²⁹⁹.</p> |
| Description | Enter a description of the function with text. |
| Length | <p>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.</p> <p>ⓘ If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.</p> |
| Name | <p>Specify the name of the function to be encrypted.</p> <p>The function name must exactly match the name used in the export list of the linked map file.</p> <p>Please note the correct spelling (case sensitive, underline, etc.). Microsoft Dependency Walker shows dependencies between 32-or 64-bit Windows PE files. A tree view shows all linked modules and imported and exported functions are displayed in tables. Dependency Walker is part of Windows XP SP2 Support Tools and also part of Microsoft Visual Studio including version 8.0 (Visual Studio 2008, i.e. version 9.0 no longer provides Dependency Walker).</p> |
| 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. Command line option see here ²⁸⁵ . |
| 3. Click the "OK" button. The new functions are added to the function list. | |

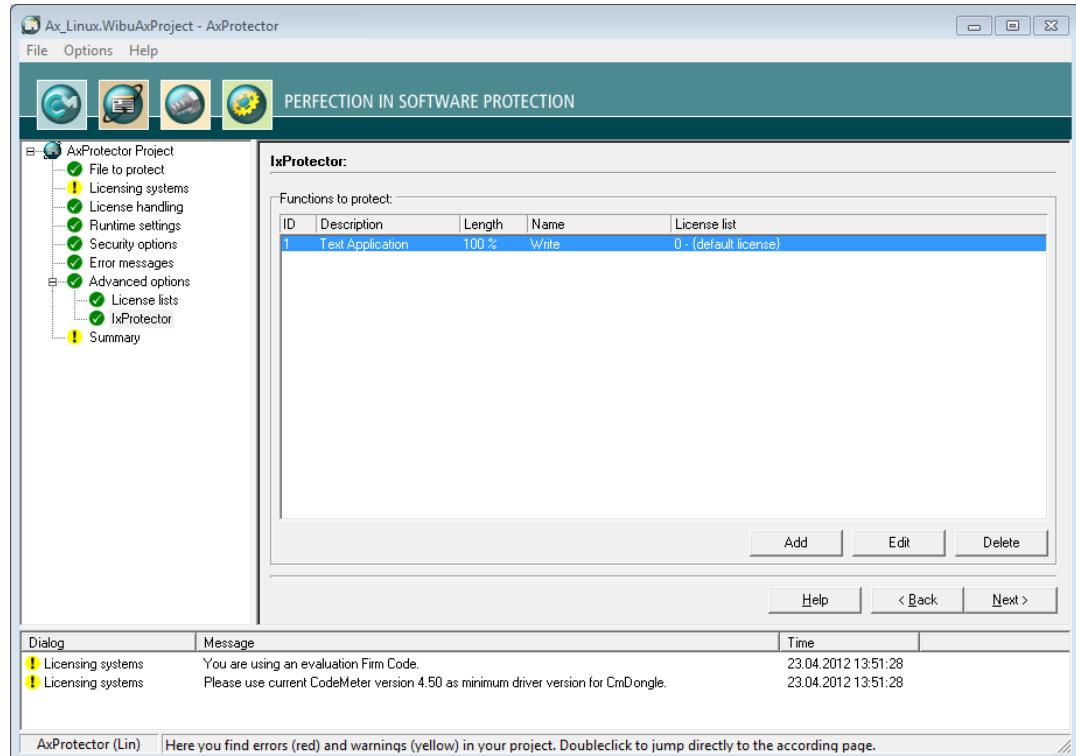


Figure 103: AxProtector - Linux - "Completed Function List"

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](#)²⁹⁵ type AxProtector.exe @*.wbc.

Alternatively, using the **"File - export wbc file"** menu item, you can also create the corresponding *.wbc file.

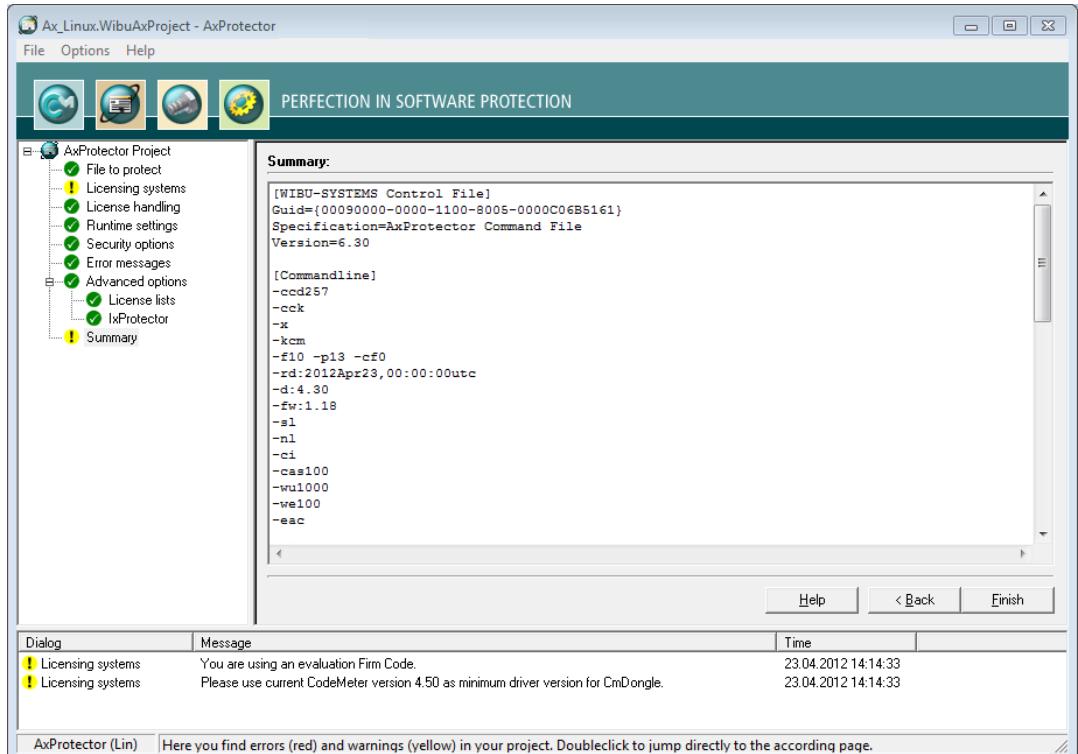


Figure 104: AxProtector - Linux "Summary"

| 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.

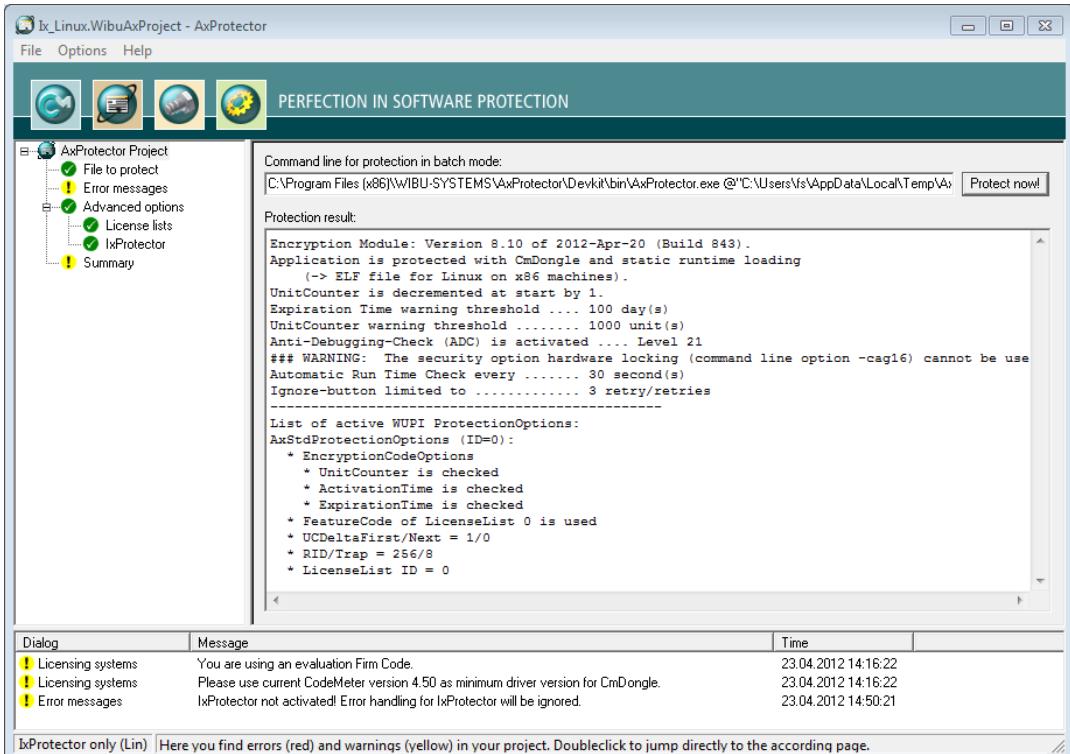


Figure 105: AxProtector - Linux "Encryption Result"

| Element | Description |
|-------------|---|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now!" button. Then the AxProtector commandline is executed in batch mode.</p> <p>i 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.</p> |

7.5 IxProtector Tab

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 <small>↳ 200</small> | ✓ | Windows commandline <small>↳ 270</small> |

7.5.1.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

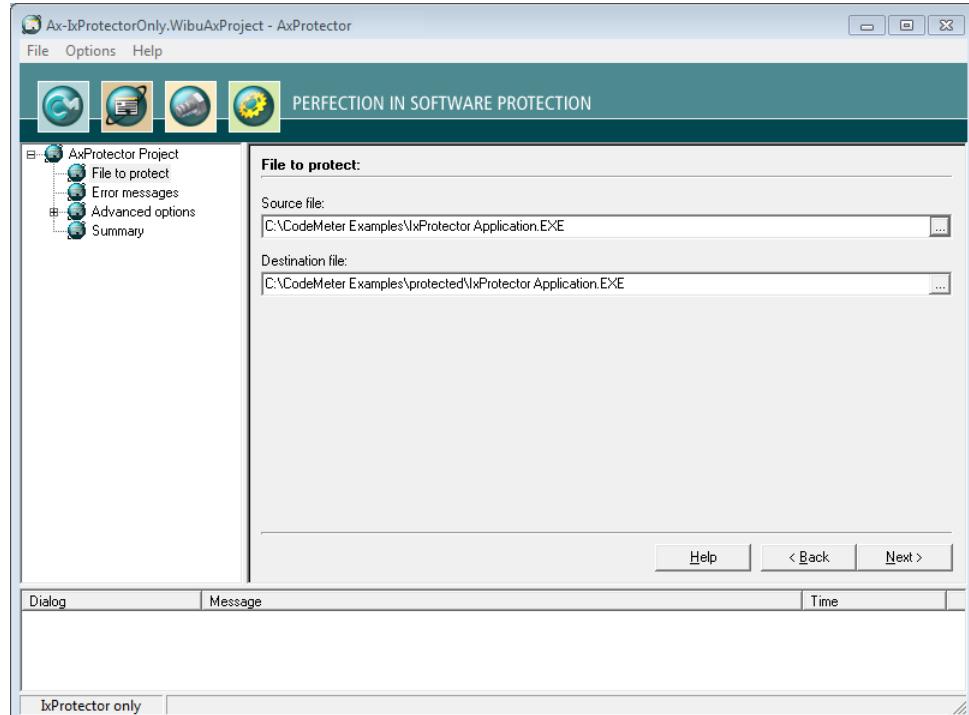


Figure 106: *AxProtector - IxProtector only Windows "File to Protect"*

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. i 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, <i>AxProtector</i> 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 . |

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.

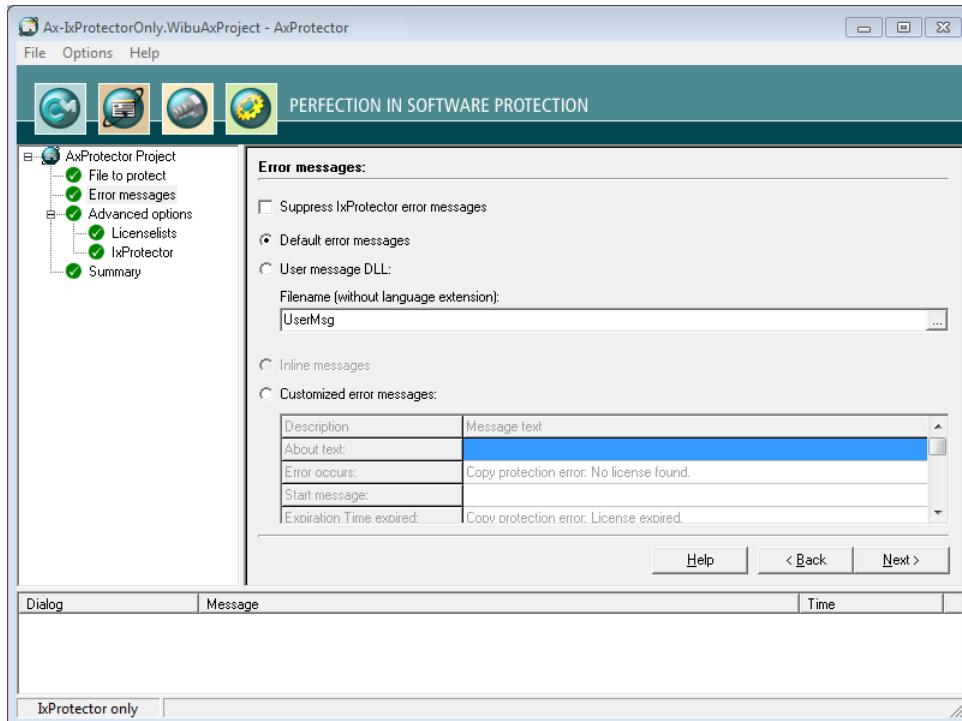
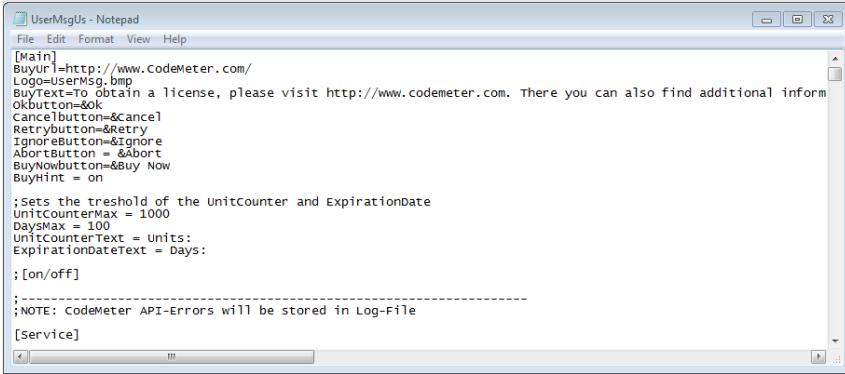


Figure 107: AxProtector - IxProtector only Windows "Error Messages"

Error Messages

| Element | Description |
|-------------------------------------|---|
| Suppress IxProtector Error Messages | The output of IxProtector error messages is suppressed (commandline option see here ²⁸¹). i If you do not activate this option, when using IxProtector errors, additional message windows are displayed along with the messages your 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 ²⁸²). |
| 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 ²⁸³). |

| Element | Description |
|---------------------------|---|
| | <p>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.</p>  <pre data-bbox="293 370 803 695"> [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 information about our software. CancelButton=&cancel RetryButton=&retry IgnoreButton=&ignore AbortButton = &Abort BuyNowButton=&Buy Now BuyHint = on ;Sets the threshold 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] </pre> |
| | <p>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.</p> |
| Inline Messages | <p>Links for .NET projects, with an inline assembly which can also be configured by *.ini files.</p> <p> This option is available for the encryption of .NET applications only.</p> |
| Customized Error Messages | <p>Activate this option to enter customized error messages displayed in the message boxes below.</p> |

7.5.1.3 Advanced Options

This input window lets you set further encryption options.

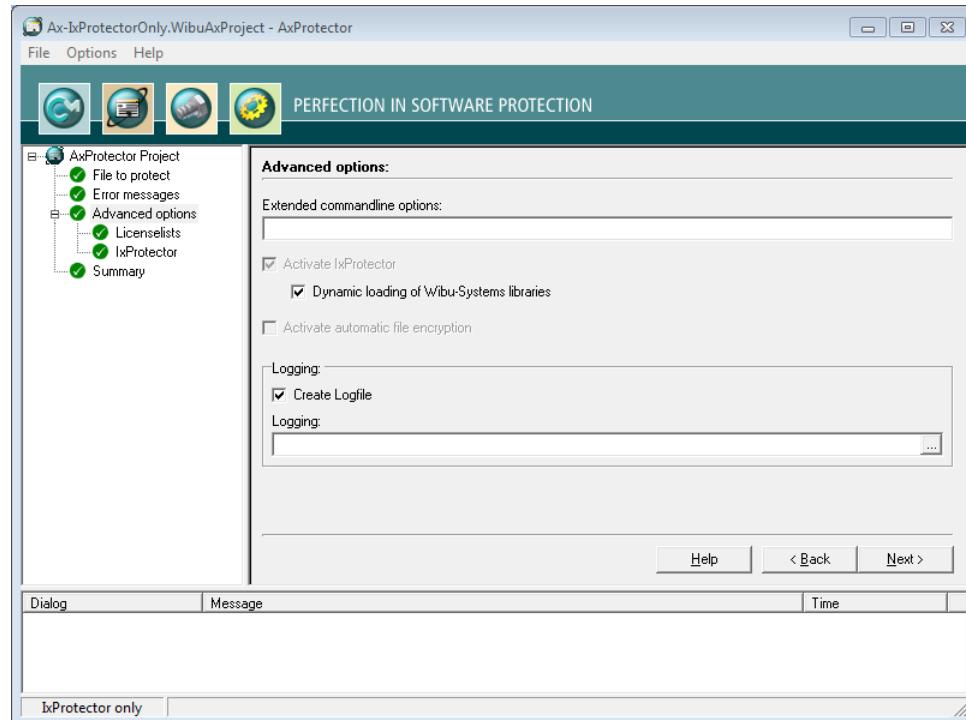


Figure 109: AxProtector - IxProtector only Windows "Advanced Options"

| Element | Description |
|---|---|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the AxProtector commandline. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i For more information please contact support at Wibu-Systems. </div> |
| 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 ²⁸¹) |
| 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. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

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 \(WUPI\)](#)²⁹⁶.

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

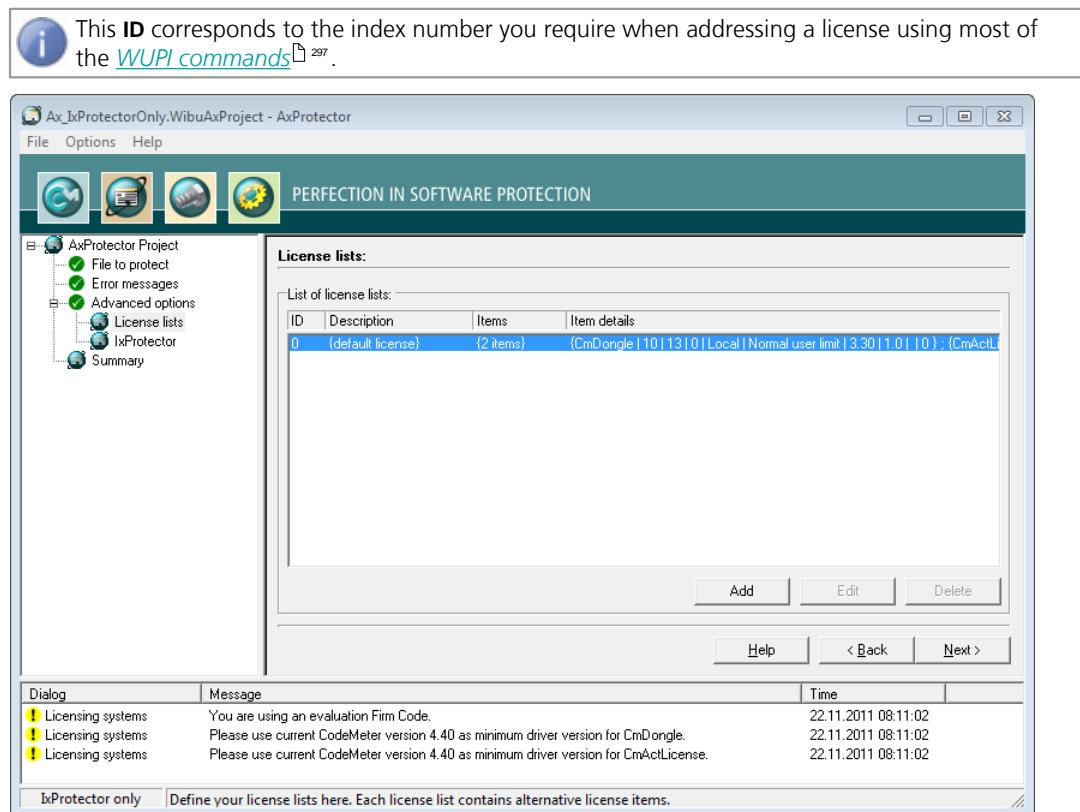
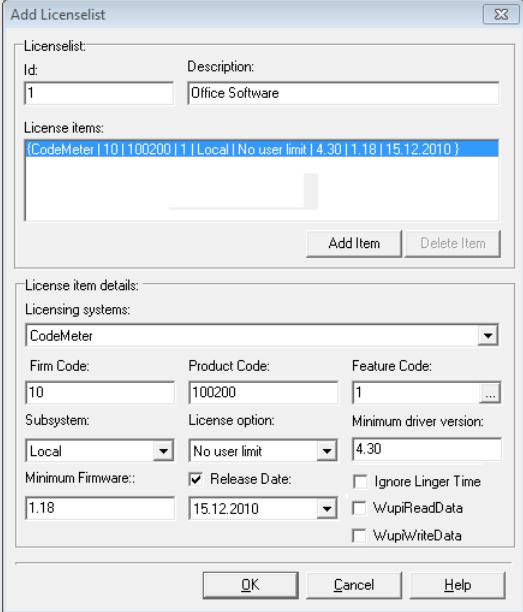
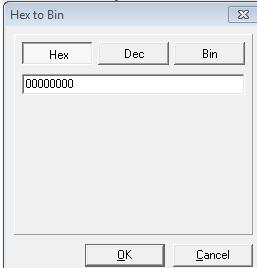


Figure 110: AxProtector - IxProtector only Windows "License Lists"

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 | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p>i 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.</p> |

| Element | Description |
|---|--|
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  <p>The dialog box has two main sections:</p> <ul style="list-style-type: none"> Licenseslist: Shows a table with one row: Id: 1 and Description: Office Software. License items: A list box containing the text: (CodeMeter 10 100200 1 Local No user limit 4.30 1.18 15.12.2010). <p>License item details: This section contains various dropdowns and input fields: <ul style="list-style-type: none"> Licensing systems: CodeMeter Firm Code: 10 Product Code: 100200 Feature Code: 1 Subsystem: Local License option: No user limit Minimum driver version: 4.30 Minimum Firmware: 1.18 Release Date: 15.12.2010 Checkboxes: Release Date (checked), Ignore Linger Time, WapiReadData, WapiWriteData </p> <p>Buttons at the bottom: OK, Cancel, Help.</p> |
| Figure 111: AxProtector - IxProtector only Windows "Add License Lists" | |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). |
| 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. Using the "... button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format. |
|  | |

| Element | Description |
|------------------------|--|
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>Starting with Firmware version 1.18 <i>CodeMeter</i>® 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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 1. Activate the "Release Date" checkbox to type in the Release Date. The current date is pre-set. 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>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 <i>CodeMeter</i> Developer Guide).</p> |
| WupiReadData | Activate this option to read data ³⁰⁰ from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

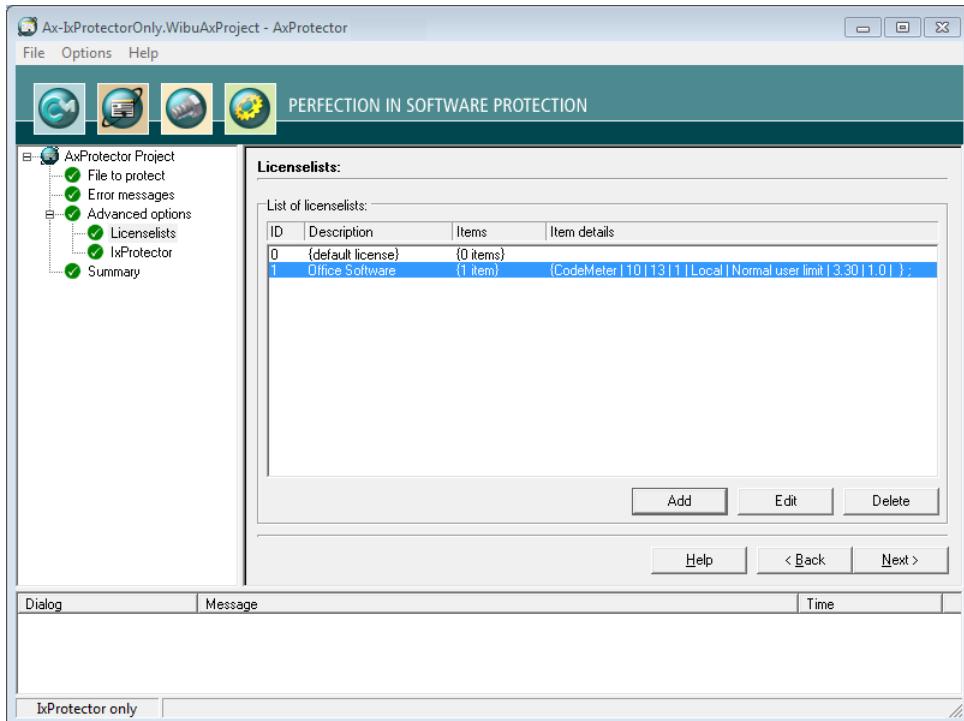


Figure 112: AxProtector - IxProtector only Windows "Completed License Lists"

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 (*.d11) 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.

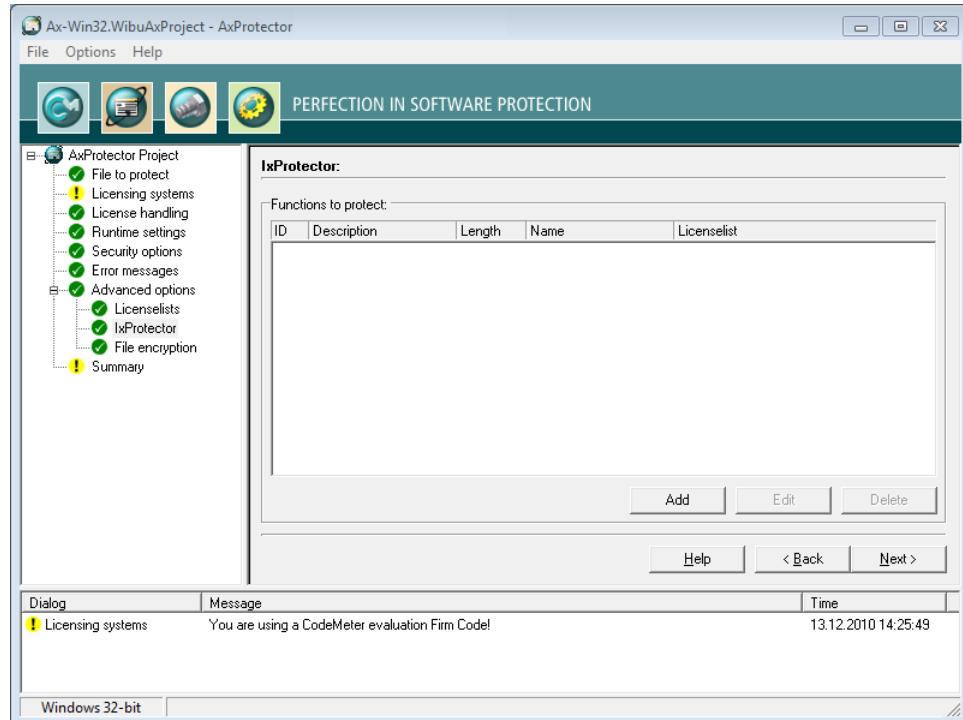
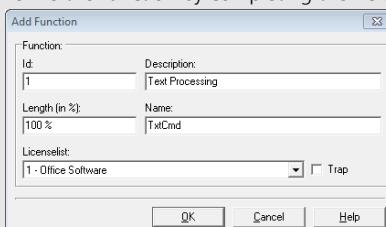


Figure 113: AxProtector - IxProtector only Windows "Function List"

| Element | Description |
|----------------------|--|
| Functions to protect | <p>Lists all specified function lists, including all properties. This menu item lets you also create function lists. Please proceed as follows:</p> <ol style="list-style-type: none"> 1. Click the "Add" button in the group "IxProtector Options". 2. Define the function by completing the fields in the "Function" group.  <p>The 'Add Function' dialog box contains fields for Function, Id (set to 1), Description (Text Processing), Length (in %) (set to 100 %), Name (TxtCmd), Licenselist (1 - Office Software), and a Trap checkbox. Buttons for OK, Cancel, and Help are at the bottom.</p> |
| | Figure 114: AxProtector - IxProtector only Windows "Add Function" |

| Element | Description | |
|---|-------------|--|
| | Element | Description |
| | | <p>i This Id corresponds to the identification you use when calling the WUPI commands WupiDecryptCode²⁹⁸ and WupiEncryptCode²⁹⁹.</p> |
| Description | | Enter a description of the function with text. |
| Length | | <p>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.</p> <p>i If you do not close the number by a percentage character, the specified number is interpreted as number of bytes.</p> |
| Name | | <p>Specify the name of the function to be encrypted.</p> <p>i 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.). Microsoft Dependency Walker shows dependencies between 32-or 64-bit Windows PE files. A tree view shows all linked modules and imported and exported functions are displayed in tables. Dependency Walker is part of Windows XP SP2 Support Tools and also part of Microsoft Visual Studio including version 8.0 (Visual Studio 2008, i.e. version 9.0 no longer provides Dependency Walker).</p> |
| 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. Command line option see here ²⁸⁵ . |
| <p>3. Click the "OK" button. The new functions are added to the function list.</p> | | |

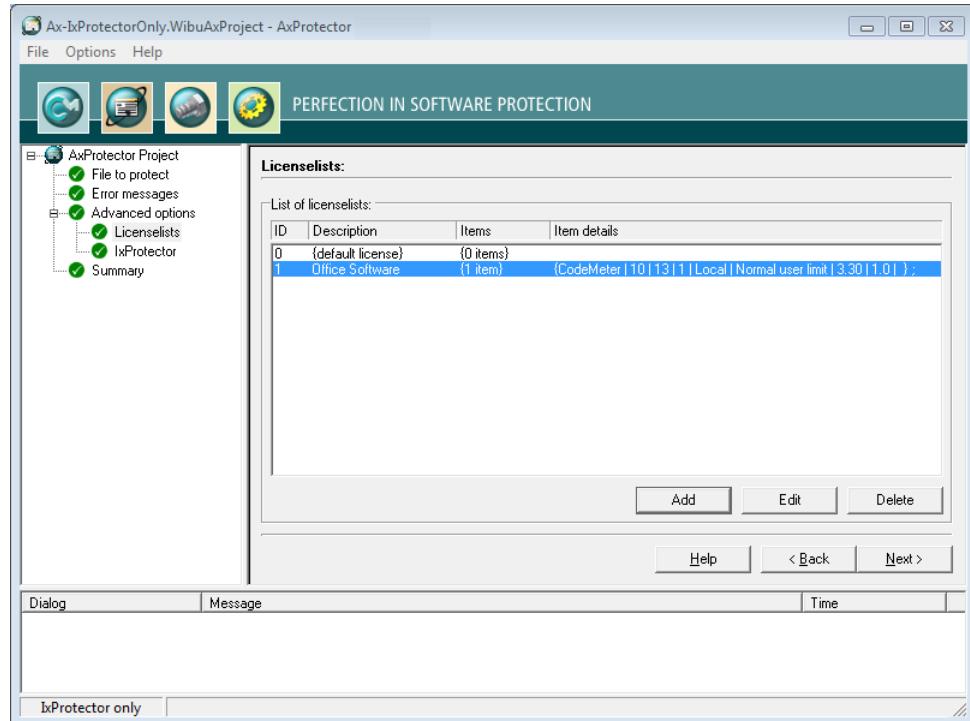


Figure 115: AxProtector - IxProtector only Windows "Completed Function List"

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](#)²⁹⁵ type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

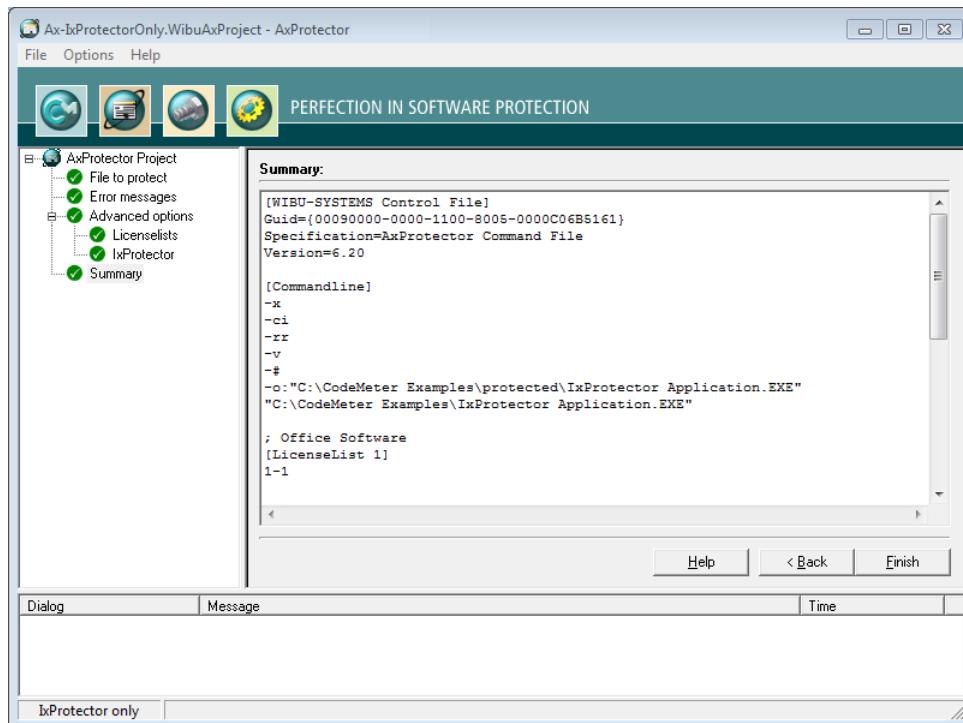


Figure 116: AxProtector - IxProtector only Windows "Summary"

| 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.

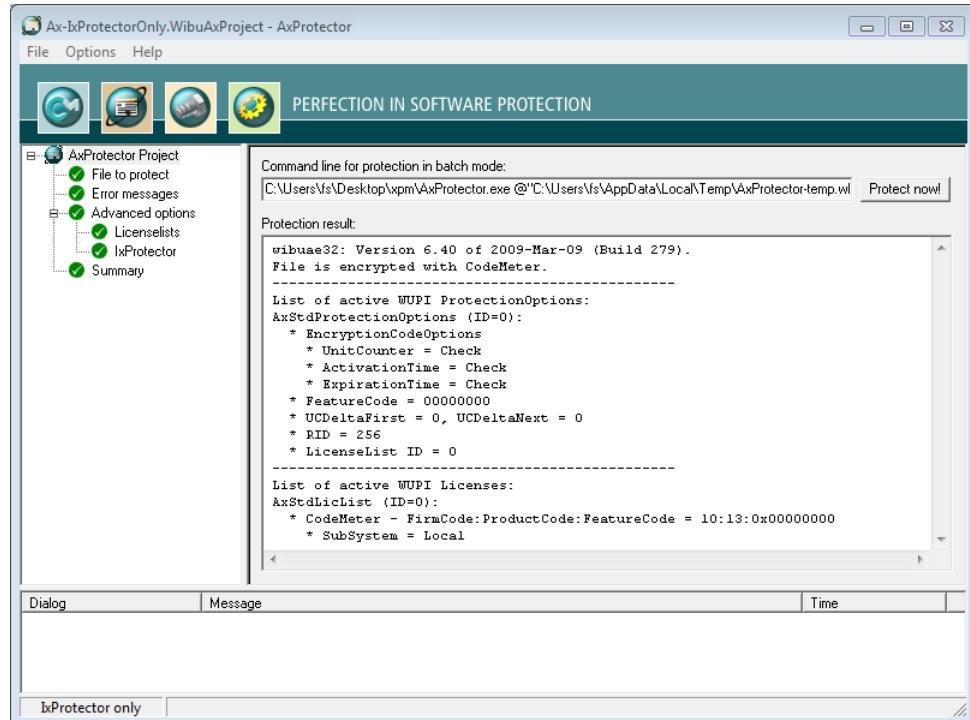


Figure 117: AxProtector - IxProtector only "Encryption Result"

| Element | Description |
|-------------|--|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the <i>AxProtector</i> commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i You are also able to copy the <i>AxProtector</i> 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. </div> |

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*.

i 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 pro-

tected 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 <small>213</small> |  | .NET commandline <small>270</small> |

7.5.2.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

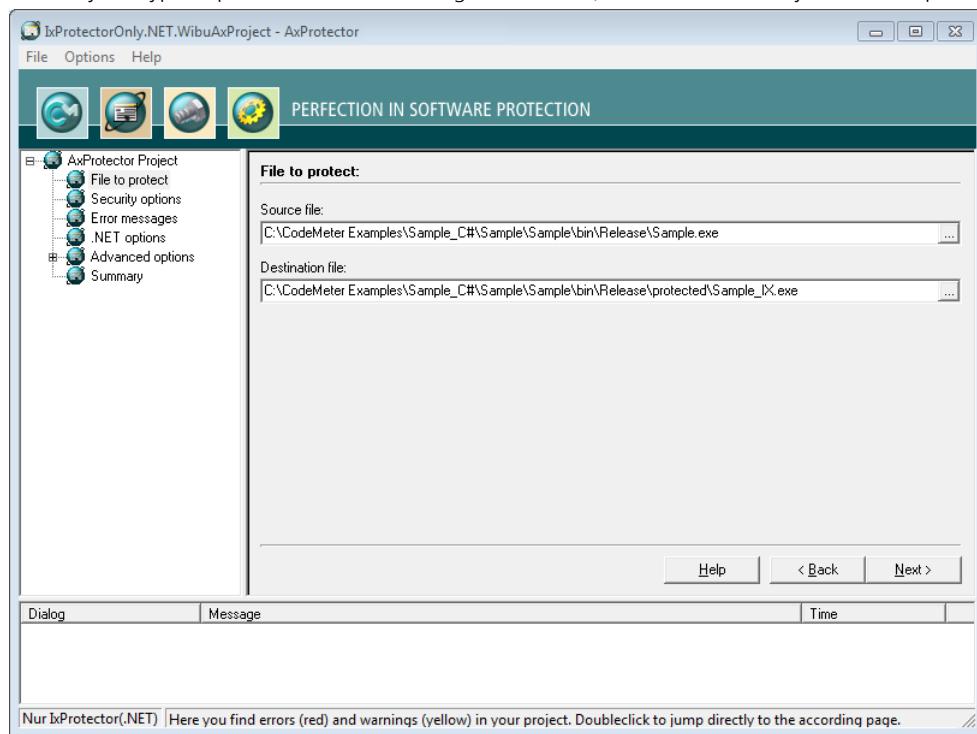


Figure 118: *AxProtector - IxProtector only (.NET)* "File to Protect"

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. |

| Element | Description |
|------------------|--|
| Destination File | After you selected the source file, <i>AxProtector</i> 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 . |

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.

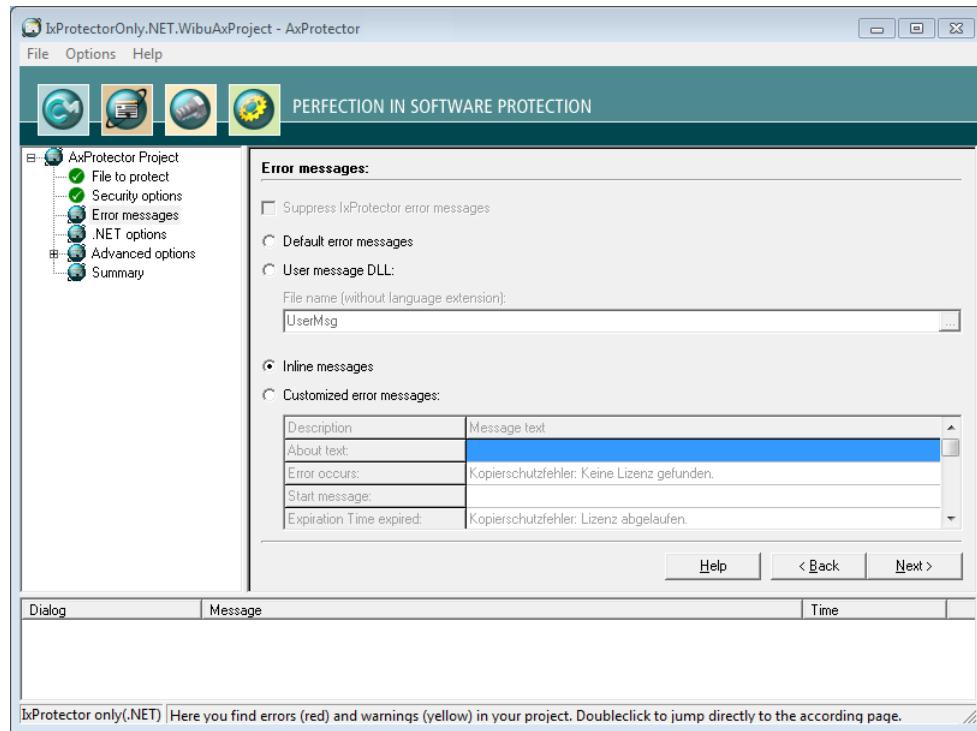
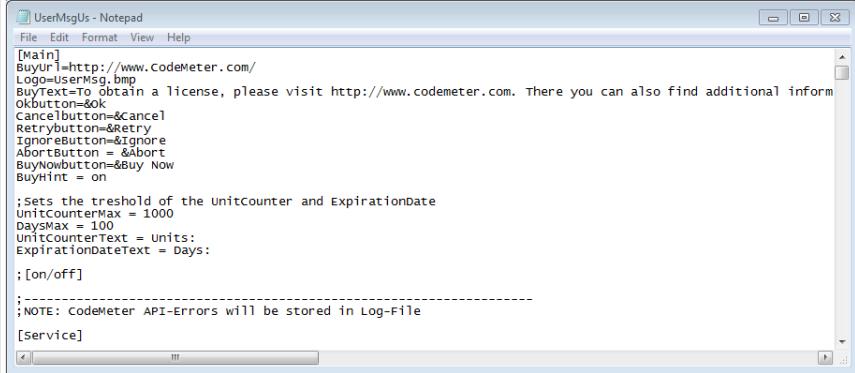


Figure 119: AxProtector - IxProtector only (.NET) "Error Messages"

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). |
| 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 |

| Element | Description |
|---------------------------|---|
| | <p>designs to this file, for example, by using separate logos or text (commandline option see here²⁸⁸).</p> <p> 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.</p>  <pre>[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 information about the product. OkButton=&OK CancelButton=&Cancel RetryButton=&Retry IgnoreButton=&Ignore AbortButton=&Abort BuyNowButton=&Buy Now BuyHint = on ;sets the threshold 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]</pre> |
| | <p>Figure 120: AxProtector – UserMsgUs.ini</p> <p>File name (without Language Extension)</p> <p>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.</p> |
| Inline Messages | <p>Links for .NET projects, with an inline assembly, can also be configured by *.ini files (commandline option see here²⁸⁹).</p> <p> 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.</p> |
| Customized Error Messages | <p>Activate this option to enter customized error messages displayed in the message boxes below.</p> |

7.5.2.3 .NET Options

This page allows you to specify further .NET settings.

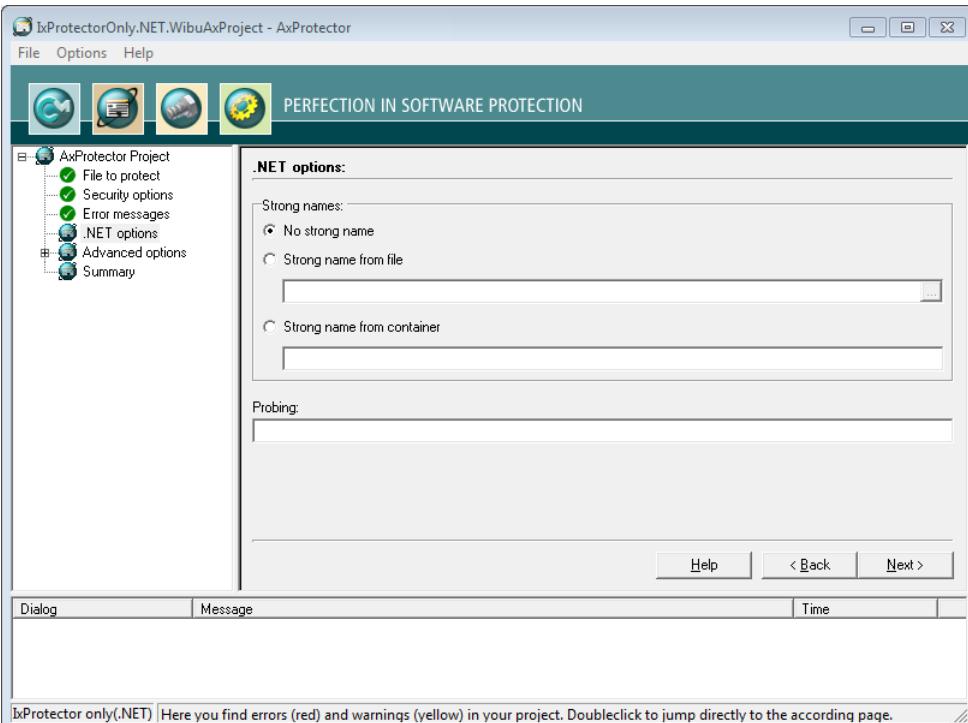


Figure 121: AxProtector - IxProtector only (.NET) ".NET Options"

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 ²⁸⁹). |
| Strong Name from Container | Activate this checkbox to use a container file to sign the program class (commandline option see here ²⁸⁹). |
| Probing | <p>This group allows you to specify the location of signed program classes in an <code>app.config</code> file.</p> <p> Specify the path to which the access to the program class refers separated by ";". Alternatively, specify the respective <code>app.config</code> file.</p> <p>Commandline option see here²⁸⁹.</p> |

7.5.2.4 Advanced Options

This input window lets you set further encryption options.

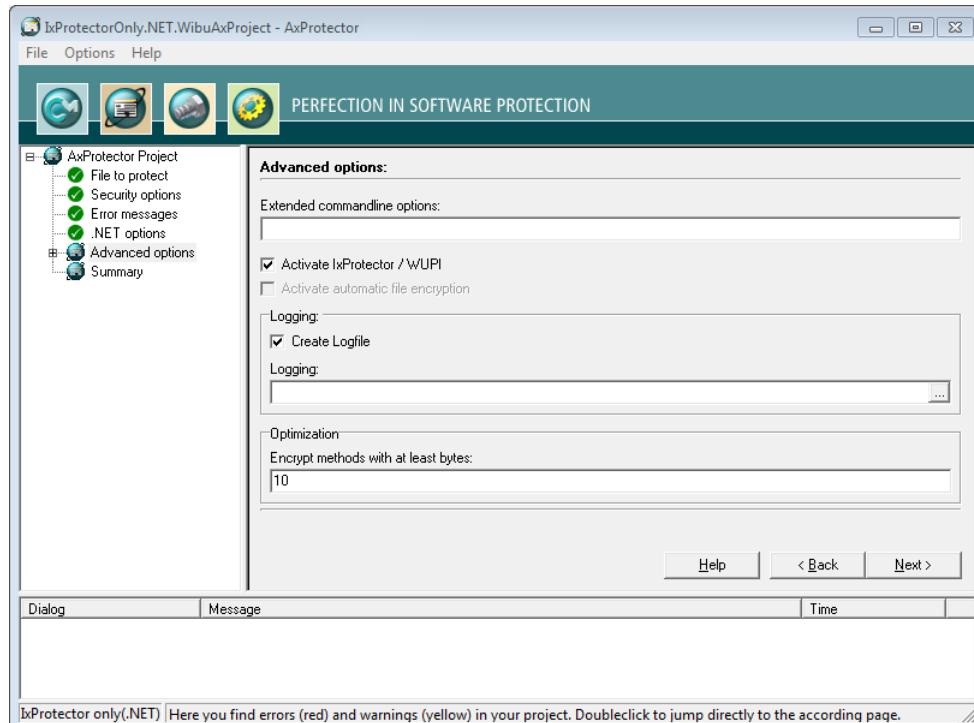


Figure 122: AxProtector - IxProtector only (.NET) "Advanced Options"

| Element | Description |
|------------------------------|--|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the <i>AxProtector</i> commandline. i 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 <i>IxProtector</i> via the Software Protection-API ²⁰⁶ . (commandline option here ²⁰⁷). |
| Create Logfile | Activate this checkbox to create file logging for the activities of <i>AxProtector</i> . |
| Logging | Specify the path and file name of this log file. i 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. |

| Element | Description |
|---------|--|
| | 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 see here ²⁸³ . |

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 \(WUPI\)](#)²⁹⁶.

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](#)²⁹⁷.

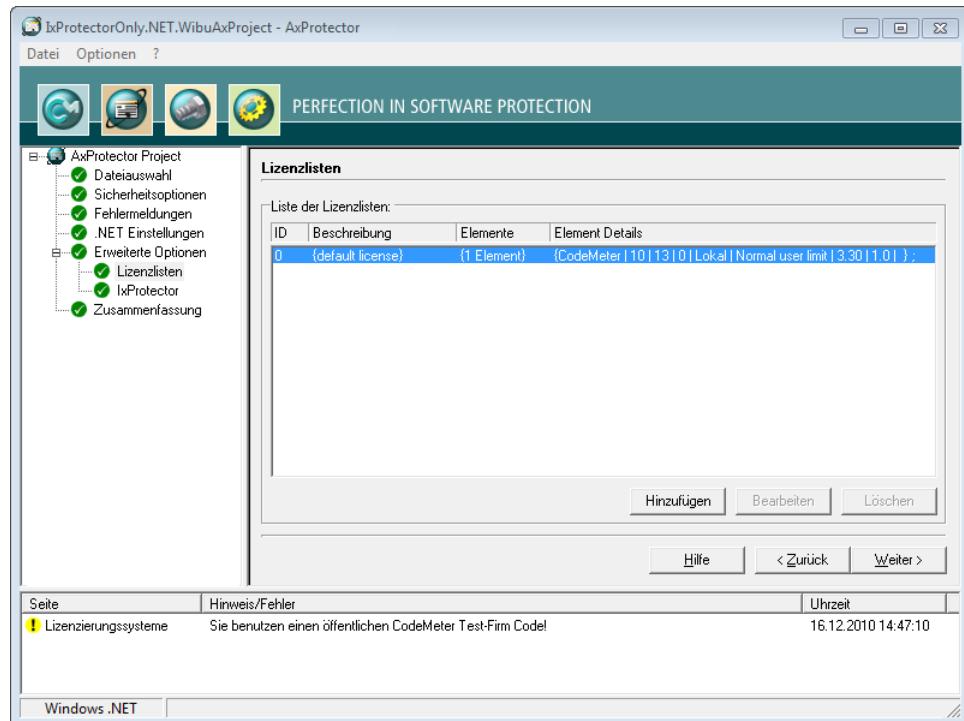


Figure 123: AxProtector - IxProtector only (.NET) "License Lists"

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**.

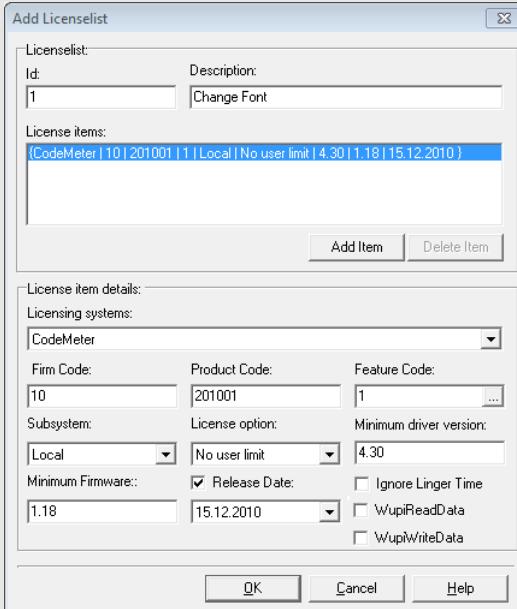
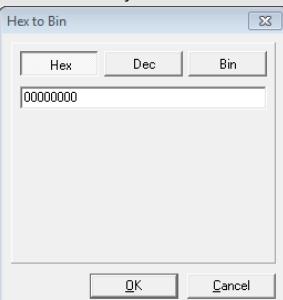
| Element | Description |
|-------------------|--|
| Id | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p> By default, an ID of 0 is initially set by the selection of the licensing system. Following, you are able to add license list entries starting from 1.</p> |
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> or <i>WibuKey</i>). |
| 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. |

Figure 124: AxProtector - IxProtector Only (.NET) - "Add License Lists"

| Element | Description |
|------------------------|---|
| | <p>Using the "... button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format.</p>  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>Starting with Firmware version 1.18 <i>CodeMeter</i>® 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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>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 <i>CodeMeter</i> Developer Guide).</p> |
| WupiReadData | Activate this option to read data from the <i>CmContainer</i> if this data has been previously stored at a defined location. |

| Element | Description |
|---------------|---|
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

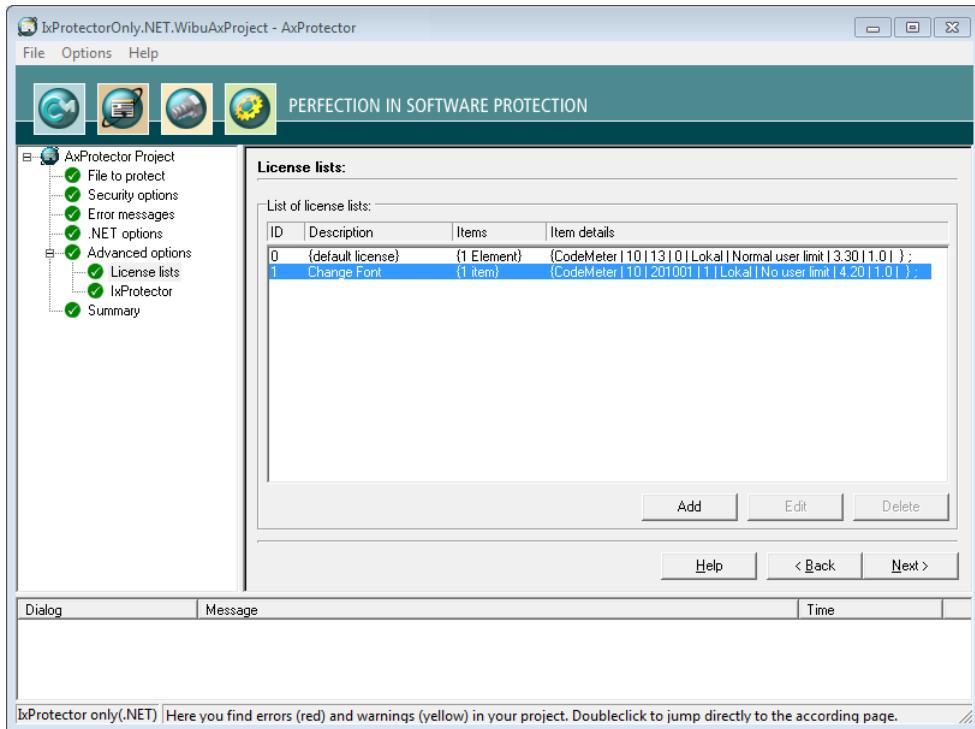


Figure 125: AxProtector - IxProtector only (.NET) - "Completed License List"

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.

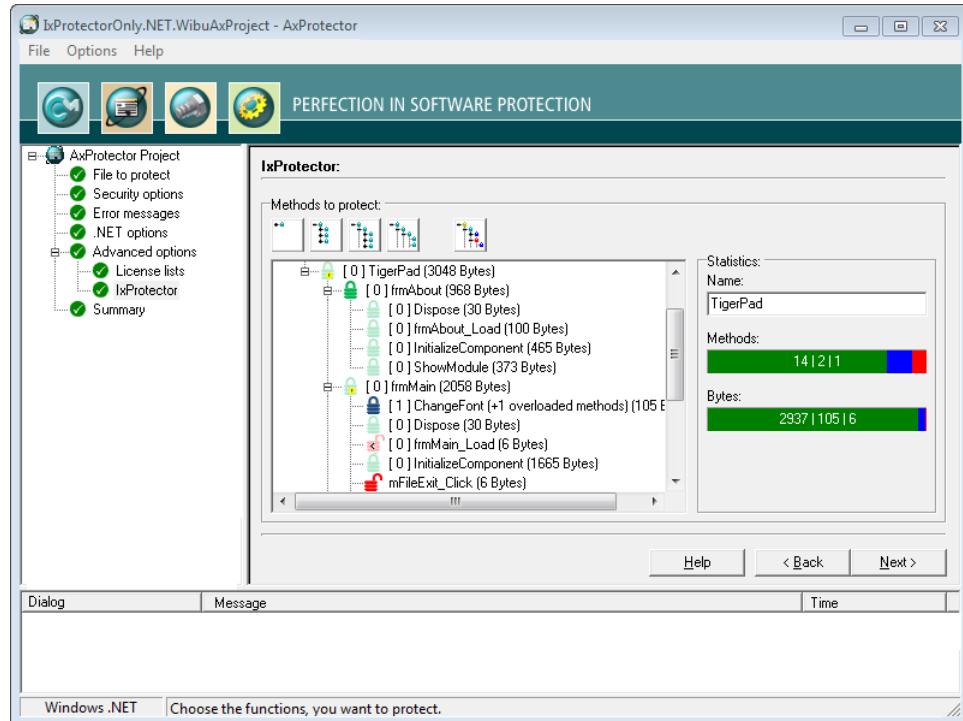


Figure 126: AxProtector - IxProtector only (.NET) "IxProtector"

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. |
| [+][*][*] | 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. |

| Element | Description | |
|---------|---|--|
| | Color | Description |
| Methods | Green | Shows that the method will be encrypted using <i>AxProtector</i> and that the License List ID has a value of 0 (default license) |
| | Blue | Shows that the method will be encrypted using <i>IxProtector</i> and that the License List ID has a value unequal 0. |
| | Red | Shows that the method is 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 <i>AxProtector</i> and that the License List ID has a value of 0 (default license) |
| | Blue | Shows that the method will be encrypted using <i>IxProtector</i> and that the License List ID has a value unequal 0. |
| | Red | Shows that the method is 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 <i>AxProtector</i> (License List ID with a value of 0, i.e. default license). |
| | Encrypts the selected element using <i>IxProtector</i> (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](#)²⁹⁵ type *AxProtector.exe @*.wbc*.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

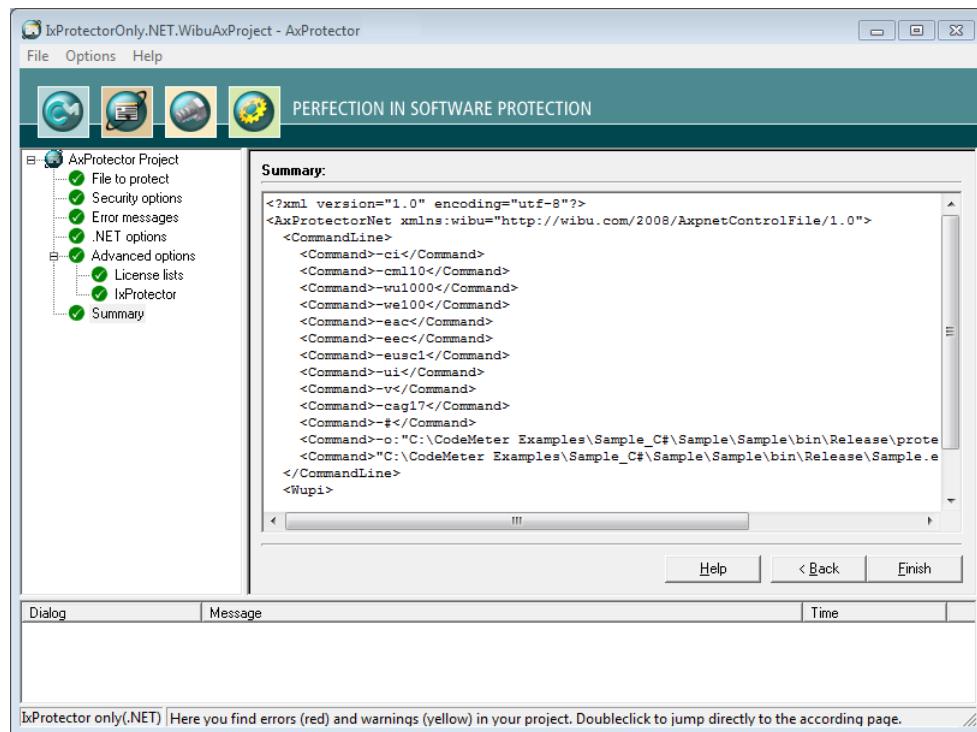


Figure 127: *AxProtector - IxProtector only (.NET)* "Summary"

| Element | Description |
|---------|--|
| Finish | Starts the encryption using <i>AxProtector</i> 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.

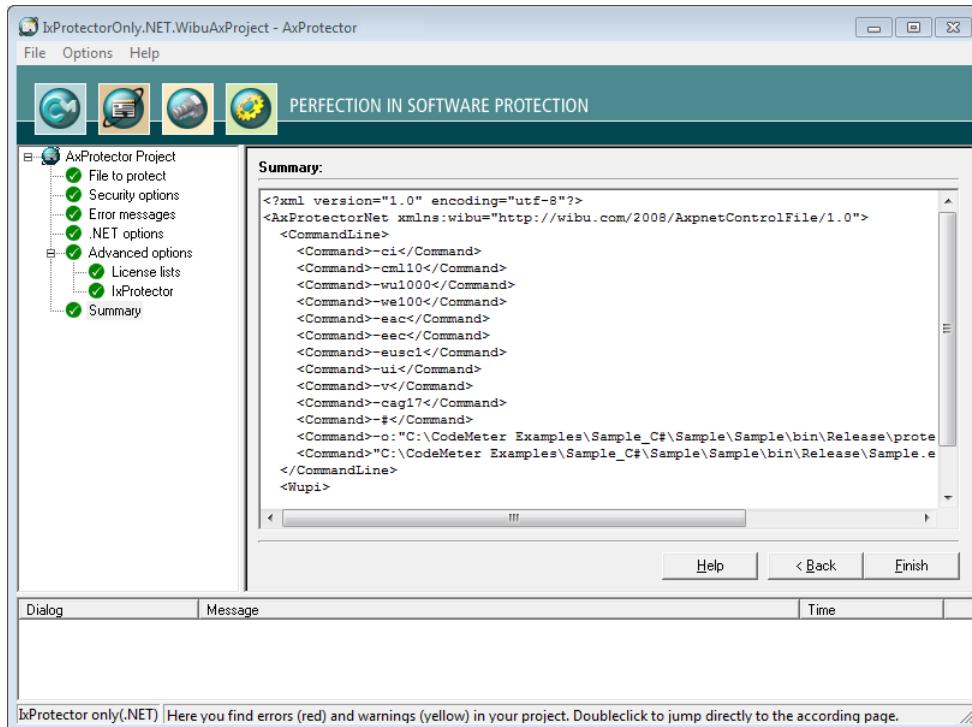


Figure 128: AxProtector - IxProtector only (.NET) "Encryption Result"

| Element | Description |
|-------------|---|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the <i>AxProtector</i> commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i You are also able to copy the <i>AxProtector</i> 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.</p> </div> |

7.5.3 Mac OS X 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*.

i 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 |
|-------------------------------|--|----------------|---|
| Mac OS X Application or Dylib |  IxProtector Mac ²²⁶ | ✓ | Windows commandline ²⁷⁰ In a separate commandline for Mac, running on Mac OS X operating systems, you are also able to insert encryption parameter ²²⁸ . |

7.5.3.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

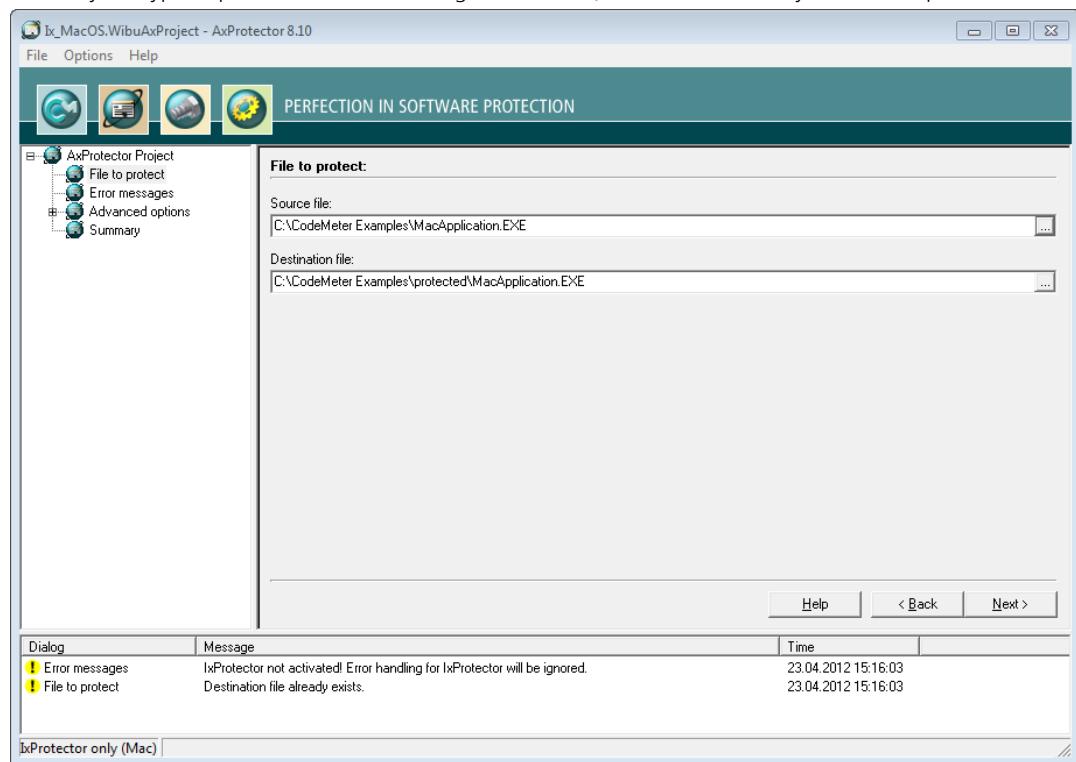


Figure 129: *IxProtector Mac "File to Protect"*

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. |
| 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 ²⁸⁹ . |

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.

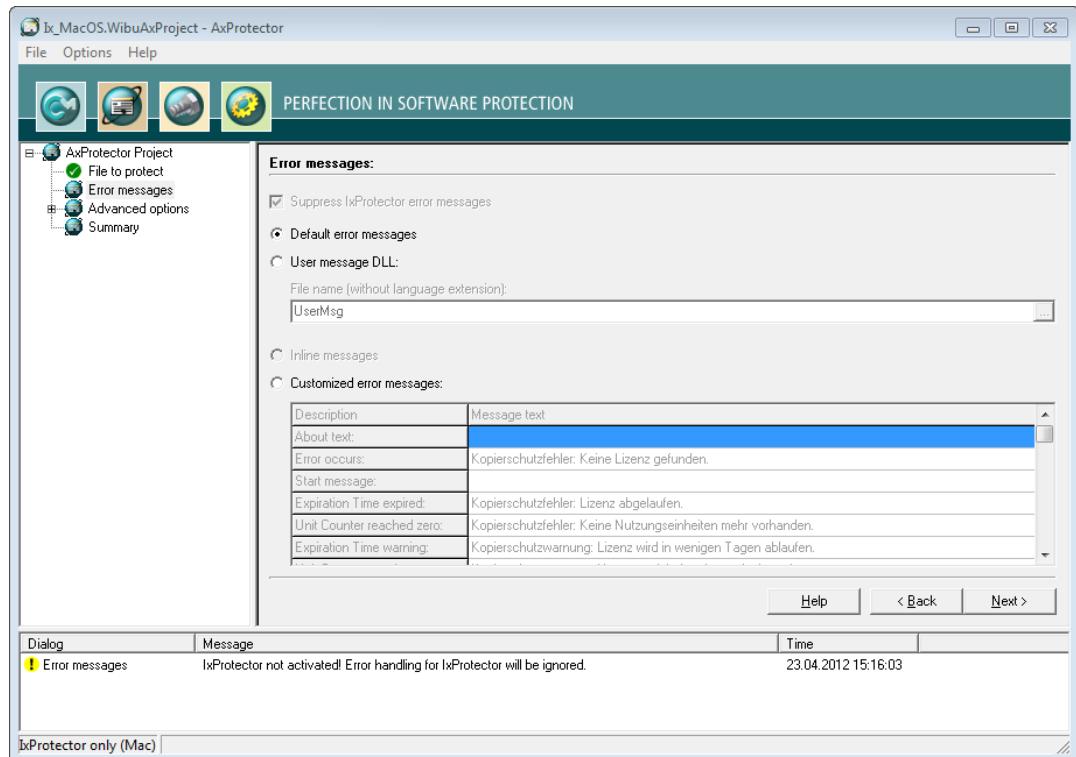
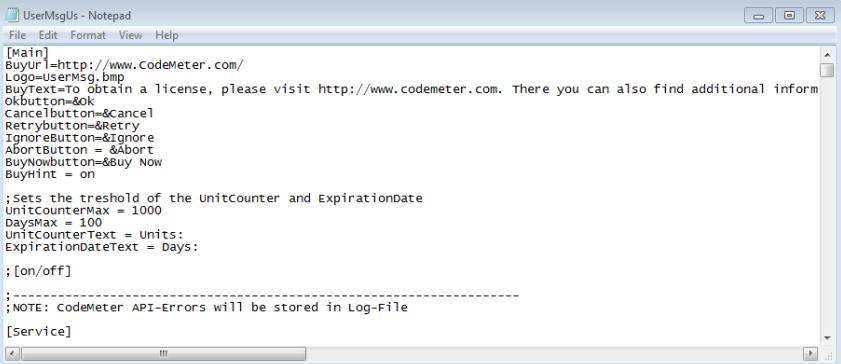


Figure 130: IxProtector Mac "Error Messages"

Error Messages

| Element | Description |
|--|---|
| Default Error Messages | All errors occurring at the runtime of a protected application display default error messages.. |
| User Message DLL | <p>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.</p> |
|  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. | |
|  <pre data-bbox="288 493 1129 857"> UserMsgUs - Notepad File Edit Format View Help [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 information. Okbutton=&Ok CancelButton=&Cancel RetryButton=&Retry Logo=AxProtector.bmp AbortButton = &Abort BuyNowButton=&Buy Now BuyHint = on ; Sets the threshold 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] </pre> | |
| Figure 131: 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. | |
| Customized Error Messages | Activate this option to enter customized error messages displayed in the message boxes below. |

7.5.3.3 Advanced Options

This input window lets you set further encryption options.

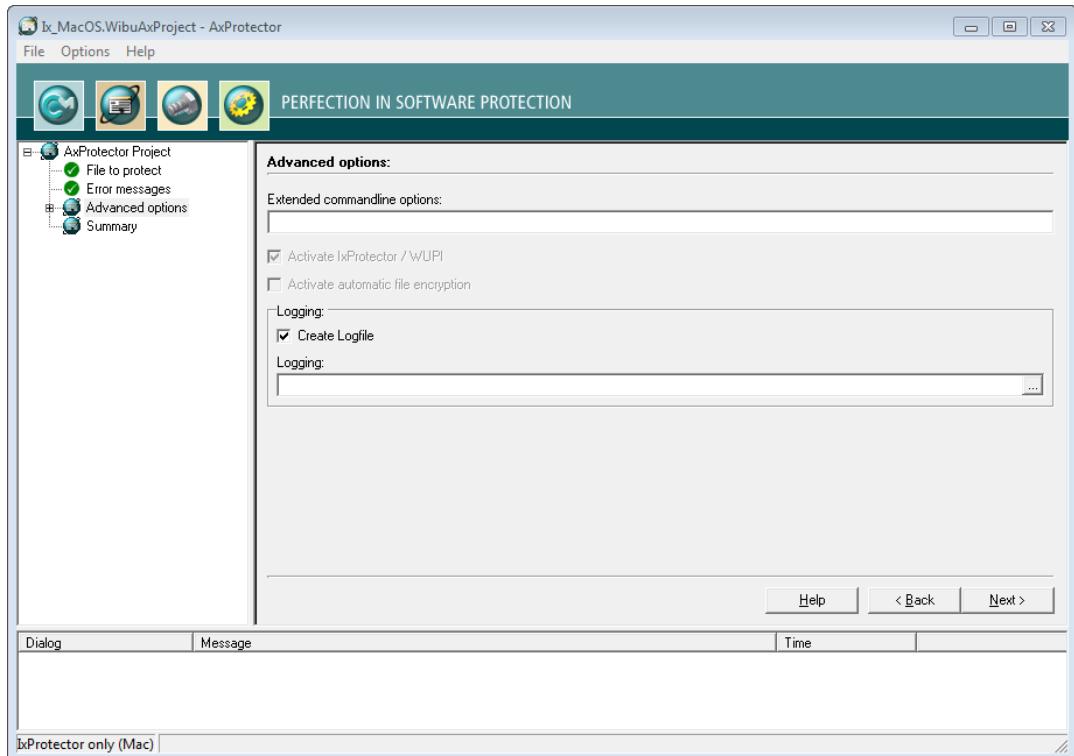


Figure 132: IxProtector Mac "Advanced Options"

| Element | Description |
|---|---|
| Extended Commandline Options | Here you are able to directly enter extended options or new feature functions using the AxProtector commandline. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i For more information please contact support at Wibu-Systems. </div> |
| 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. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i 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. </div> |

7.5.3.3.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *IxProtector* via the [Software Protection-API \(WUPI\)](#)²⁹⁶.

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

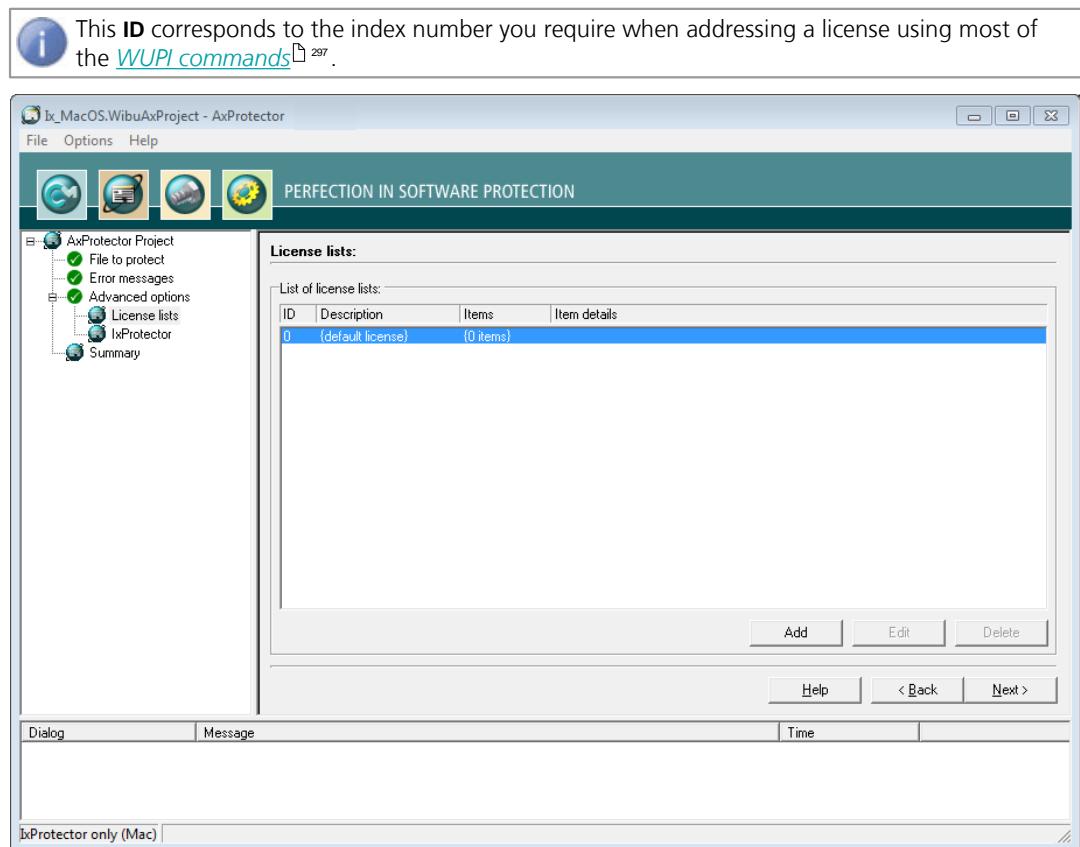
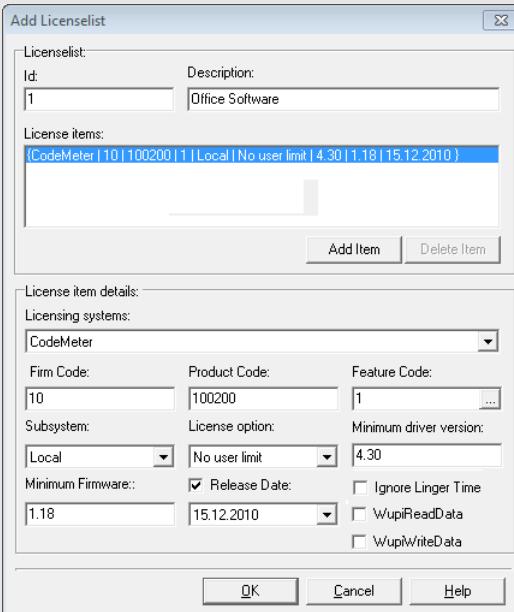


Figure 133: *IxProtector* Mac "License Lists"

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. |

| Element | Description |
|-------------------|---|
| |  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 . |
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  |
| | <p align="center">Figure 134: IxProtector Mac "Add License Lists"</p> |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). |
| 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 |
|------------------------|---|
| | <p>Using the "... button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format.</p>  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>Starting with Firmware version 1.18 <i>CodeMeter</i>® 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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>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 <i>CodeMeter</i> Developer Guide).</p> |
| WupiReadData | Activate this option to read data from the <i>CmContainer</i> if this data has been previously stored at a defined location. |

| Element | Description |
|---------------|---|
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

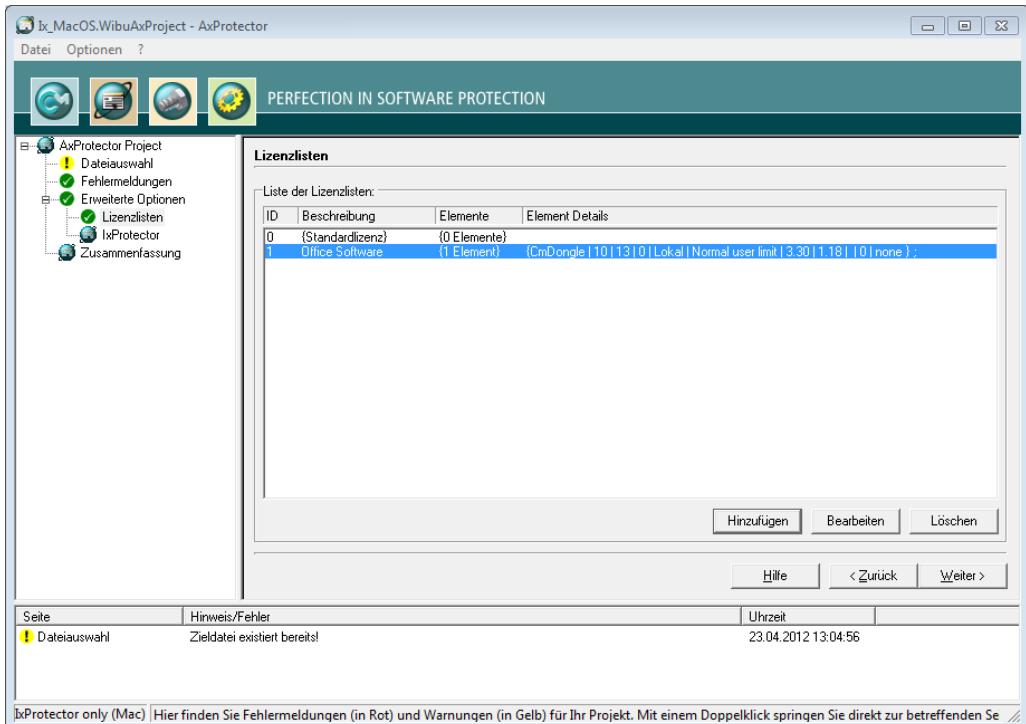


Figure 135: *IxProtector Mac "Completed License Lists"*

7.5.3.3.2 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.

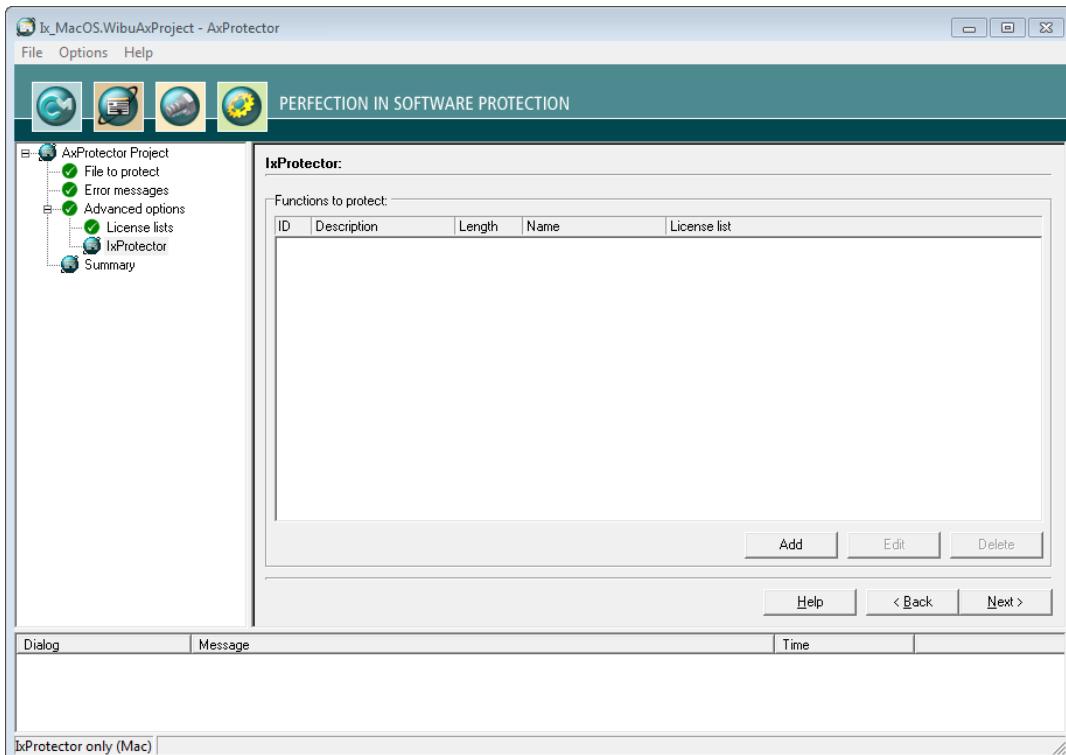
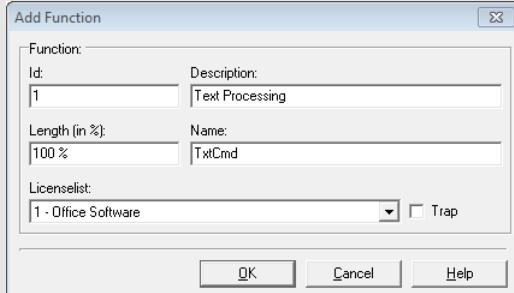


Figure 136: IxProtector Mac "Function List"

| 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: |

| Element | Description |
|---------|--|
| | <p>1. Click the "Add" button in the group "IxProtector Options".</p> <p>2. Define the function by completing the fields in the "Function" group.</p>  <p>Figure 137: IxProtector Mac "Add Function"</p> |

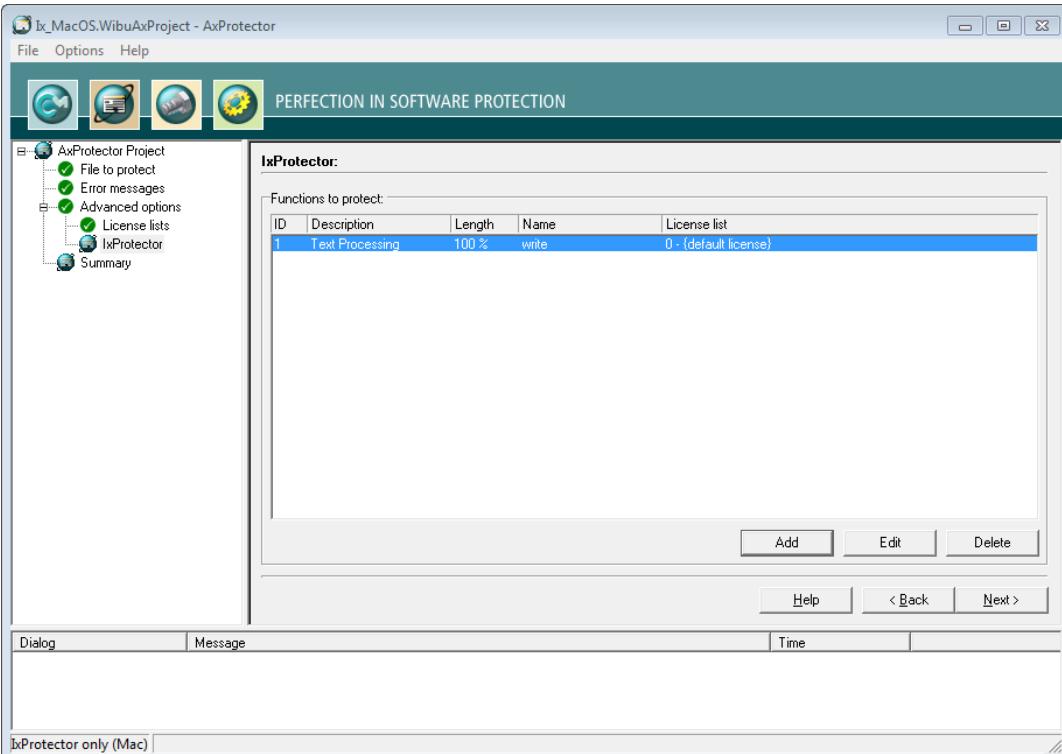


Figure 138: IxProtector Mac "Completed Function List"

7.5.3.4 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](#)²⁹⁵ type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

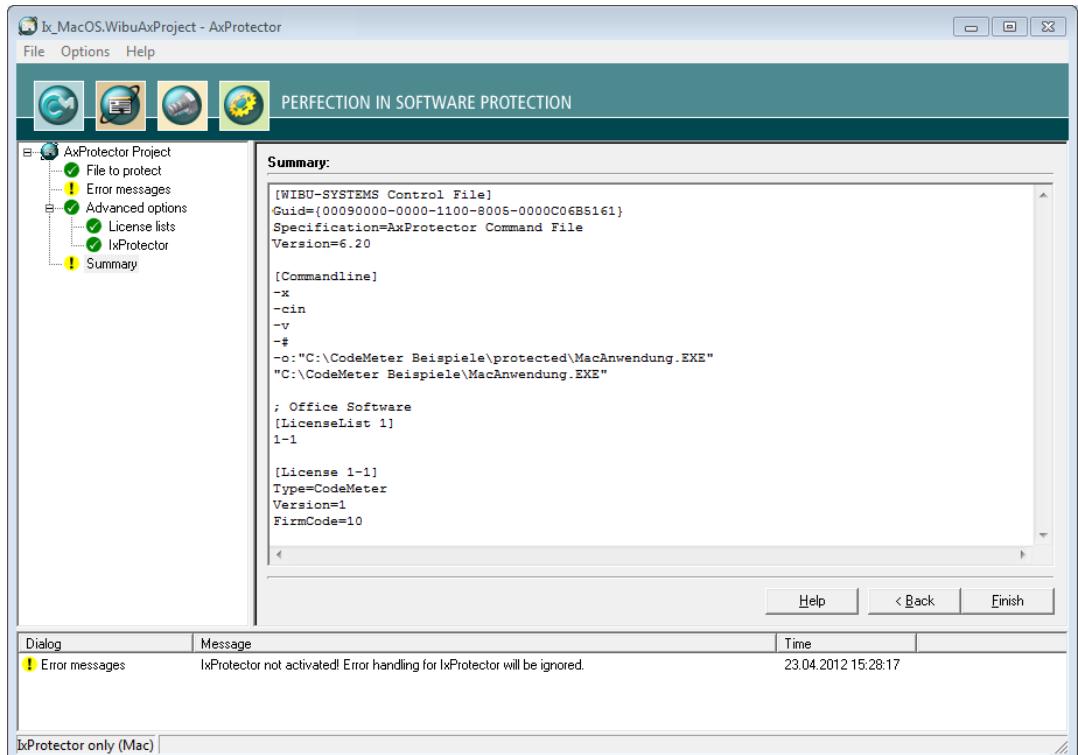


Figure 139: AxProtector - IxProtector only Mac "Summary"

| 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.

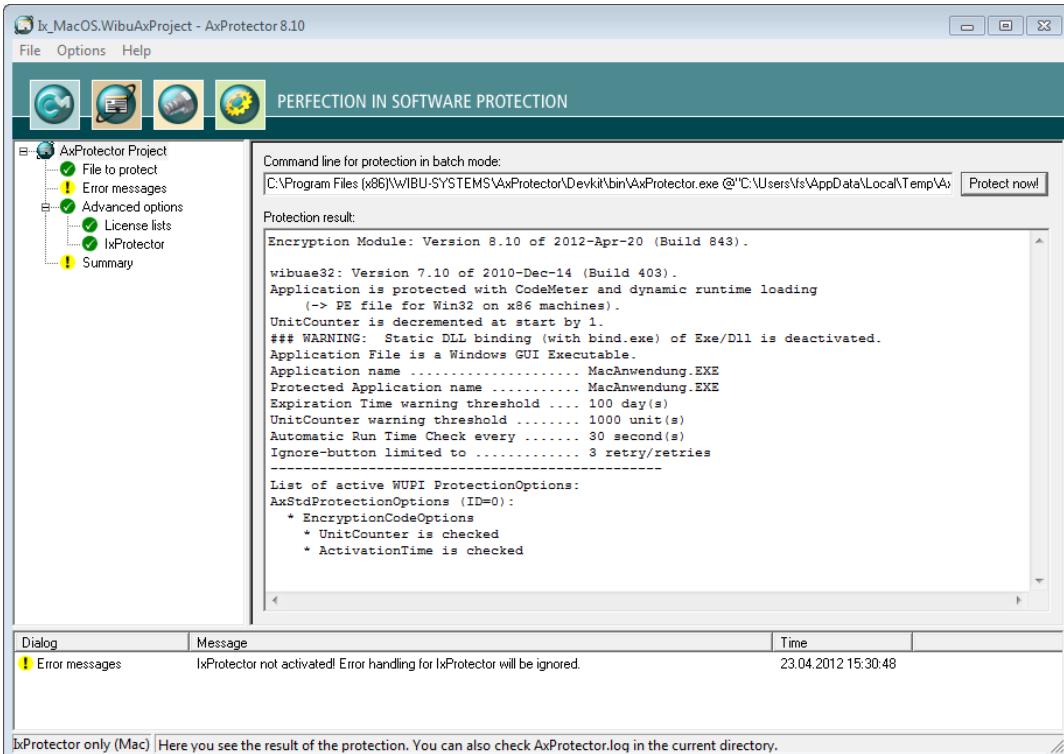


Figure 140: AxProtector - IxProtector only Mac "Encryption Result"

| Element | Description |
|-------------|---|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now!" button. Then the <i>AxProtector</i> commandline is executed in batch mode.</p> <p>Info: You are also able to copy the <i>AxProtector</i> 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.</p> |

7.5.4 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 pro-

ject 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 ²⁴⁰ | ✓ | Windows commandline ²⁷⁰ In a separate commandline for Linux, running on Linux operating systems, you are also able to insert encryption parameter ²⁵⁰ . |

7.5.4.1 File to protect

To safely encrypt respective code areas using *AxProtector*, first select the file you want to protect.

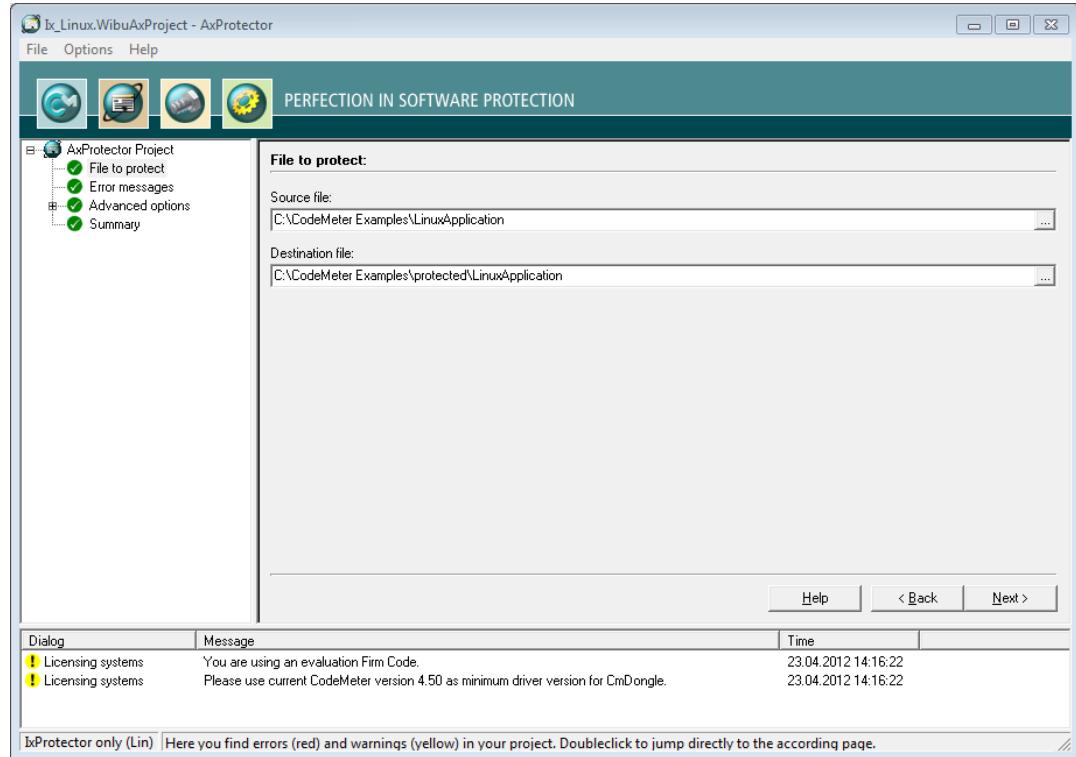


Figure 141: *IxProtector* Linux "File to Protect"

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, <i>AxProtector</i> 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 ²⁸⁹ . |

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.

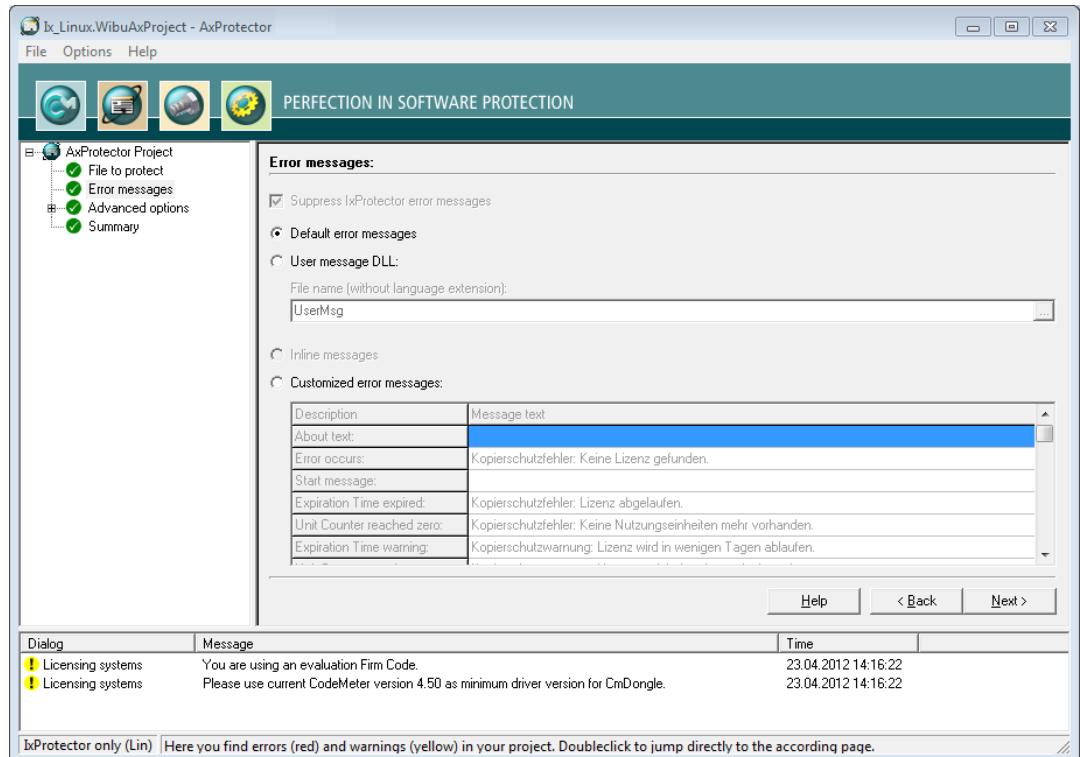
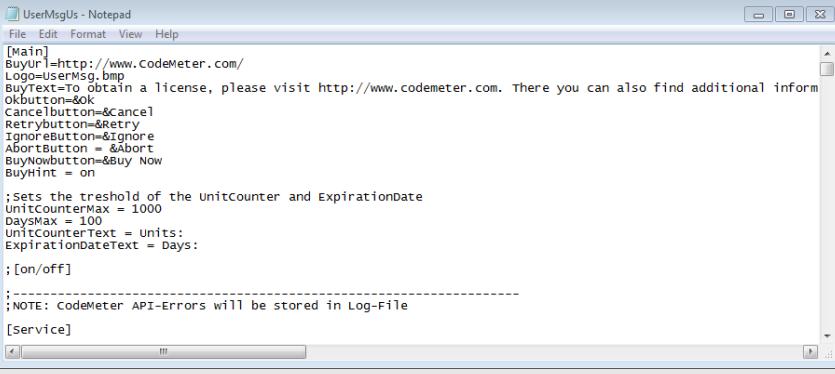


Figure 142: IxProtector Linux "Error Messages"

Error Messages

| Element | Description |
|-------------------------------------|---|
| Suppress IxProtector Error Messages | The output of IxProtector error messages is suppressed (commandline option see here ²⁸¹). i If you do not activate this option, when using IxProtector errors, additional message windows are displayed along with the messages your 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 ²⁸⁷). |
| 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 ²⁸⁸). |

| Element | Description |
|---------------------------|---|
| | <p> 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.</p>  <pre data-bbox="288 363 1123 737"> UserMsgUs - Notepad File Edit Format View Help [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 information about our software. CancelButton=&cancel Retrybutton=&retry Ignorebutton=&ignore Abortbutton = &Abort BuyNowbutton=&Buy Now BuyHint = on ;Sets the threshold 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] </pre> |
| | <p>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.</p> |
| Inline Messages | <p>Links for .NET projects, with an inline assembly which can also be configured by *.ini files (commandline option see here²⁸⁹).</p> |
| | <p> This option is available for the encryption of .NET applications only.</p> |
| Customized Error Messages | <p>Activate this option to enter customized error messages displayed in the message boxes below.</p> |

7.5.4.3 Advanced Options

This input window lets you set further encryption options.

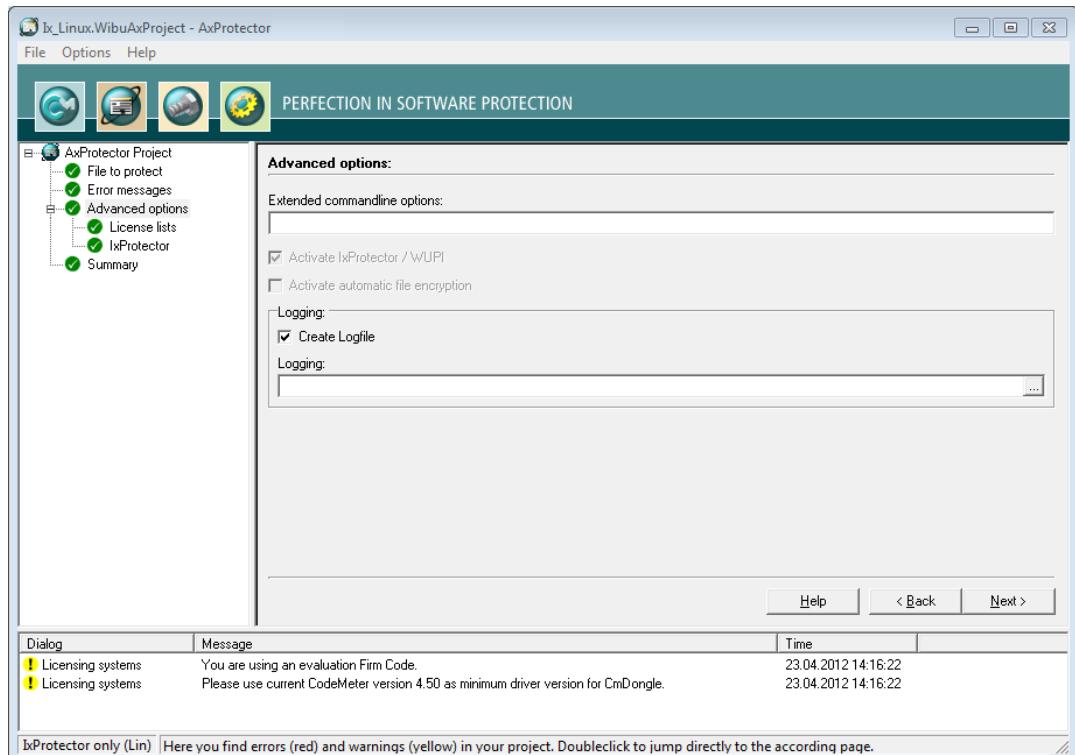


Figure 144:IxProtector Linux "Advanced Options"

| 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. |
| 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 \(WUPI\)](#)²⁹⁶.

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

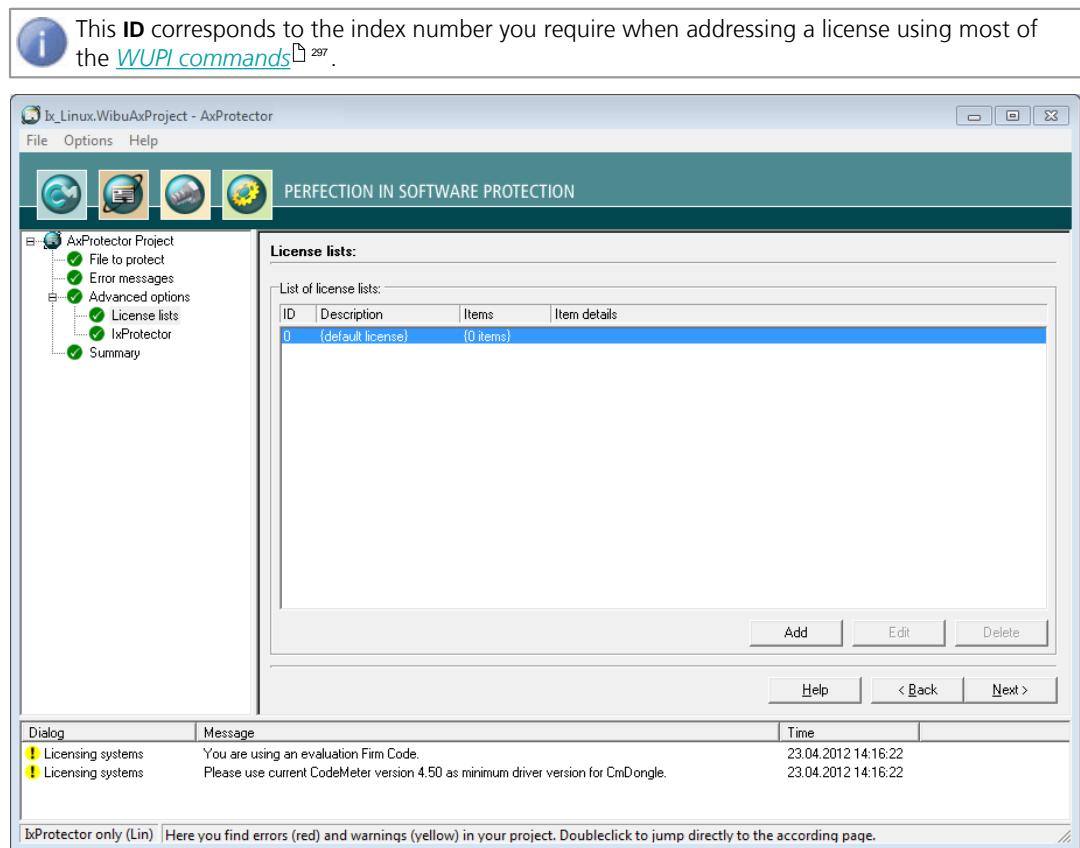
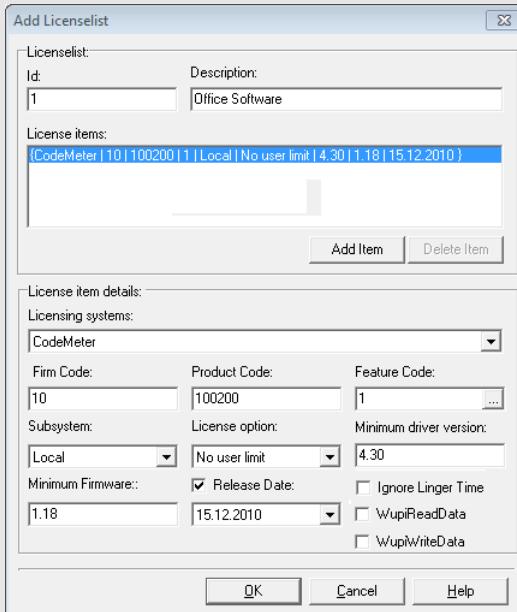


Figure 145: *IxProtector* Linux "License Lists"

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. |

| Element | Description |
|-------------------|--|
| | <p> 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.</p> |
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  <p>The screenshot shows the 'Add License List' dialog box. At the top, there's a section for 'Licenselist' with fields for 'Id' (set to 1) and 'Description' (set to 'Office Software'). Below this is a list titled 'License items' containing '(CodeMeter 10 100200 1 Local No user limit 4.30 1.18 15.12.2010)'. Underneath is a 'License item details' group with various configuration options:</p> <ul style="list-style-type: none"> Licensing systems: Set to 'CodeMeter'. Firm Code: '10'. Product Code: '100200'. Feature Code: '1'. Subsystem: 'Local'. License option: 'No user limit'. Minimum driver version: '4.30'. Minimum Firmware: '1.18'. Release Date: '15.12.2010'. Checkboxes: 'Release Date' (checked), 'Ignore Linger Time', 'WupReadData', and 'WupWriteData'. <p>At the bottom are 'OK', 'Cancel', and 'Help' buttons.</p> |
| | <p align="center">Figure 146: IxProtector Linux "Add License Lists"</p> |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> , or <i>WibuKey</i>). |
| 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 |
|------------------------|---|
| | <p>Using the "... button opens a Hex to Bin window where you can input in hexadecimal, decimal or binary format.</p>  |
| Subsystem | <p>Select the subsystem in which the protected application is to search (local or network), and define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • 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 | <p>Starting with Firmware version 1.18 <i>CodeMeter</i>® 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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 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 | <p>Activate this option to ignore a programmed LingerTime.</p> <p>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 <i>CodeMeter</i> Developer Guide).</p> |
| WupiReadData | Activate this option to read data from the <i>CmContainer</i> if this data has been previously stored at a defined location. |

| Element | Description |
|---------------|---|
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

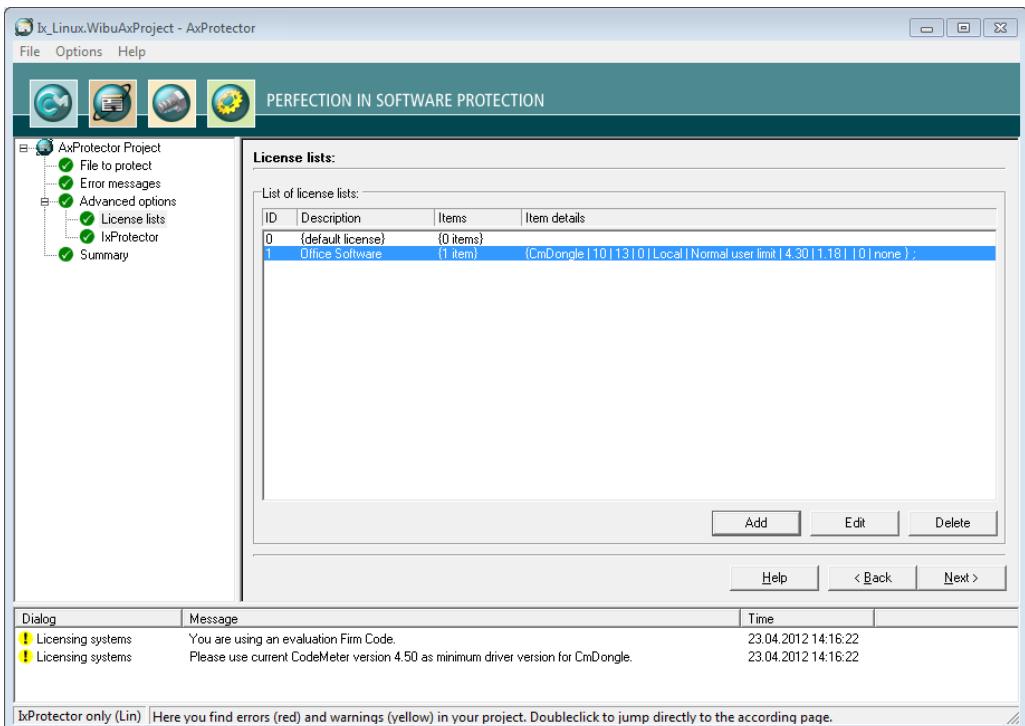


Figure 147: *IxProtector* Linux "Completed License Lists"

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.

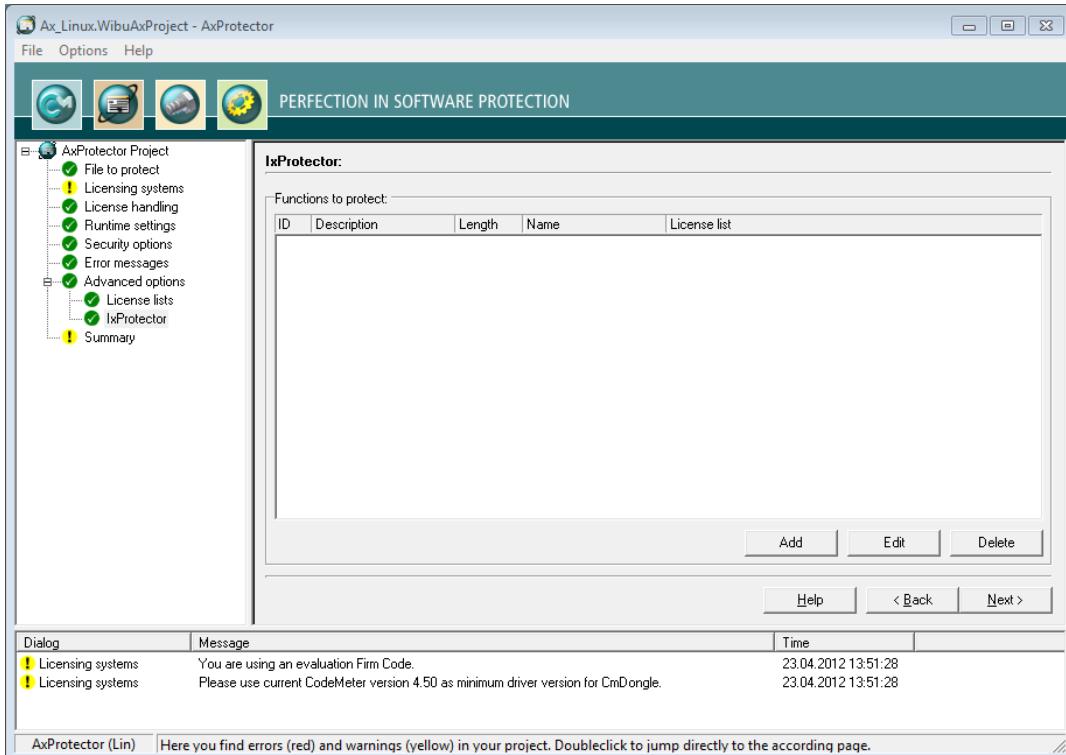
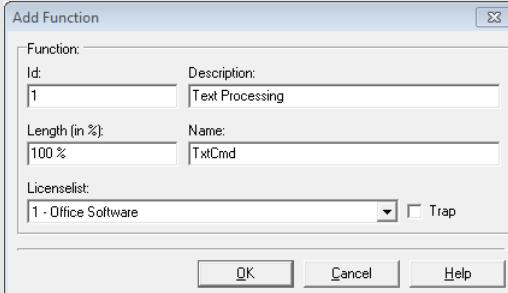


Figure 148: IxProtector Linux "Function List"

| Element | Description |
|----------------------|--|
| Functions to protect | <p>Lists all specified function lists, including all properties.</p> <p>This menu item lets you also create function lists. Please proceed as follows:</p> |

| Element | Description |
|---------|---|
| | <p>1. Click the "Add" button in the group "IxProtector Options".</p> <p>2. Define the function by completing the fields in the "Function" group.</p>  <p>Figure 149: AxProtector - IxProtector only "Add Function"</p> |

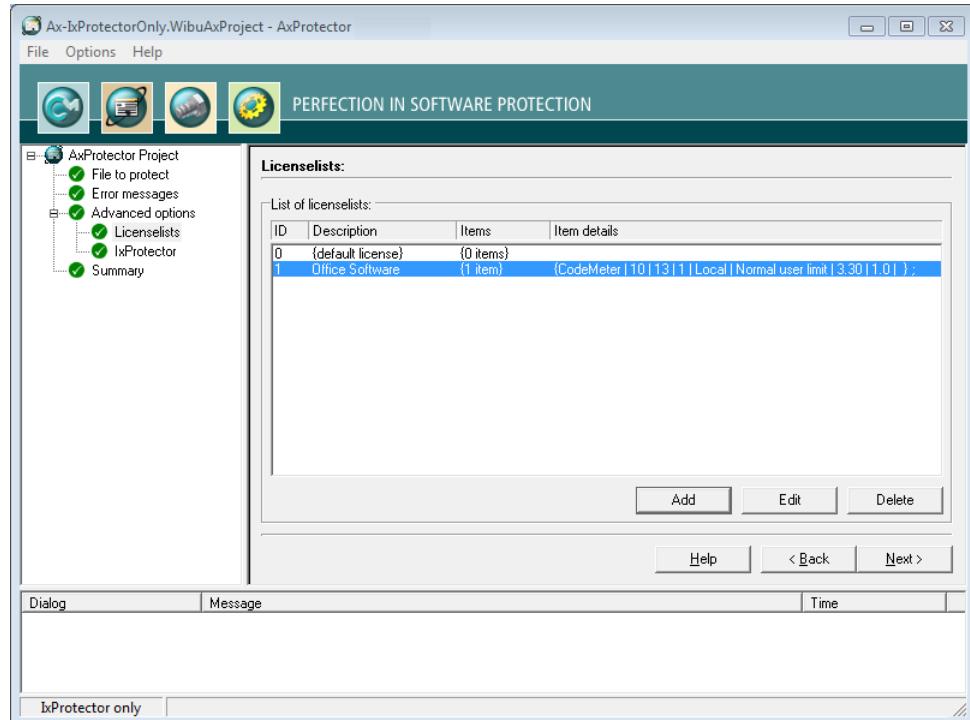


Figure 150: AxProtector - IxProtector only "Completed Function List"

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](#)²⁹⁵ type AxProtector.exe @*.wbc.

Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

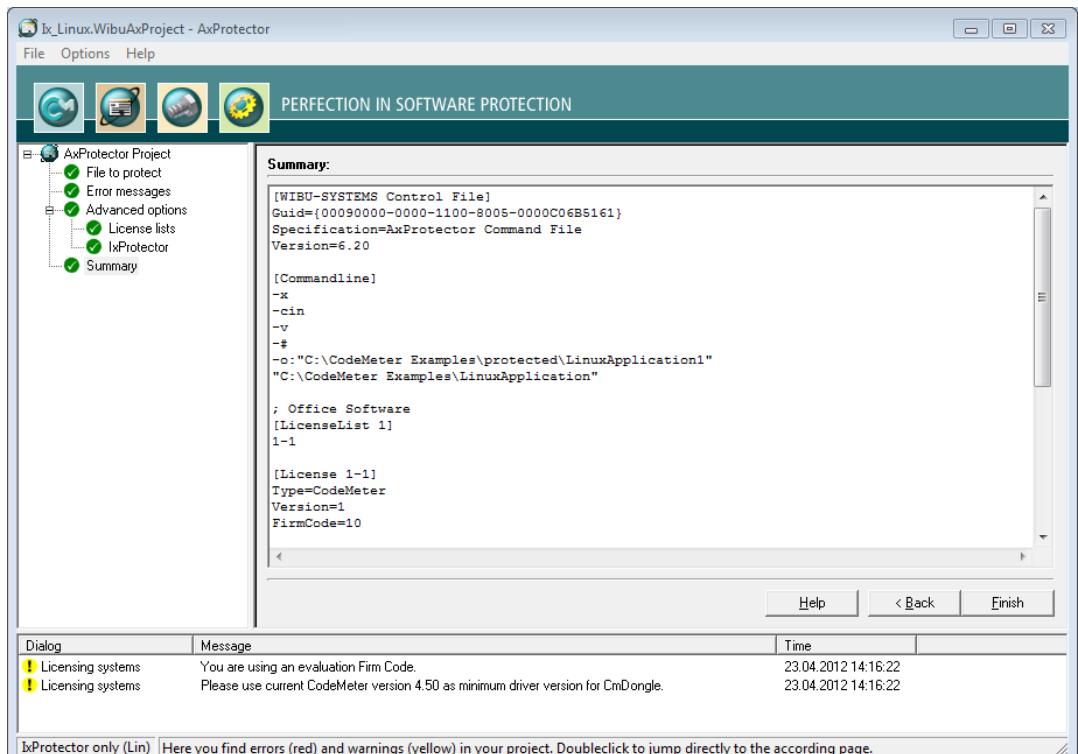


Figure 151: AxProtector - IxProtector only Linux "Summary"

| 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.

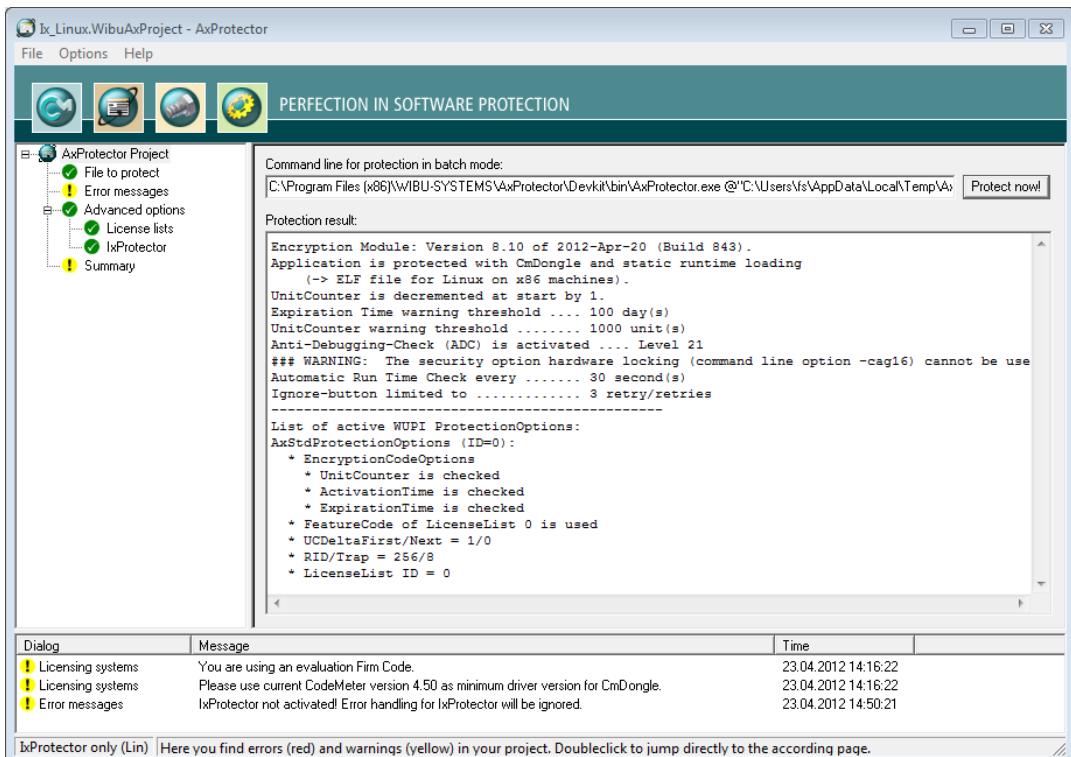


Figure 152: AxProtector - IxProtector only Linux "Encryption Result"

| Element | Description |
|-------------|--|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the AxProtector commandline is executed in batch mode.</p> <p>i 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.</p> |

7.6 Other Tab

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.

7.6.1.1 File to protect

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

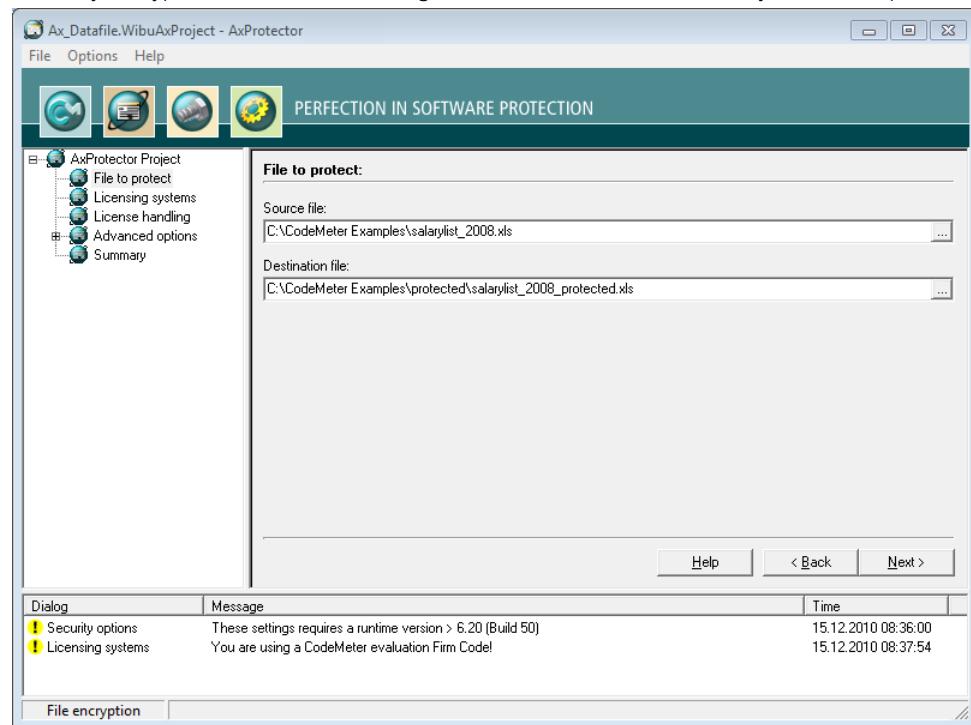


Figure 153: *AxProtector* - File Encryption "File to Protect"

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  . |

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 can select which protection schemes will be used. Depending on your requirements, you can select one or all of the check boxes (*CmDongle* and/or *CmActLicense*, *WibuKey*).

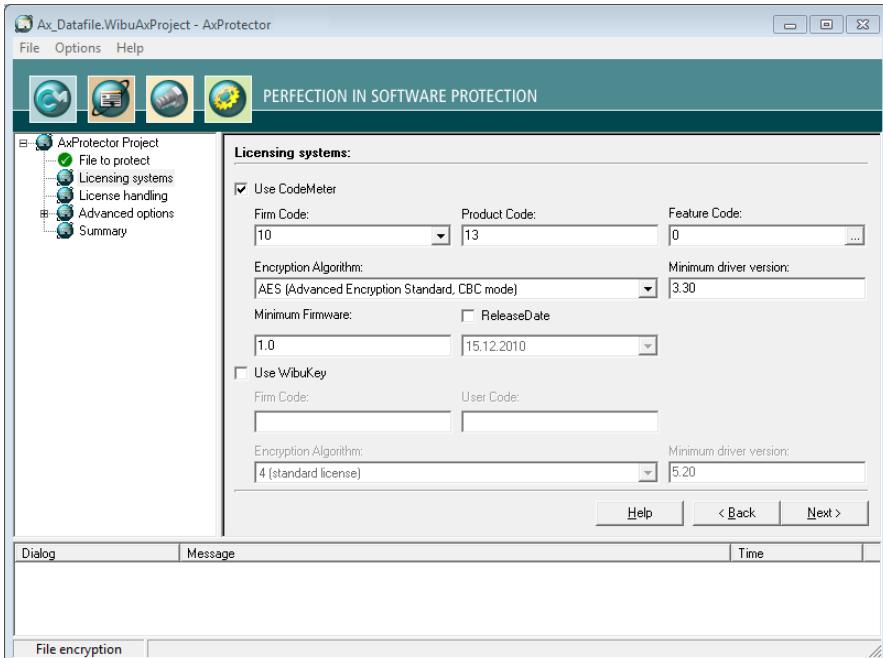
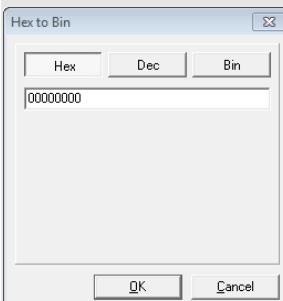


Figure 154: AxProtector - File Encryption "Licensing Systems"

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.

For *CmDongle* and *CmActLicense* the following settings are available:

| Element | Description |
|--------------|---|
| Firm Code | <p>Specify the Firm Code to be used for encrypting the software.</p> <p>i The Firm Code 10 used in figure above is the <i>CmDongle</i> evaluation <i>Firm Code</i> found in the <i>CodeMeter® Software Development Kit (SDK)</i>. In real life you would not use a Firm Code of 10, since this would be insecure. As a registered licensor, you will be issued your own unique Firm Code. The test Firm Code for <i>CmActLicense</i> is 5010.</p> <p>As a registered licensor, you will be issued your own unique Firm Code(s).</p> <p>Commandline option see here  ²⁷¹.</p> |
| Product Code | <p>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.</p> <p>Commandline option see here  ²⁷¹.</p> |

| Element | Description |
|------------------------|--|
| Feature Code | <p>Enter the Feature Code which defines, for example, the encryption of different software versions.</p> <p> 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.</p> <p>Using the "..." button you may enter the feature map value in hexadecimal, decimal or binary format.</p>  |
| Encryption algorithm | Select the algorithm to encrypt your software. Currently, <i>CodeMeter®</i> solely supports AES (Advanced Encryption Standard). Commandline option see here ²⁷² . |
| Minimum Driver Version | Enter the minimum driver version required for the installed <i>CodeMeter License Server</i> . When setting the minimum driver version to 3.20 the session handling for terminal servers is automated. This means that <i>AxProtector</i> automatically handles sessions of the protected software, and each session is allocated one of the available licenses. |
| Release Date | 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 ²⁷² . |
| Minimum Firmware | Starting with Firmware version 1.18 <i>CodeMeter®</i> supports the Product Item Option Maintenance Period ⁴⁵ . Specify the minimum firmware version required. In order to use the Product Item Option Maintenance Period you require the firmware version 1.18. Commandline option see here ²⁷² . |

WibuKey

For setting *WibuKey* options, see the separate "WibuKey Developer Guide".

7.6.1.3 License Handling

This input window lets you to define whether the protected application is to search for existing licenses locally in the *CmContainer*, on the network or both. Moreover, you can define the license allocation (access) mode.

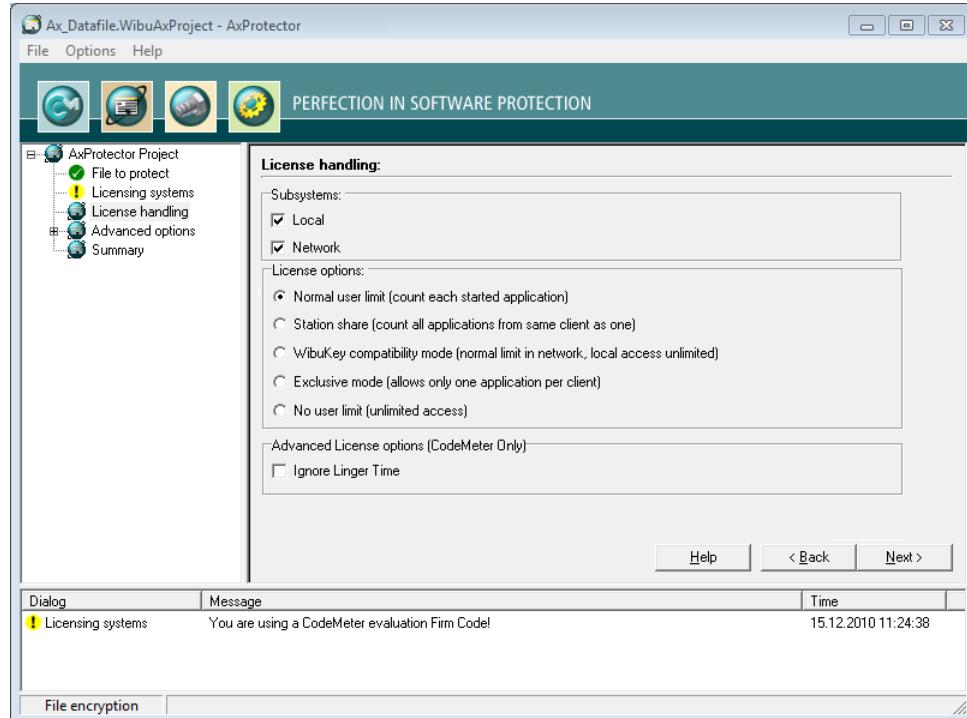


Figure 156: AxProtector - File Encryption "License Handling"

Subsystems

Here you can define in which subsystem (local or network) the protected application is to search for matching license(s) (commandline options see [here](#)²⁷²).

| 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 <i>CodeMeter License Server</i> runs and is activated as network server. On selecting both subsystems at the same time, the license is first sought locally and then subsequently on the network. |

License Options

In this group you define how started instances of the protected applications perform together with the allocation of licenses.

| Element | Description |
|----------------------------|---|
| Normal user limit | Here each started instance allocates a single license. It does not make a difference if the <i>CmContainer</i> 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 on 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 <u>recommends</u> the setting 'normal user limit' and 'station share'. |
| Exclusive Mode | Here a protected application can be started only <u>once</u> 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. |

Linger Time

| Element | Description |
|--------------------|--|
| 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 <i>CodeMeter</i> Developer Guide). |

7.6.1.4 Advanced Options

This input window lets you set further encrypting options.

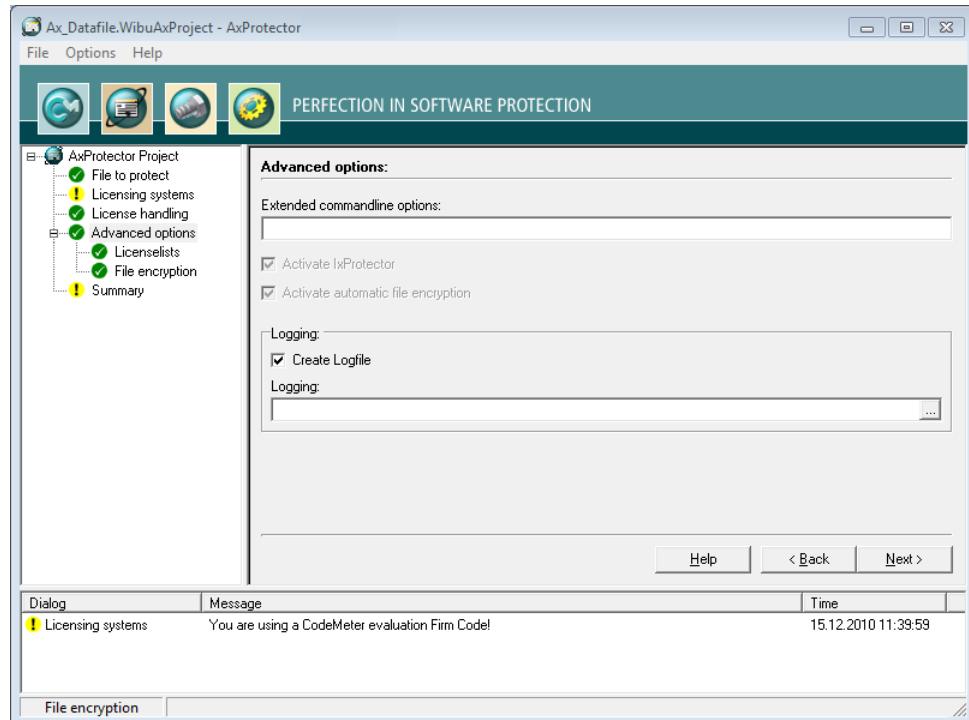


Figure 157: AxProtector - File Encryption "Advanced Options"

| Element | Description |
|------------------------------|--|
| Extended Commandline Options | <p>Here you are able to directly enter extended options or new feature functions using the <i>AxProtector</i> commandline.</p> <p> For more information please contact support at Wibu-Systems.</p> |
| Create Logfile | Activate this checkbox to create file logging for the activities of <i>AxProtector</i> . |
| Logging | <p>Specify the path and file name of this log file.</p> <p> 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.</p> |

7.6.1.4.1 License Lists

This menu item lets you manage license lists. Those you need to protect using *lXProtector* via the [Software Protection-API \(WUPI\)](#)²⁹⁶.

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

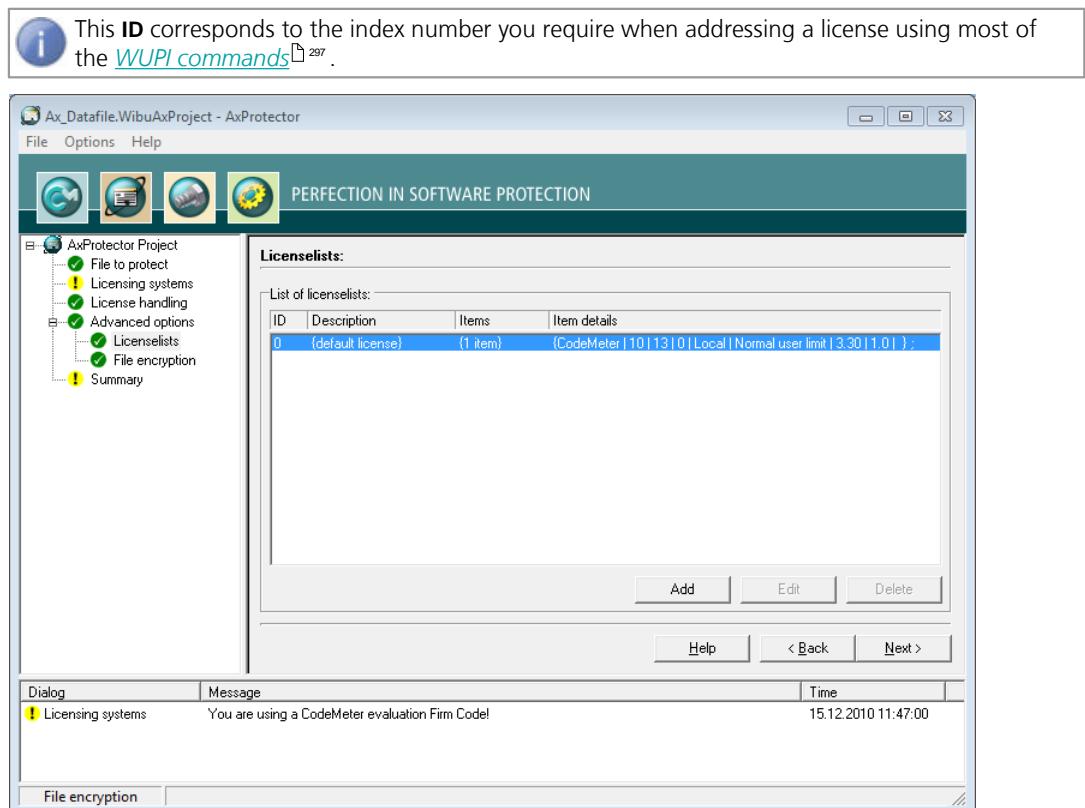
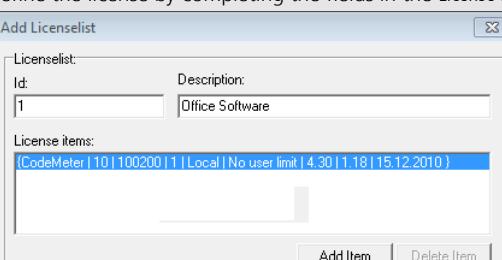
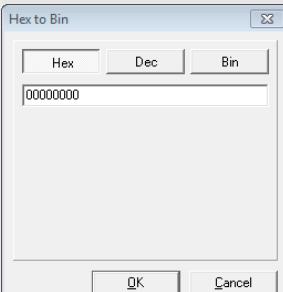


Figure 158: AxProtector - File Encryption "License Lists"

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 | <p>This ID uniquely identifies a license list and serves for referencing.</p> <p>i 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 from 1.</p> |

| Element | Description | | | | | | | | | | | | | | | | | | | | | |
|-------------------|---|--|---------------|---------------|----|--------|---|------------|-----------------|-------------------------|-------|---------------|------|-------------------|---------------|--------------------|------|------------|---------------------------------------|--|--|--|
| Description | <p>Here you will describe a license list with text.</p> <p>3. Define the license by completing the fields in the License item details group.</p>  <p>License item details:</p> <p>Licensing systems: CodeMeter</p> <table border="1"> <tr> <td>Firm Code:</td> <td>Product Code:</td> <td>Feature Code:</td> </tr> <tr> <td>10</td> <td>100200</td> <td>1</td> </tr> <tr> <td>Subsystem:</td> <td>License option:</td> <td>Minimum driver version:</td> </tr> <tr> <td>Local</td> <td>No user limit</td> <td>4.30</td> </tr> <tr> <td>Minimum Firmware:</td> <td>Release Date:</td> <td>Ignore Linger Time</td> </tr> <tr> <td>1.18</td> <td>15.12.2010</td> <td><input type="checkbox"/> WupiReadData</td> </tr> <tr> <td></td> <td></td> <td><input type="checkbox"/> WupiWriteData</td> </tr> </table> <p>Buttons: OK, Cancel, Help</p> | Firm Code: | Product Code: | Feature Code: | 10 | 100200 | 1 | Subsystem: | License option: | Minimum driver version: | Local | No user limit | 4.30 | Minimum Firmware: | Release Date: | Ignore Linger Time | 1.18 | 15.12.2010 | <input type="checkbox"/> WupiReadData | | | <input type="checkbox"/> WupiWriteData |
| Firm Code: | Product Code: | Feature Code: | | | | | | | | | | | | | | | | | | | | |
| 10 | 100200 | 1 | | | | | | | | | | | | | | | | | | | | |
| Subsystem: | License option: | Minimum driver version: | | | | | | | | | | | | | | | | | | | | |
| Local | No user limit | 4.30 | | | | | | | | | | | | | | | | | | | | |
| Minimum Firmware: | Release Date: | Ignore Linger Time | | | | | | | | | | | | | | | | | | | | |
| 1.18 | 15.12.2010 | <input type="checkbox"/> WupiReadData | | | | | | | | | | | | | | | | | | | | |
| | | <input type="checkbox"/> WupiWriteData | | | | | | | | | | | | | | | | | | | | |
| | Figure 159: AxProtector - File Encryption "Add License Lists" | | | | | | | | | | | | | | | | | | | | | |
| Licensing Systems | Select the licensing system used for protection of the license (<i>CmDongle</i> , <i>CmActLicense</i> or <i>WiluKey</i>). | | | | | | | | | | | | | | | | | | | | | |
| 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. | | | | | | | | | | | | | | | | | | | | | |
| |  | | | | | | | | | | | | | | | | | | | | | |
| Subsystem | Select the subsystem in which the protected application is to search (local or network), and | | | | | | | | | | | | | | | | | | | | | |

| Element | Description |
|--------------------|---|
| | <p>define the search order.</p> <p>License Options</p> <p>Select the options for license allocation:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Compatibility Mode • Exclusive mode • No User limit |
| Licensing Systems | Specify the required minimum driver version for the protected application. |
| Release Date | <p>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.</p> <p>To store the Release Date, please proceed as follows:</p> <ol style="list-style-type: none"> 1. Activate the "Release Date" checkbox to type in the Release Date. The current date is pre-set. 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 | <p>Activate this option to ignore a programmed Linger Time.</p> <p>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 <i>CodeMeter Developer Guide</i>).</p> |
| WupiReadData | Activate this option to read data ³⁰⁰ from the <i>CmContainer</i> if this data has been previously stored at a defined location. |
| WupiWriteData | Activate this option to write data ³⁰⁰ into a <i>CmContainer</i> 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.

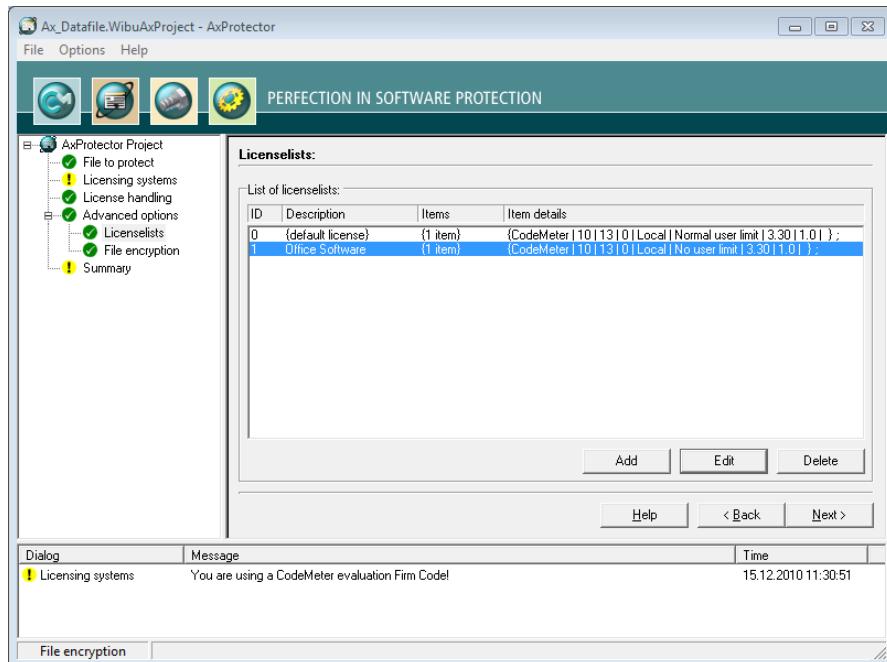


Figure 160: AxProtector - File Encryption "Completed License Lists"

7.6.1.4.2 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.

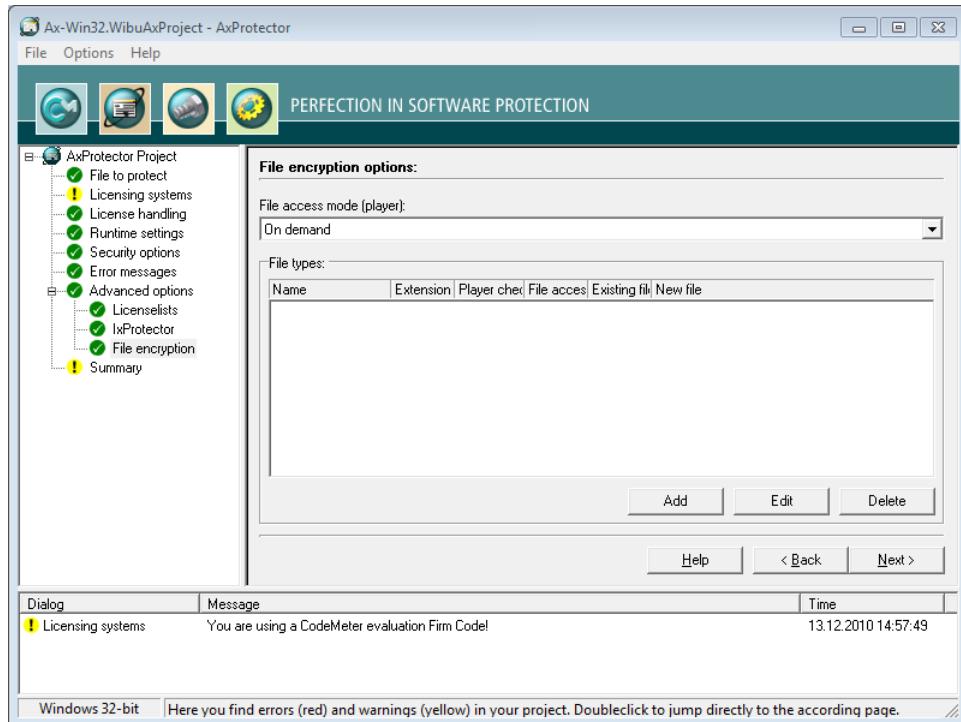


Figure 161: AxProtector - File Encryption "File Encryption"

| Element | Description |
|---------------|---|
| Add File Type | <p>1. Click on the "Add" button to add a new file type.</p> <p>The 'Add File Type' dialog box has fields for 'Name' and 'Extension'. Under 'Player check', it shows '0 - (default license)'. Under 'File access mode', it shows 'On demand'. In the 'Write options' section, 'Existing file' is selected for 'File access mode' and 'New file' is selected for 'File type'. Buttons for OK, Cancel, and Help are at the bottom.</p> |

Figure 162: 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

| Element | Description | | | | |
|---|---|---|---|-----------------|---|
| | <p>files.</p> <p>4. In the "Player Check" dropdown you define whether the license options of the accessing application (player) are checked when the encryption takes place.</p> <table border="1" data-bbox="318 337 1129 540"> <tr> <td data-bbox="318 337 408 402">License List</td><td data-bbox="408 337 1129 402">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.</td></tr> <tr> <td data-bbox="318 467 408 540">No player check</td><td data-bbox="408 467 1129 540">No check of the accessing application is performed.</td></tr> </table> | 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. |
| 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. | | | | |
| | <p>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.</p> <table border="1" data-bbox="318 625 1129 792"> <tr> <td data-bbox="318 625 408 792"></td><td data-bbox="408 625 1129 792"> <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> </td></tr> </table> |  | <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> | | |
|  | <p>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.</p> <p>Since the selection of different runtime settings for the player and the data are possible, at runtime the more restrictive settings apply.</p> | | | | |
| | <table border="1" data-bbox="318 792 1129 1089"> <tr> <td data-bbox="318 792 408 1089">On demand</td><td data-bbox="408 792 1129 1089"> <p>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.</p> <p> 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.</p> </td></tr> </table> | On demand | <p>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.</p> <p> 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.</p> | | |
| On demand | <p>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.</p> <p> 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.</p> | | | | |
| | <table border="1" data-bbox="318 1089 1129 1360"> <tr> <td data-bbox="318 1089 408 1360">At once</td><td data-bbox="408 1089 1129 1360"> <p>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.</p> <p> 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.</p> </td></tr> </table> | At once | <p>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.</p> <p> 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.</p> | | |
| At once | <p>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.</p> <p> 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.</p> | | | | |
| | <table border="1" data-bbox="318 1360 1129 1466"> <tr> <td data-bbox="318 1360 408 1466">Huge file mode</td><td data-bbox="408 1360 1129 1466"> <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> This mode requires no additional memory. Multiple accesses to the sa-</p> </td></tr> </table> | Huge file mode | <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> This mode requires no additional memory. Multiple accesses to the sa-</p> | | |
| Huge file mode | <p>The player reads the currently required parts of the protected file and decrypts them. This data is not cached in the memory.</p> <p> This mode requires no additional memory. Multiple accesses to the sa-</p> | | | | |

| Element | Description | | | | | | | | | | | | | | |
|--------------|---|----------|--|------------|--|--------------|--|-------|---------------------------------------|------------|----------------------------|--|---|--------------|--|
| | <p>This mode means that the data has to be read and decrypted each time. This mode is available for read access only.</p> | | | | | | | | | | | | | | |
| | <p>6. In the "Write Options" define the settings on how changes are saved.</p> <p>Existing File</p> <p>7. In this group you define the settings on how changes to an existing file are saved.</p> <table border="1"> <tr> <td>Original</td><td>Changes are allowed. Where the file was encrypted, it is re-encrypted. Unencrypted files are saved with no decryption.</td></tr> <tr> <td>No writing</td><td>Write actions are not allowed. Just read-only access is allowed.</td></tr> <tr> <td>License list</td><td>Changes are only encrypted using the license options defined in the selected license list.</td></tr> </table> <p>New File</p> <p>In this group you will define the settings on how new files are saved.</p> <table border="1"> <tr> <td>Plain</td><td>New files are only saved unencrypted.</td></tr> <tr> <td>No writing</td><td>New files cannot be saved.</td></tr> <tr> <td></td><td>(i) A new file is saved, however no data is saved to this file.</td></tr> <tr> <td>License List</td><td>New files are only encrypted using the license options defined in the selected license list.</td></tr> </table> | 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. | Plain | New files are only saved unencrypted. | No writing | New files cannot be saved. | | (i) 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. |
| 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. | | | | | | | | | | | | | | |
| Plain | New files are only saved unencrypted. | | | | | | | | | | | | | | |
| No writing | New files cannot be saved. | | | | | | | | | | | | | | |
| | (i) 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. | | | | | | | | | | | | | | |

7.6.1.5 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](#)²⁹⁵ type AxProtector.exe @*.wbc.
 Alternatively, using the "**File - export wbc file**" menu item, you can also create the corresponding *.wbc file.

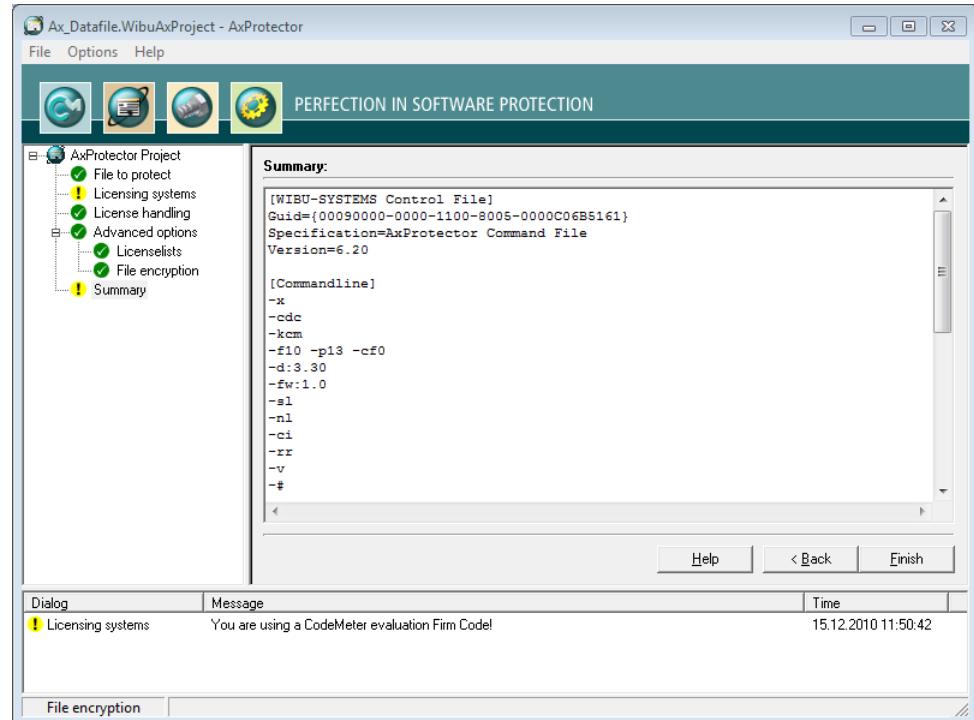


Figure 163: AxProtector - File Encryption "Summary"

| 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.

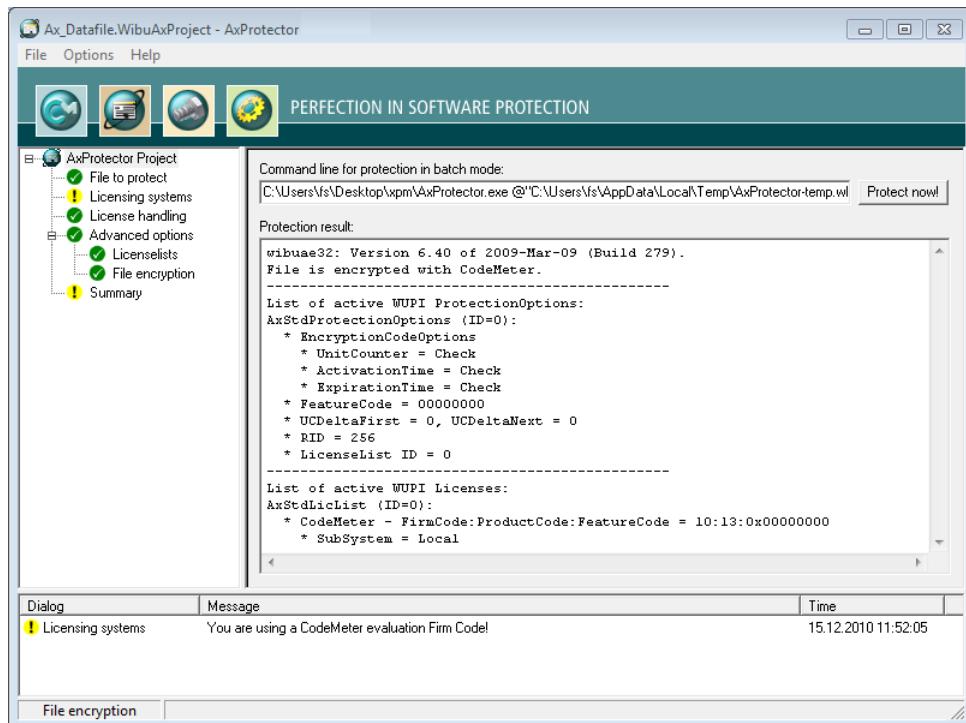


Figure 164: AxProtector - File Encryption "Encryption Result"

| Element | Description |
|-------------|---|
| Protect Now | <p>When you need to repeat the encryption operation, click the "Protect now" button. Then the <i>AxProtector</i> commandline is executed in batch mode.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i You are also able to copy the <i>AxProtector</i> 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. </div> |

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 | |
| AxProtectorNet.exe | |
| AxProtectorMacX | |
| AxProtector.jar | |
| 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

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 -A

valid for

Searches a pre-defined space within the application to be protected to insert security code.

This space is used only when defined big enough. The space must an initial *AxProtector* signature.

Option **-A[AES]**

valid for

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

Specifies the licensing system.

Parameter **-KCA**

uses *CmActLicense*.

Parameter **-KCM**

uses *CmDongle* (Default).

Parameter **-KWK [-x]**

uses *WibuKey*. *x* stands for:

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

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

Option **-Fx**

valid for

Specifies the Firm Code(x) to be used.

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

Option **-Px**

valid for

Specifies the Product Code (x) to be used.

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

| | |
|---|-----------------------|
| Option | -CFx |
| valid for | |
| Specifies the Feature Code (x) of the Feature Map to be used. | |
| Only valid when using <i>CodeMeter®</i> . | |
| valid for | |
| Defines the Release Date for encrypting and decrypting (only <i>CmDongle</i> and <i>CmActLicense</i> only). Specification is in format year, month, and optional hours, minutes, seconds, and the timezone. The input of [:now] applies the current date. | |
| Requires <i>CodeMeter®</i> Version 4.30 and Firmware-Version 1.18. | |
| Option | -D:v |
| valid for | |
| Specifies the minimum driver version (v). Input of v using (x.y). Default setting: <i>CodeMeter®</i> 4.20. Default setting: <i>WibuKey</i> 5.20. | |
| Option | -FW:v |
| valid for | |
| Defines the minimum firmware version(v). Input of v using (x.y). Default setting: <i>CodeMeter®</i> 1.0. Is not used with <i>WibuKey</i> | |
| Option | -S([L][N W] C) |
| valid for | |
| Specifies the search order of the subsystem when searching for licenses. The options N and W may be used alternatively only. | |
| Parameter | -SL |
| Uses the local subsystem (local). | |
| Parameter | -SN |
| Uses the network subsystem (network). | |

| | |
|---|-----------------------------|
| Parameter | -SLN |
| Uses first the local subsystem (local), then the network subsystem (network). | |
| Parameter | -SNL |
| Uses first the network subsystem (network), then the local subsystem (local). | |
| Parameter | -SW |
| Uses the Wide Area Network subsystem (WAN). | |
| Parameter | -SLW |
| Uses first the local subsystem (local), then the Wide Area Network subsystem (WAN). | |
| Parameter | -SWL |
| Uses first the Wide Area Network subsystem (WAN), then the local subsystem (local). | |
| Parameter | -SC |
| First searches the local and then the network subsystem, and if a network was found the network drive. | |
| Option | -N[C[A]L[A]N X X[A]] |
| valid for       | |
| Specifies the network access. | |
| Parameter | -NC[A] |
| convenient mode (compatibility mode): here each started instance on the network allocates a normal user limit, and the local access is unlimited (no user limit). | |
|  Since this is the default license allocation with <i>WibuKey</i> , this option ensures compatibility when both licensing systems are used at the same time. (A: uses auto cancel (<i>WibuKey</i> only)). | |
| Parameter | -NL[A] |
| normal user limit: here each started instance allocates a license regardless whether the <i>CmContainer</i> is found locally or on the network. (A: uses auto cancel (<i>WibuKey</i> only)). | |
| Parameter | -NN |
| no user limit: here any number of instances can be started locally or on the network. No licenses are allocated. | |
| Parameter | -NS |
| station share: here several started applications on a client allocate only a single license.  You use this option when allowing the end-user to start the protected application several times. On terminal server each session allocates a single license. In virtual machines each machine allocates a single license. | |

Parameter **-NX[A]**

exclusive mode: here only a single started instance per PC is allowed. Each access allocates a single license.

If the exclusive access is defined, the automatic runtime check is no longer internally activated.

Using option ['-car'](#)²⁷⁷ for the runtime check has now to be set explicitly.

(A: uses auto cancel (WibuKey only)).

7.7.3 Options for Encrypting and Decrypting

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

Encrypts the executable file using automatic encryption.

Parameter **-CAA <1>**

valid for    

Activates the security options (Advanced Protection Schemes, APS).

<1> covers the options [0 , 15]

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

| Option | Description |
|---|--|
| 1 | Resource encryption applies (APS 1) |
| 2 | Static modification applies (APS 2) |
| 4 | Dynamic modification applies (APS 3) |
| 8 | Extended static modification applies (APS 4) |
|  CAA6 | CAA6 applies APS 2 and 3 |
|  CAA13 | CAA13 applies APS 1, 4 and 8 |

Parameter **-CACt<,u>**

valid for    

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

Parameter **-CAD<m>**

valid for    

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.

| Parameter | -CAD<m> | | | | | | | | | | | | |
|-----------|---|--------|-------------|---|---|---|--|---|--|---|--|----|---|
| | <p>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.</p> <p>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.</p> | | | | | | | | | | | | |
| Parameter | -CAE | | | | | | | | | | | | |
| valid for |     <p>Activates instantly the detection that a has been removed from the PC ('plug-out') (<i>CmDongle</i> only).</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  If the connection to a <i>CmDongle</i> 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. </div> | | | | | | | | | | | | |
| Parameter | -CAG<1> | | | | | | | | | | | | |
| valid for |     <p>Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC). <> covers the options [0,127]</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  When applying more than one Anti-Debugging mechanism (ADC), you can combine options with "or". </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; padding-bottom: 5px;">Option</th><th style="text-align: left; padding-bottom: 5px;">Description</th></tr> </thead> <tbody> <tr> <td style="padding-bottom: 5px;">1</td><td style="padding-bottom: 5px;">Checks whether a debugger is attached to your application. In the case a debugger is detected, the application does not start (ADC1).</td></tr> <tr> <td style="padding-bottom: 5px;">2</td><td style="padding-bottom: 5px;">Checks additionally for Kernel debugger programs, e.g. "SoftICE". In the case a debugger is detected, the application does not start (ADC2).</td></tr> <tr> <td style="padding-bottom: 5px;">4</td><td style="padding-bottom: 5px;">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).</td></tr> <tr> <td style="padding-bottom: 5px;">8</td><td style="padding-bottom: 5px;">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).</td></tr> <tr> <td style="padding-bottom: 5px;">16</td><td style="padding-bottom: 5px;">Locking of the license entry and thus of the hardware when a debugger program has been detected (ADC5).</td></tr> </tbody> </table> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Using this option requires that the developer has programmed the <i>CmContainer</i> with a Firm Access Counter (FAC). The Firm Access Counter (FAC) is located at the Firm Item level of a <i>CmContainer</i>. 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. The <i>CmContainer</i> is locked when the FAC has a value of 0. </div> | 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). |
| 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). | | | | | | | | | | | | |

| Parameter | -CAG<1> |
|-----------|--|
| Option | Description |
| | Depending on the settings all licenses of a vendor can be locked if a hacker attack occurs. |
| | The owner / end-user of the locked <i>CmContainer</i> 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). |
| 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 <i>CmContainer</i>). |
| 256 | Firm Access Counter decrementing by 1 is performed (only in combination with ADC5 and <i>CmContainer</i>). |

| Parameter | -CAG<1> |
|-----------|---|
| valid for |  |
| | Activates Anti-Debugging mechanisms (Anti-Debug Checks, ADC). <1> covers the options [0 , 17] |
| | <p> When applying more than one Anti-Debugging mechanism (ADC), you can combine options by "or". The default setting for <1> is 17.</p> |
| 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). <p> Using this option requires that the developer has programmed the <i>CmContainer</i> with a Firm Access Counter (FAC). The Firm Access Counter (FAC) is located at the Firm Item level of a <i>CmContainer</i>. 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. The <i>CmContainer</i> is locked when the FAC has a value of 0.</p> |
| | Depending on the settings all licenses of a vendor can be locked if a hacker attack occurs. |
| | The owner / end-user of the locked <i>CmContainer</i> 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 CAG16). |

Parameter **-CAG<1>**

Option Description

on 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



Activates Anti-Debugging mechanisms (Anti-Debugging-Checks, ADC).

<1> covers the options [0, 7]



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

The default setting for <1> is 7.

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 version 6 and 7.

Parameter **-CAL**

valid for



Limits the automatic encryption to specified areas.

Parameter **-CAM**

valid for



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

Parameter **-CAR<t>,<m>**

valid for



Adds a runtime check to the automatic encryption.

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

| | |
|------------------|--|
| Parameter | -CAS<p> |
| valid for |   |
| | Specifies the size of the protected application to be encrypted. You enter the length, in percent, anywhere from 0 to 100%. The default setting is 75 percent. |
| Parameter | -CAT(t)(,u) |
| valid for |      |
| | On each start of the application, a Certified Time update is triggered. Then the application starts regardless of whether the update was successful or not, and writes it into the <i>CmContainer</i> . Then the application starts when the time difference between the Certified Time and the system's PC-Time is not greater than <t> specified in hours. <u> specifies the valid time span in hours within which the difference between the Certified Time and the system time is allowed to range without a new Certified Time update (CodeMeter® only). |
| |  <t> has to be equal or greater than <u>. |
| Parameter | -CAV |
| valid for |      |
| | Adds a virus check to the automatically encrypted application using a checksum. |
| Parameter | -CAZ |
| valid for |      |
| | Saves the time when the encryption was performed within the protected application (<i>CmContainer</i> 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][E],[H],[I],[M],[O],[Q],[R],[S],[X]]] |
| | Sets compatibility parameters. |
| Parameter | -CCA |
| valid for |    |
| | Defines the target system / subsystem in combination with the option -ccx for .NET executables including debugging of encrypted applications. <a> contains the target platform [1,2]. <s> contains the subsystem [9] 1: x86 / Intel 32 Bit 2: ArmV4i 9: Windows CE System |

Parameter `-CCB`

valid for   

  disables usage of proprietary (B)ase relocation representation.

Parameter `-CCE`

valid for   

Specifies that the PE is not enlarged.

Parameter `-CCH`

valid for   

Prevents all global hooking in a protected application.

Parameter `-CCI`

valid for   

Allows to use the protected application in a way that the added protection does not change the eventual existing loading sequence of DLL files. This replaces the option "`-CCM`" which is no longer required.

Parameter `-CCM`

valid for   

States that the protected application is to load the library `wibucrt32/64.dll` to avoid problems in the loading sequence by the library `msvcr*.dll`.

Parameter `-CCO`

valid for   

Activates special handling for ActiveX / OCX images.

Parameter `-CCQ`

valid for   

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

Parameter `-CCR`

valid for   

Deactivates the renaming of sections.

Parameter `-CCS`

valid for     

| | |
|------------------|-------------|
| Parameter | -CCS |
|------------------|-------------|

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 | -CCX |
|------------------|-------------|

valid for   

States that also so-called mixed-mode assemblies are protected. This allows to encrypt .NET assemblies which cannot be encrypted using *AxProtector .NET*. Next to Win32 also Win64/x64 mixed-mode assemblies can be protected using the native *AxProtector*.

The library `wbcor32/64.dll` is required to allow running the protected assembly.

| | |
|---------------|--------------------------------|
| Option | -CC[D[flags]],[[K],[S]] |
|---------------|--------------------------------|

valid for 

| | |
|------------------|-------------|
| 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)

| | |
|------------------|-------------|
| Parameter | -CCK |
|------------------|-------------|

valid for 

does not explicitly unload shared object.

| | |
|---------------|---|
| Option | -CD[C][H](K [CA] CM [WK]F[x Py]) |
|---------------|---|

valid for  

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

| | |
|------------------|-------------|
| Parameter | -CDC |
|------------------|-------------|

valid for  

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

| | |
|------------------|--------------|
| Parameter | -CDCH |
|------------------|--------------|

valid for  

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

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

valid for



Specifies the used licensing system:
 CA uses *CmActLicense*
 CM uses *CmDongle*
 WK uses *WibuKey*.

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

valid for



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

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

valid for



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

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

valid for



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

Parameter **-CIH**

valid for



Defines that Wupixxx functions are not dynamically interfere in the *IxProtector* process (hooking).

Parameter **-CIN**

valid for



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

Parameter **-CID**

valid for



Encrypts within executable files explicitly defined source code areas in order to use them with *IxProtector*. WUPI now is also supported on Mac and Linux (Option-*cid*). Currently, a dynamically loaded variant is applied.

Option **-CI**

valid for



Option -CI

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](#)²⁰⁰).

Option -CK<n>

valid for 

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  

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. | False 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 |

Option -CO(n))

The obfuscation attributes are always interpreted.

Option -CPA

valid for 

Deactivates encryption of property accessors.

Option -CMD<n>

valid for 

activates reencrypting of methods after discarding.

<n> allows specification of seconds.

Option -CML<n>

valid for 

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 

Encrypts class constructors in .NET (MSIL) code.

Option -CP<l>

gilt für  

Installs a cleanup mechanism deleting all created files and registry entries on exiting an application if it has been started on a *CmContainer*.

<l> , causes that all deleted entries are written to a log file into the directory where the application locates.

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

Defines additional checks while encryption and decryption operations are performed.

Parameter -EA

valid for     

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

C Checks if the Product Item Option Activation Time exists.

I Ignores the Product Item Option Activation Time (*CodeMeter®* only).

R Requires the Product Item Option Activation Time.

| | |
|------------------|---|
| Parameter | -EE |
| valid for | |
| | Activates an Expiration Time check. C Checks if the Product Item Option Expiration Time exists. I Ignores the Product Item Option Expiration Time (CodeMeter® only). R Requires the Product Item Option Expiration Time. |
| Parameter | -EF |
| valid for | |
| | Activates the decrement of the Firm Access Counter (CodeMeter® only). |
| Parameter | -EM |
| valid for | |
| | Activates an Maintenance Period check. C Checks, if the Product Item Option Maintenance Period exists. I Ignores the Product Item Option Maintenance Period (CodeMeter® only). R Requires the Product Item Option Maintenance Period. |
| Parameter | -ET |
| valid for | |
| | Enforces an Certified Time after the <i>CmContainer</i> is activated. |
| | This option requires an activated Expiration Time. |
| Parameter | -EU |
| valid for | |
| | Activates an Unit Counter check and the counter decrementing by the specified value <n>. S Checks and decrements at the start of the protected application only. R Checks and decrements on each runtime check. The option R includes the option S. For options S and R the following options exist: C Checks whether the Product Item Option Unit Counter exists (default setting). R(R) Requires the Product Item Option Unit Counter. <n> specifies the decrement. The default setting is 0. I Ignores the Product Item Option Unit Counter (CodeMeter® only). |

| | |
|------------------|---|
| Parameter | -EU |
| |  -eurr2 activates a required Unit Counter on each runtime check and decrements it each time by the value of 2. |
| Option | -RIDx[y] |
| valid for |      |
| | Specifies the number of RID variants (x) and traps (y). If RID=0 is set automatically, then the default value of 256 is used. |
| Option | -RIDIx[x,y] |
| valid for |      |
| | Specifies the number of RID variants (x) and traps (y) when using <i>IxProtector</i> (WUPI). If RID is set to a value of 0, automatically the default value (64/8) is applied. |
| Option | -G[o,I][["Marker",.]] |
| valid for |      |
| | Excludes the specified range from the encryption. <o> Defines the exclusion at the beginning of the range. <1> Defines the length of the range to be excluded (  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 |      |
| | Sets the minimum Firmware version in the encryption. |
| Option | -W[C[t]]E[t][P][U[c]] |
| valid for |      |
| | Specifies the threshold of issued warnings. |
| Parameter | -WC[t] |
| valid for |      |
| | Specifies the threshold <t> in hours for the Certified Time. |
| Parameter | -WE[t] |
| valid for |      |
| | Specifies the threshold <t> in days for the Expiration Time. |

Parameter `-WP[t]`

valid for

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

Parameter `-WU[c]`

valid for

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

Option `-SILVERLIGHT(3|4)`

valid for

Specifies that a Silverlight DLL is encrypted (only *CodeMeter®*).

- 3 the DLL to be protected is compiled for Silverlight 3 (Default).
- 4 the DLL to be protected is compiled for Silverlight 4.

Option `-XAP:Filename`

valid for

expands the silverlight protectee dll from its XAP file encrypts it, and replaces the old one in a copy of the XAP file (*CodeMeter®* only).Naming of the file must use the extension.Only valid together with option `-SILVERLIGHT(3 | 4)`

7.7.4 Runtime Options

Option `-I`

valid for

Specifies that the exception handling for plug-in DLL files is used.

This option exclusively works with DLL files. When the plug-in is loaded and no licensing system linked, the plug-in does not close the complete application, and does not issue error messages.

Option `-L:xx`

valid for

Specifies the language of the user-defined message texts.

`cn`: sets Chinese`de`: sets German`fr`: sets French`jp`: sets Japanese`us`: sets English (default setting)

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

valid for     

Specifies the output text of user messages of the protected application.

"msg" holds the string for the desired event.

Line breaks, inverted comma, tabs, etc. can be specified in the output text. Type "\n", "\r", "\t", "\" in the string at the desired position.

Parameter **-MA**

valid for     

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

Parameter **-MC**

valid for     

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

Parameter **-ME**

valid for     

Holds the text when an error has occurred.

Parameter **-MI**

valid for     

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

Parameter **-MS**

valid for     

Holds the text displayed while the protected application is started.

Parameter **-MT**

valid for     

Holds the text displayed when the date of the Expiration Time has been reached

Parameter **-MU**

valid for     

Holds the text displayed when the Unit Counter has reached a value of 0.

| Parameter | -MWC |
|-----------|------|
| valid for | |

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

| Parameter | -MWP |
|-----------|------|
| valid for | |

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

| Parameter | -MWT |
|-----------|------|
| valid for | |

Holds the text displayed when the end of the Expiration Time or the Usage Period has been reached.

| Parameter | -MWU |
|-----------|------|
| valid for | |

Holds the text displayed when the Unit Counter is about to reach a value of 0.

| Option | -U[:FileName] |
|-----------|---------------|
| valid for | |

Calls the file `UserMsgXX.dll` where XX stands for a country placeholder, e.g. De, Us, etc..

[:FileName] When specifying the `FileName`, the user-defined Message DLL holds the name `FileName-
meXX.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 only).

| | |
|-----------|--|
| valid for | |
|-----------|--|

Calls the specified class for the message handling.

The class must be a secondary class to the `com.wibu.xpm.MessageHandler`. For example, `com.wibu.xpmSwingMessageHandler` as default standard message handler of the runtime package.

| Option | -UM[:FileName] |
|-----------|----------------|
| valid for | |

Calls the user-defined message assembly `UserMsg` if this assembly exists.

If [:FileName] is specified the implemented message assembly holds the name <FileName>.dll.



Specify the name without a file extension *.dll.

Option **-UI**

valid for 

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

Option **-ANF**

valid for 

Specifies the text displayed when an assembly is not found.

Default setting: The assembly "#requiredassembly#" could not be found.

Option **-PROBING:<Name>**

valid for 

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 

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>.

Option **-TRAP[1:n]**

valid for 

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.

Option **-PRIO[0..31]<S>**

valid for 

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 

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

7.7.5 Java-specific Settings

Option `-ja:<params>`

valid for 

Specifies arguments transferred to the Main Class at runtime.

Option `-jb:<number>`

valid for 

Activates or de-activates a special error exception handling.



Contact Wibu-Systems support for more details.

Option `-jd:<ClassLoader>`

valid for 

Specifies an alternative WIBU ClassLoader.

Currently, the following ClassLoader are available:

| | |
|--------------------------|---|
| <code>ClassLoader</code> | <code>CassLoader</code> derived from <code>java.net.URLClassLoader</code> |
|--------------------------|---|

| | |
|----------------------------------|--|
| <code>DelegateClassLoader</code> | <code>ClassLoader</code> derived from <code>java.lang.ClassLoader</code> |
|----------------------------------|--|

Option `-jd:vmin[-vmax]`

valid for 

Specifies the used minimum (and maximum) Java version.

The version must match the format as specified in the system property '`java.version`'. The final number can be omitted.



e.g. `-jd:1.4-1.5.0_04` allows the runtime versions from 1.4 to Java 5 Update 0 Maintenance 4.

Option `-jh:[a|e|n]`

valid for 

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 Renames no encrypted classes.

Option `-jm:<Main-class>`

valid for 

Specifies the starting Main Class.

Option **-jn:[p|s|t]**

valid for 

Activates the native class load process.



This process requires an intervention into the source code of the application.

- p** Uses JVMPPI (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 

Specifies which classes are encrypted.

- w** Whitelist: all classes matching this list are encrypted.

- b** Blacklist: 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>],[lsf],[e:[e]]]<list>**

valid for 

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.

- 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 -jvm:valid for 

Considers the selected option for the virtual machine.

Some Java runtime settings require a separate internal handling for the project type *AxProtector Java*.

The following option is available:

<server>: The Java virtual machine is started with the server.

Option -jxvalid for Exits the application using the `System.exit()` call after the 'Main-Class' main method has returned a value.***IxProtector Java* with *AxProtector Version 9.0*****Method Encryption**Starting with Version 9.0 the option to encrypt single methods using *IxProtector* is introduced. Therefor the commandline option `-ci` is featured.

Please note that once you set the option `-ci`:

- the options `-jn:t` and `-jh:n` are activated by default and must not be specified.
- JVMPPI is no longer supported but JVMTI only
- the renaming of classes into `.class.wibu` is not possible
- encryption of single classes instead of `Jar` files is not possible
- classloader (except `SystemClassLoader / ToolsSysCI`) are not supported.

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 not protected | Method is not protected |
| @Protected | <p>Class is protected (corresponds to <code>@Protected(licenseList=0)</code>)</p> <p>Optional parameters:</p> <p><u>licenseList</u></p> <ul style="list-style-type: none"> • <code>@Protected(licenseList=1)</code> encrypts the class using license list 1 (or another specified index entry, e.g. 2, 3 etc.). <p><u>scope</u></p> <ul style="list-style-type: none"> • <code>@Protected(scope = {Class})</code> specifies that only this class is encrypted. | <p>Method is protected (corresponds to <code>@Protected(licenseList=0)</code>)</p> <p>Optional Parameter:</p> <p><u>licenseList</u></p> <ul style="list-style-type: none"> • <code>@Protected(licenseList=1)</code> encrypts the class using license list 1 (or another specified index entry, e.g. 2, 3 etc.). |

| Annotation | Class | Method |
|--------------|--|--|
| | <ul style="list-style-type: none"> • @Protected(scope = {Method}) specifies that encryption is performed for the methods only. 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. <p>The scope and licenseList options may be combined.</p> | |
| @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. |

Setting parameter in a XML file

In addition to WBC files, a XML format is supported. The parameter for the automatic encryption of an executable file may be integrated using the option `-@xml`²⁹⁵. Below see an example file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AxProtectorJava xmlns:wibu="http://wibu.com/2013/AxpJavaControlFile/1.0">
    <CommandLine>
        <Command>-kcm</Command>
        <Command>-f10</Command>
        <Command>-p13</Command>
        <Command>-ci</Command>
        <Command>-jb:24941</Command>
        <Command>-jb:25383:D:\Tools\apache-tomcat-7.0.34\lib\servlet-api.jar</Command>
        <Command>-o:D:\tools\apache-tomcat-7.0.34\webapps\TestWar.war</Command>
        <Command>TestWar.war</Command>
    </CommandLine>
    <Wupi>
        <LicenseList Index="0">
            <License>CM10-13</License>
            <License>CMA5010-13</License>
            <License>WK10-13</License>
        </LicenseList>
        <LicenseList Index="1">
            <License>CM10-13</License>
        </LicenseList>
        <License Name="CM10-13">
            <Type>CodeMeter</Type>
        </License>
    </Wupi>
</AxProtectorJava>
```

```

        <FirmCode>10</FirmCode>
        <ProductCode>13</ProductCode>
        <FeatureCode>1</FeatureCode>
        <SubSystem>WanLocal</SubSystem>
        <Access>StationShare</Access>
        <MinimumDriverVersion>5.00</MinimumDriverVersion>
    </License>
    <License Name="CMA5010-13">
        <Type>CodeMeterAct</Type>
        <FirmCode>5010</FirmCode>
        <ProductCode>13</ProductCode>
        <SubSystem>Local</SubSystem>
        <Access>UserLimit</Access>
    </License>
    <License Name="WK10-13">
        <Type>WibuKey</Type>
        <FirmCode>10</FirmCode>
        <ProductCode>13</ProductCode>
        <SubSystem>LocalLan</SubSystem>
        <Access>NoUserLimit</Access>
    </License>
</Wupi>
</AxProtectorJava>
```

Creating machine-readable Class files

If you set the parameter `-ci` for method encryption (`lxDProtector`), for external application which analyze annotations, e.g. Tomcat starting with Version 7, the option `-jff:[c|w]` is introduced. It 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 is embedded in the constants sections.

Option `-jff:[c|w]`

valid for 

defines the encryption result of methods.

 Setting the option `-ci` for method encryption is required.

- w creates encrypted (not readable class files (default)).
- c creates as result valid class files.

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. The new option `-jcp` allows the announcement of further external libraries.

Option -jcp:<additional Jar files>

valid for 

announces AxProtector that external libraries exist additionally to the ones stated in the classpath.

e.g. `-jcp: javaee-api-7.0.jar;someapi.jar`

7.7.6 Operational Options

Option -!

valid for    

Creates a command file (*.wbc).

Option -v

valid for    

Activates the verbose mode.

 In the case of  use -vn.

Option -#[File]

valid for    

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

Option -EXTRACT

valid for 

Prints content of assembly (enter -EXTRACT? for details).

Option -? or -h

valid for    

Shows the options in commandline mode.

Option -@cmds.wbc

valid for    

Specifies a *.wbc file holding parameters for the automatic encryption of an executable file.

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.

IxProtector

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](#)³⁰⁸, 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](#)³¹³ 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 provides access to the same license entries. For example, WUPI allocates the license entry also used by *AxProtector* and by calling [WupiGetHandle](#)³⁰⁹ you can read out the entry to be further processed using *CodeMeter Core API*.

8.1 IxProtector 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.

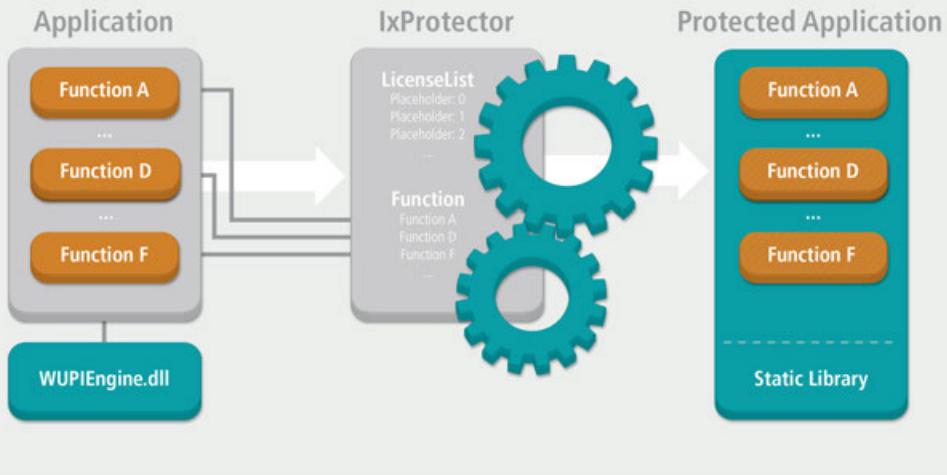


Figure 165: *IxProtector - Software Protection API - WUPI*

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.

8.2 WUPI Functions

The lean and effective *CodeMeter Software Protection API* provides the following functions.



Except for the functions ***WupiEncryptCode()*** and ***WupiDecryptCode()*** referring to *IxProtector*, all other functions relate to license lists.

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 <u>cannot</u> 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 (<i>LicenseList</i>) or on a <i>CmContainer</i> . |
| | 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 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
HiddenDataAccessCode=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](#)³⁰⁰ 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](#)³⁰¹ 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](#)³⁰⁰.

| 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](#)³⁰¹ 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

WupiWriteData (int iLicenseList, int iOffset, void* pvData, unsigned int cbData);

This function returns FALSE (0) if an error occurs, otherwise TRUE (1).
If the return value has a value of 0 call [WupiGetLastError](#)³⁰¹ 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](#)³⁰⁰.

| 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](#)³⁰¹ 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.

8.2.1 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](#)⁶⁴.

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](#)³⁰²,
2. [creating index-based license and function lists](#)³⁰²,
3. [programming license entries](#)³⁰⁶,
4. [integration into the source code](#)³⁰⁷,
5. [encryption of the application](#)³⁰⁸.

8.2.1.1 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.

8.2.1.2 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.

| Module | Firm Code, Product Code, Feature Code | Function Name |
|---------------|---------------------------------------|---------------|
| Basic License | 10:201000:1 | Save |
| Change Font | 10:201001:1 | 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.

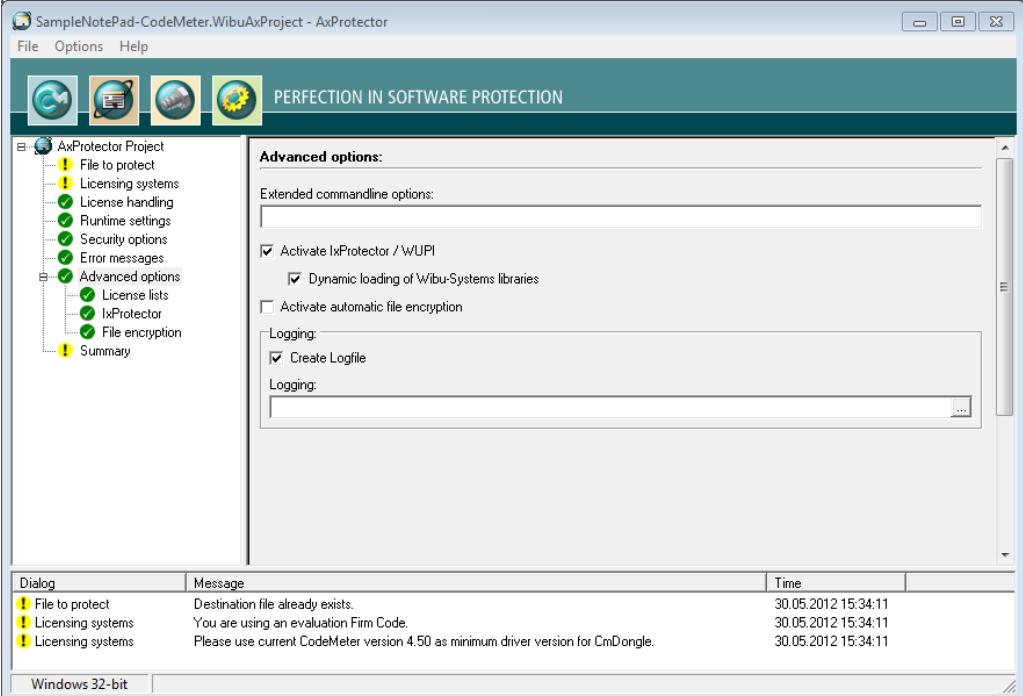


Figure 166: Activate *IxProtector* in *AxProtector*

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](#)"

 Unless you have a special reason, Wibu-Systems recommends using *IxProtector* within *AxProtector*.

2. Navigate to the "**License Lists**" input window.

| License lists: | | | |
|------------------------|-------------------|----------|--|
| List of license lists: | | | |
| ID | Description | Items | Item details |
| 0 | {default license} | {1 item} | {CmDongle 10 201000 1 Local - Network Normal user limit 4.30 1.18 0 none } ; |
| 1 | Font | {1 item} | {CmDongle 10 201001 1 Local - Network Normal user limit 4.30 1.18 0 none } ; |

[Add](#) [Edit](#) [Delete](#)

Figure 167: *AxProtector* - 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.

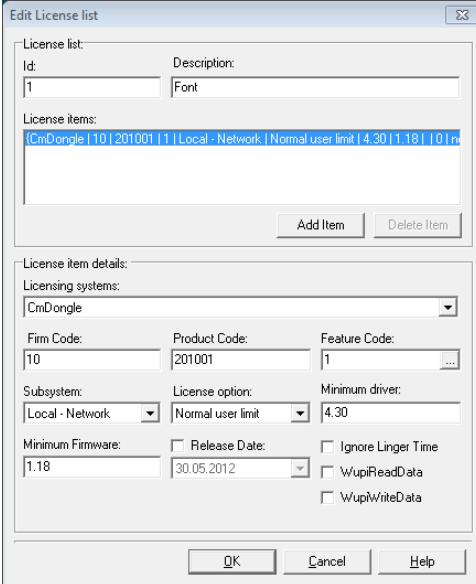


Figure 168: *IxProtector* - License List Entry

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](#)³⁰², i.e. Firm Code 10 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.

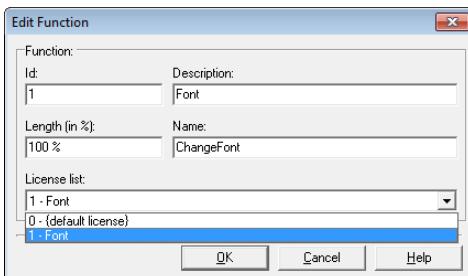


Figure 169: *IxProtector* - Function List Entry

In the "Second 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](#)³⁰² 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.

8.2.1.3 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](#)³²², [CmBoxPgm](#)³³¹ or [CodeMeter License Central](#)³⁶⁰.

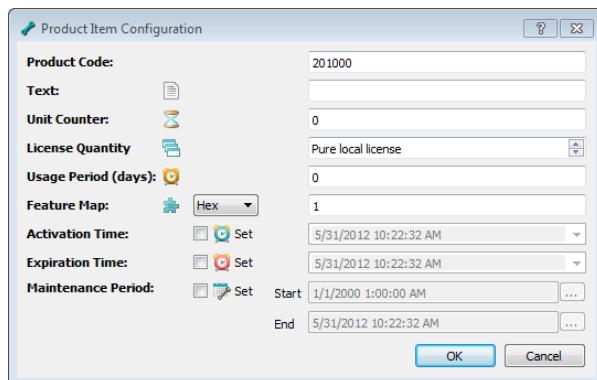
The programming covers:

- a Product Item with a Product Code 201000 for the license container with the Test Firm Code 10 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 10 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 10 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 10 and a Feature Code value of 1 for the Feature Map.

`CmBoxPgm.exe /f10 /p201000 /pfm1 /ca`

2. Create a Product Item with Product Code 201001 in license container with the Test Firm Code 10 and a Feature Code value of 1 for the Feature Map.

`CmBoxPgm.exe /f10 /p201001 /pfm1 /ca`

8.2.1.4 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 versi-
```

```
on."),
    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()
```

8.2.1.5 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.

8.3 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 sea-

mingless.

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](#)²⁹⁸ 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.

8.3.1 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](#)³¹³. 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" system menu item).

8.3.1.1 Access API

This API covers all functions to access a *CmContainer*.

| Command | Description |
|------------------|---|
| CmAccess | accesses a subsystem, a <i>CmContainer</i> , a Firm Item, or product entry (Product Item) in a given subsystem. |
| CmAccess2 | executes an access as CmAccess but provides extended functions (available since <i>CodeMeter® Version 3.30</i>).  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. |

8.3.1.2 Authentication API

This API covers all functions to execute authentication operations.

| Command | Description |
|-----------------------------------|---|
| <code>CmCalculateDigest</code> | calculates a 32 bytes hash value of an entered input sequence for the use in an authentication operation. The algorithm SHA-256 is applied. |
| <code>CmCalculateSignature</code> | calculates an ECDSA (Elliptic Curve Digital Signature Algorithm) signature with the specified hash value in the <i>CmContainer</i> . |
| <code>CmGetPublicKey</code> | reads the public key from a <i>CmContainer</i> . |
| <code>CmValidateSignature</code> | validates a ECDSA (Elliptic Curve Digital Signature Algorithm) signature with the specified public key. |

8.3.1.3 Enabling API

This API covers functions required for [Enabling](#)³⁸³ operations (directed activation or deactivation of complete *CmContainer*, or single Firm Item or license entries).

Basic Enabling functions may be implemented using `CmGetInfo()` and `CmProgram()`. Thus using the following functions is necessary only in rare cases.

| Command | Description |
|--|--|
| <code>CmEnablingWriteApplicationKey</code> | executes operations for the Enabling Access Code. |
| <code>CmEnablingGetApplicationContext</code> | reads the contents of the application to be activated or deactivated. |
| <code>CmEnablingGetChallenge</code> | reads the session ID from the <i>CmContainer</i> to be activated or deactivated. |
| <code>CmEnablingSendResponse</code> | activates a <i>CmContainer</i> or a <i>CmContainer</i> entry. |
| <code>CmEnablingWithdrawAccessRights</code> | deactivates a <i>CmContainer</i> or a <i>CmContainer</i> entry. |

8.3.1.4 Encryption API

This API covers all functions required for encryption and decryption operations of data.

| Command | Description |
|-------------------------------------|---|
| <code>CmCrypt, CmCrypt2</code> | encrypts or decrypts data directly or indirectly using a <i>CmContainer</i> . |
| <code>CmCryptEcies</code> | encrypts a specified byte sequence with the ECIES (Elliptic Curve Integrated Encryption Scheme) algorithm. |
| <code>CmCrypt_Sim</code> | encrypts or decrypts data directly or indirectly using the Firm Security Box entry of the desired Firm Code. |
| <code>CmCaluculatePioCoreKey</code> | calculates the core key for the encryption of the PIO Hidden Data. This operation requires a Firm Security Box. |
| <code>CmGetSecureData</code> | reads encrypted Hidden Data from the <i>CmContainer</i> using the Product Item Option Encryption Key (PIOEK). |

| Command | Description |
|-------------------------------|--|
| <code>CmDecryptPioData</code> | decrypts a Hidden Data sequence read using the Product Item Option Encryption Key (PIODK). |
| <code>CmGetPioDataKey</code> | calculates the key required to decrypt Hidden Data. |

8.3.1.5 Error Management API

This API covers functions required for handling error messages.

| Command | Description |
|----------------------------------|---|
| <code>CmConvertString</code> | converts the input in a specified string. |
| <code>CmGetLastErrorCode</code> | queries the last error code. |
| <code>CmGetLastErrorText</code> | queries the last error text. |
| <code>CmGetLastErrorText2</code> | queries the last error text as <code>CmGetLastErrorText</code> but provides extended functions. |
| <code>CmSetLastErrorText</code> | sets an error code in a internally used global error code variable. |

8.3.1.6 Management API

This API covers all functions required for *CodeMeter®* event-related operations.

| Command | Description |
|-------------------------------|--|
| <code>CmCheckEvents</code> | waits until a (local) event occurs, and returns the results. |
| <code>CmGetBoxes</code> | identifies all connected <i>CmContainer</i> which are connected to the same connection. |
| <code>CmGetBoxContents</code> | reads all entries of a <i>CmContainer</i> . |
| <code>CmGetInfo</code> | queries data in the <i>CmContainer</i> . Differently used query parameters result in different results. |
| <code>CmGetServers</code> | searches the local network for running <i>CodeMeter License Server</i> to which a <i>CmContainer</i> is connected. |
| <code>CmGetVersion</code> | calculates the version of the related <i>CodeMeter®</i> module. |

8.3.1.7 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 <i>CmContainer</i> 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 <i>CmContainer</i> entry. |
| CmProgram | programs different entries into a <i>CmContainer</i> . |
| CmValidateEntry | checks a specified sequence. |

8.3.1.8 Remote Update API

This API covers all functions required for the remote programming of license request and update files (*.WibuCmRaC and *.WibuCmRaU files).

| Command | Description |
|----------------------------|--|
| CmGetRemoteContext | saves the contents of a <i>CmContainer</i> in an encrypted and compressed remote context file (license request file) (*.WibuCmRaCf file). |
| CmSetRemoteContext2 | saves contents as CmGetRemoteContext but has an extended functional scope. |
| CmSetRemoteUpdate | programs a <i>CmContainer</i> with the specified remote activation update file (license update file) (*.WibuCmRaU file). This file holds all information to be program into a <i>CmContainer</i> . |
| CmSetRemoteUpdate2 | programs a <i>CmContainer</i> as CmSetRemoteUpdate but provides extended functions. |
| CmListRemoteUpdate | analyzes a remote activation update file (license update file) (*.WibuCmRaU file), and defines the serial numbers of all <i>CmContainer</i> 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.

8.3.1.9 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](#)³⁹⁴).

| 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 <i>CmContainer</i> . |

8.3.2 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.

8.3.2.1 Structure and Navigation

You access *CodeMeter API Guide* using *CodeMeter Start Center* or alternatively using the "**Start | All Programs | CodeMeter | Tools**" system menu item.

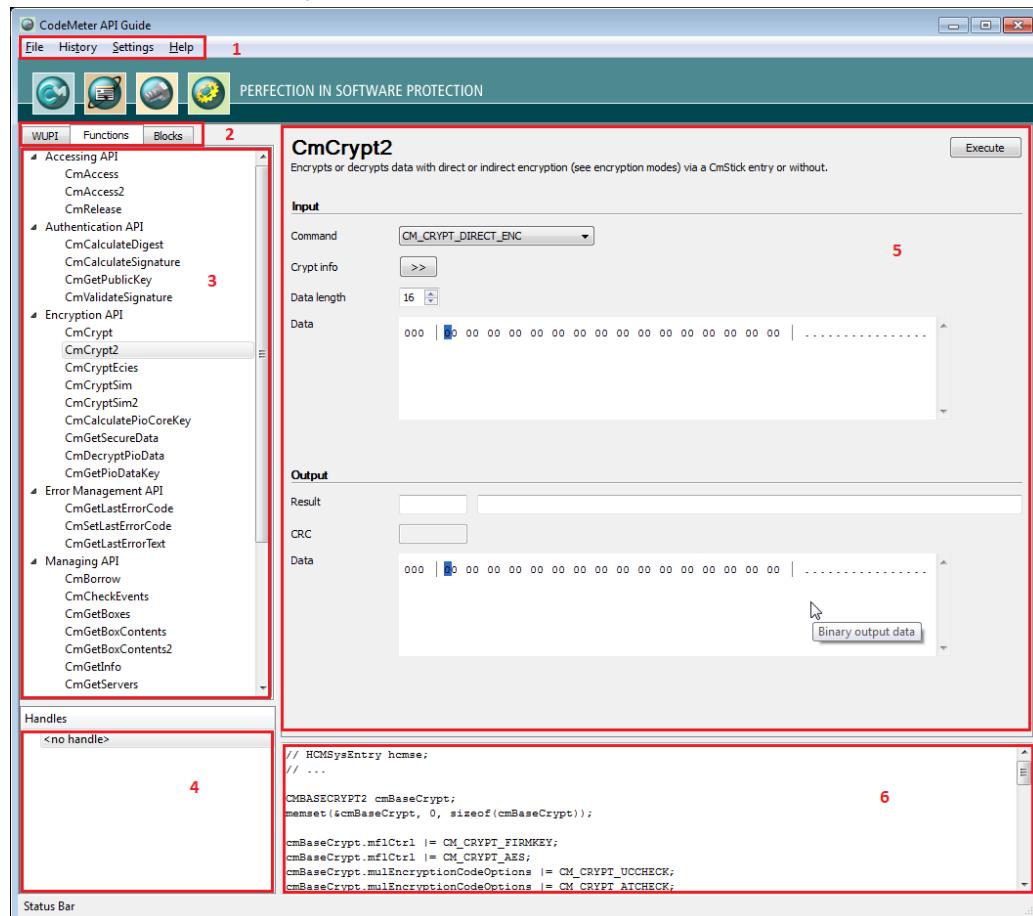


Figure 170: *CodeMeter API Guide* - Start GUI

The *CodeMeter API Guide* user interface consists of six separate areas:

- [Menu Bar](#) ³¹⁵ (1)
- [Tabs](#) ³¹⁶ to switch between WUPI, Core API, and Blocks (2)
- [Tree view](#) ³¹⁶ Function Calls (3)
- [Handle Display](#) ³¹⁷ Window (4)
- [Interactive Area](#) ³¹⁷: Input and Output field (5)
- [Source Code](#) ³¹⁷ Area (6)

8.3.2.2 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 <i>CodeMeter API Guide</i> . |

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. |
|---|

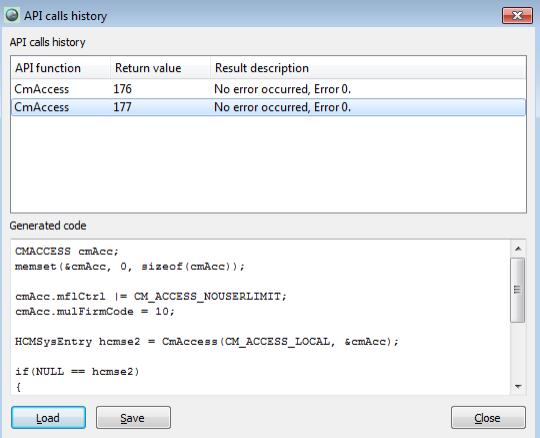
| Element | Description |
|---------|---|
| Load | Loads the *.WibuCmAPI file including generated source code into the history window. |
| Save | Saves the history of API calls in a *.WibuCmAPI file you are free to name and save at a desired location. |
| Show | Shows the history of your API calls including the generated source code in the history window.  |

Figure 171: *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 <i>CodeMeter API Guide</i> 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 <i>CodeMeter API Guide</i> version information. |

8.3.2.3 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 <i>Software Protection API</i> or WUPI (WIBU Universal Protection Interface) are clearly arranged by single functional areas. | | | | | | | | | | | | | | | | |
| Functions | In this tab you find the predominant part of the <i>CodeMeter Core API</i> functions. | | | | | | | | | | | | | | | | |
| Blocks | Next to single API functions, <i>CodeMeter API Guide</i> also provides you full functions blocks. These function blocks comprise the reading and writing of data from and into <i>CmContainer</i> , the execution of different encryption operations, and the activation of the <i>CmStick</i> LEDs. <table border="1"> <thead> <tr> <th>Block</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Read Data</td> <td>Reading Product Item Options Text, User Data, Protected Data and Protected Data.</td></tr> <tr> <td>Write Data</td> <td>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.</td></tr> <tr> <td>Sign Message</td> <td>Use of ECDSA (Elliptic Curve Digital Signature Algorithm) to sign data.</td></tr> <tr> <td>Verify Signature</td> <td>Verification of signed data.</td></tr> <tr> <td>Symmetric Encryption</td> <td>Symmetric encryption and decryption of binary data.</td></tr> <tr> <td>Asymmetric Encryption</td> <td>Asymmetric decryption of binary data.</td></tr> <tr> <td>Asymmetric Decryption</td> <td>Asymmetric decryption of binary data in the <i>CmContainer</i>.</td></tr> </tbody> </table> | 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 <i>CmContainer</i> . |
| 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 <i>CmContainer</i> . | | | | | | | | | | | | | | | | |

8.3.2.4 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.

8.3.2.5 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.

8.3.2.6 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.

8.3.2.7 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.

8.3.3 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 "WupiCalculator" you find for respective programming languages in the directory "%\Program Files%\WIBU SYSTEMS\AxProtector\DevKit\Samples\IxProtector\...\WupiCalculatorIndex".
Alternatively, find the samples using the system menu item "**Start | All Programs | CodeMeter | Samples**" or via [CodeMeter Start Center](#)⁶⁴.

8.3.3.1 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 Mac OS X or Makefile for Linux.

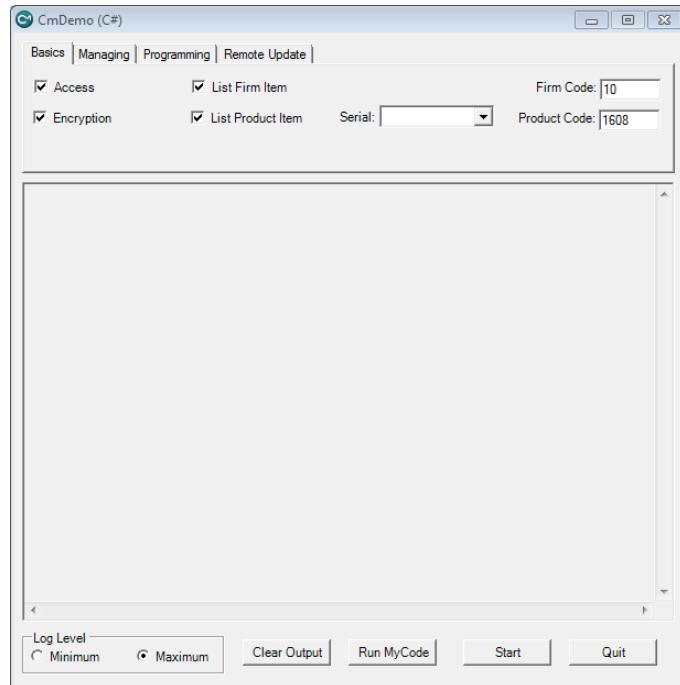


Figure 172: *CmDemo* - Overview

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 <i>CodeMeter®</i> . This section holds the source code you require for most of the <i>CodeMeter®</i> implementations. |
| Managing | This page demonstrates the complete read-out of a <i>CmContainer</i> and the querying of internal information of the <i>CmContainer</i> , e.g. version of <i>CodeMeter License Servers</i> or version of the hardware. In addition, the access to the LEDs, <i>CodeMeter®</i> 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 <i>CmContainer</i> . |

| Element | Description |
|----------------------|--|
| |  Note that you require a connected Firm Security Box. Meanwhile, these functions have been replaced by the <i>Programming API [High Level Application Programming Interface (HIP)]</i> . |
| Remote Update | This page demonstrates the <i>CodeMeter Field Activation Service (CmFAS)</i> , i.e. the remote programming of <i>CmContainer</i> without having to re-send altered <i>CmContainer</i> . |

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".  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. |

8.3.3.2 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.

8.3.3.3 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](#)³⁰¹.

9 Programming of CmContainer and Licensing Management

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

By the way, 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 license request file (*.wibuCmRaC) send by the licensee to the licensor into a license update file (*.wibuCmRaU) and the subsequent import by the licensee into his/her *CmContainer*.
- **protocol-based (SOAP)**: programming and managing of license request and license update files (*.WibuCmRaC and *.WibuCmRaU) is done by the Internet supported network protocol SOAP (Simple Object Access Protocol) using [*CodeMeter License Central*](#).

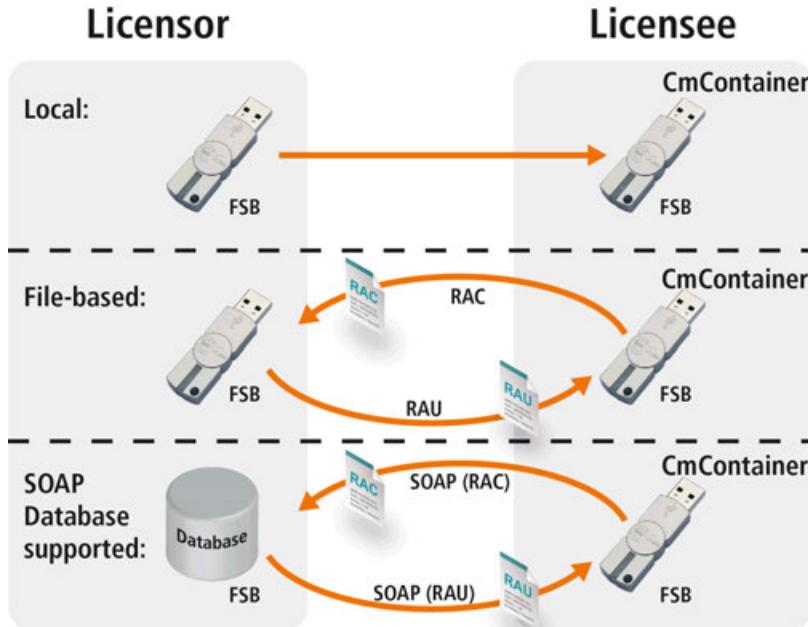


Figure 173: CmContainer Programming Options

For all three methods *CodeMeter®* provides several tools:

- [CmBoxPgm](#)³³¹: commandline tool for batch programming of CmContainer in production.
- [CodeMeter License Editor](#)³³²: graphic tool to program CmDongles or local tests of licensing strategies.
- [CodeMeter License Central](#)³³⁰: 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](#)³³⁸, 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 system menu item "**Start | All Programs | CodeMeter | Documentation | Programming-API**".

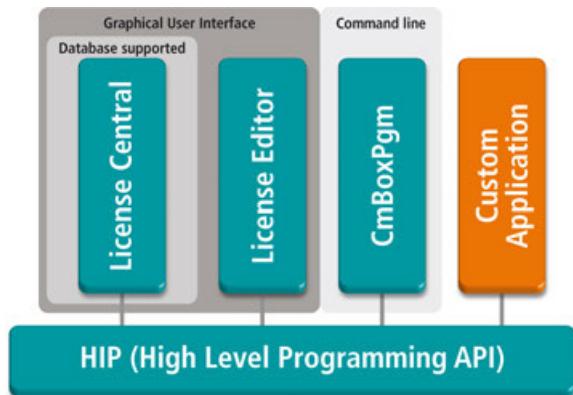


Figure 174: Tools for *CmContainer* Programming

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](#)⁶³ (*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](#)⁶³ or by using the "Start | All Programs | CodeMeter | Tools" system menu item.

9.1.1 Structure and Navigation

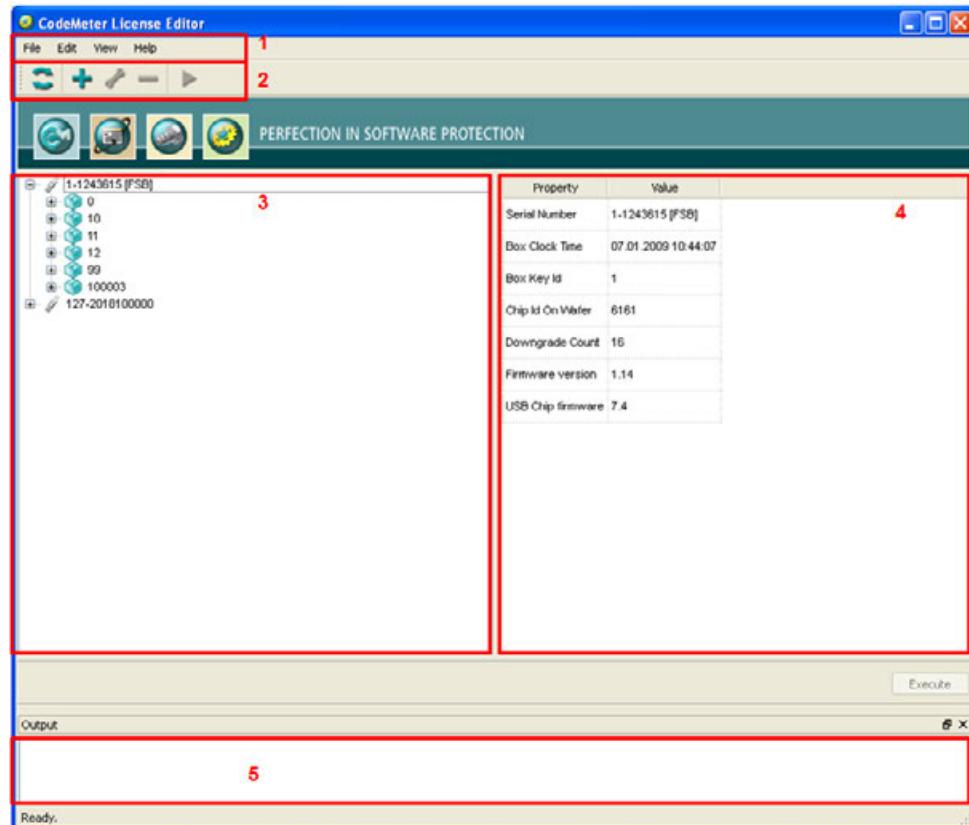


Figure 175: *CodeMeter License Editor - Start Screen*

The *CodeMeter License Editor* graphical user interface consists of five separate areas:

- [Menu Bar](#) ³²⁴ (1)
- [Symbol Bar](#) ³²⁴ (2)
- [Tree View Window](#) ³²⁵ (3)
- [Display Window](#) ³²⁶ (4)
- [Output Window](#) ³²⁶ (5)

9.1.1.1 Menu Bar

File Menu

| Element | Description |
|--------------------------------|--|
| Remote Programming Mode | Loads license information in <i>CmDongles</i> to <i>CodeMeter License Editor</i> by using *.WibuCmRaC or *.WibuCmRaM files. |
| Direct Programming Mode | Loads license information in <i>CmDongles</i> to <i>CodeMeter License Editor</i> . This menu item corresponds to the command "Execute". |
| Exit |  Closes <i>CodeMeter License Editor</i> . In order to exit <i>CodeMeter License Editor</i> 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 <i>CmDongle</i> . In order to save changes using the keyboard click the <CTRL+X> key combination. |
| Refresh |  Refreshes the view of the licenses in a <i>CmDongle</i> . In order to refresh the Item view using the keyboard click the <CTRL+R> key combination. |

View Menu

| Element | Description |
|---------------|--|
| Status | Allows you to reactivate the output window you deactivated by using the  control. |

Help Menu

| Element | Description |
|--------------|--|
| Help | Selecting this menu item opens the online help on <i>CodeMeter License Editor</i> . |
| About | Selecting this menu item opens a window informing about the <i>CodeMeter License Editor</i> version you use. |

9.1.1.2 Symbol Bar



Figure 176: *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 Items).

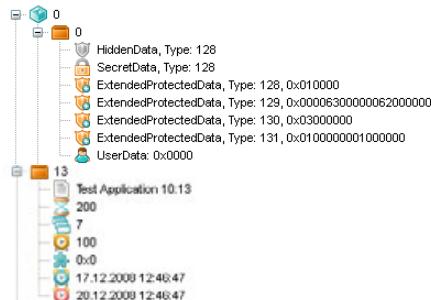


Figure 177: *CodeMeter License Editor* - Tree View

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 |

| Symbol | Object |
|--------|-------------------------|
| | Protected Data |
| | Extended 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 | 13 |
| Text | Test Application 10:13 |
| Unit Counter | 200 |
| License Quantity | 7 |
| Usage Period | 100 |
| Feature Map | 0x0 |
| Activation Time | 17.12.2008 12:46:47 |
| Expiration Time | 20.12.2008 12:46:47 |

Figure 178: 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.

```
Output
Mi 7. Jan 11:11:42 2009: (Message) : Executing commands...
Mi 7. Jan 11:11:45 2009: (Message) : Updating ProductItem 13 of CmStick with Serialnumber 1-1243615 ...done.
Mi 7. Jan 11:11:45 2009: (Message) : Executing commands finished.

Ready.
```

Figure 179: 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 | Status".

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](#)¹⁶³ or by the system menu "**Start | All Programs | CodeMeter | Tools**".

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 in **Remote Programming Mode** you load a *.WibuCmRac or *.WibuCmRaM file 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 Remote Programming Mode

Using the File Menu item "**File | Remote Programming Mode**" you load the respective *.WibuCmRac or *.WibuCmRaM files holding the license details for further editing.

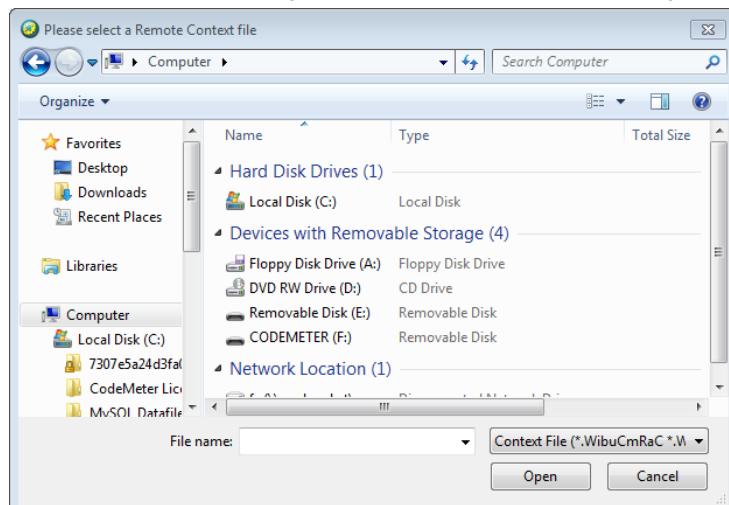


Figure 180: *CodeMeter License Editor* - Remote Context File

9.1.2.3 Creating and Editing a Firm Item

Select the "**Add Item**" or "**Modify Item**" item while on the navigation level of a *CmDongles* via:

- the 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 specify data for configuring a Firm Item.

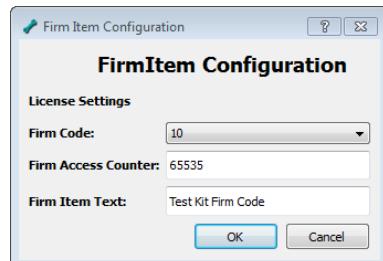


Figure 181: *CodeMeter License Editor* - Create Firm Item and Editing

| 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. This value is decremented by 1 in the case a special option has been set for a specified encryption or decryption operation. The default value is set to 65535 (0xFFFF) and the Firm Access Counter is deactivated. However the value can be programmed. If this value is 0, this Firm Item is locked ³³⁵ for encryption and decryption operations. |
| Firm Item Text | Specify the text which describes the Firm Item in greater detail. |

9.1.2.4 Deleting Firm Items

Select the item "**Delete Item**" while on the navigation level of a Firm Item via:

- the symbol in the context menu (right mouse-click) or in the symbol bar
- the **Edit** menu item of the same name.

The following dialog requires you to confirm the deletion.

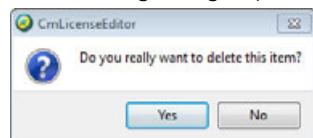


Figure 182: *CodeMeter License Editor* - Delete Firm Item

Depending on the license type of your Firm Code it may be possible that deleting a Firm Item will not work with your FSB. This is designed to avoid unintentional deletion. However, this option may be activated any time free of charge.

9.1.2.5 Creating and Editing a Product Item

Select the item "**Add Item**" or "**Modify Item**" while on the navigation level of a Firm Item via:

- the 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 specify data configuring a Product Item.

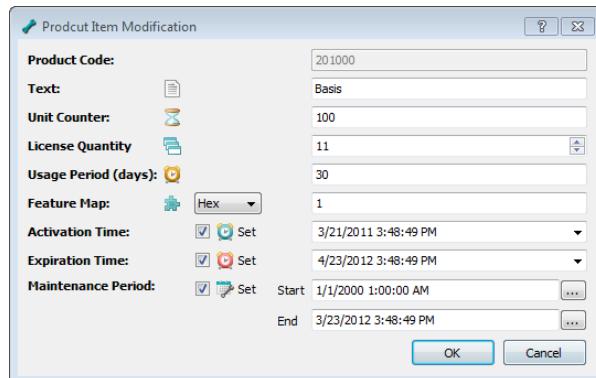


Figure 183: *CodeMeter License Editor - Create Product Item*

Next to the Product Code field, there exist seven more Product Item Options allowing you to configure the license.

| Element | Description |
|---------|--|
| | Specify a text which describes the Product Code - the actual product - in greater detail. |
| | 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. |
| | Specify a number of days for which the license is to be valid. |
| | Specify the favored combination of features to be activated. Feature may comprise modules, functions, or different versions. A dialog allows you to specify the value alternatively in the formats binary (Bin) or decimal (Dec). |
| | Activate the " Set " option. 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. |
| | Activate the " Set " option. Specify a date when the validity of a license is to expire using the calendar control. |

| Element | Description |
|---|--|
| Maintenance Period  | <p>Activate the "Set" option. Specify in the date fields start and end of the Maintenance Period for which the license is to be valid.</p>  Requires CodeMeter® Firmware 1.18 or higher. |

Figure 184: *CodeMeter License Editor - Create Maintenance Period*

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.

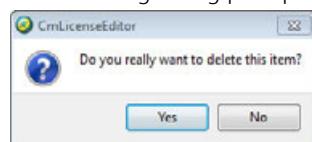
Click the **"OK"** button to save the settings.

9.1.2.6 Deleting a Product Item

Select **"Delete Item"** while on the navigation level of a Product Item using:

- the  symbol in the context menu (right mouse-click) or in the symbol bar
- the **Edit** menu item of the same name

The following dialog prompts you to confirm the deletion.

Figure 185: *CodeMeter License Editor - Delete Product Item*

9.1.2.7 Executing the Programming

Click on the **"Execute"** button above the output window. This saves all changes you have made for Firm and Product Items and transfers the detailed license information into the connected *CmDongles*.

Alternatively, you program the *CmDongle* using the **"Edit | Execute"** item or the 

9.2 CmBoxPgm

Besides programming of *CmDongle* using [CodeMeter License Editor](#)²² and [CodeMeter License Central](#)²³, 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 your 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.

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,
- Enabling Blocks.

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, Product Items or Enabling Blocks. 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 `cmbxpgm.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, Product Item or an Enabling Block.

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 explicitly 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](#)³³⁶ the Firm Item Text or to edit the text attribute in the file *CmFirm.wbc*. You find the file *CmFirm.wbc* in the directory "C:\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).

/1 - List

List the contents of selected *CmContainer*, Firm Items, Product Items or Product Item Options.

Programming examples

CmBoxPgm /1

Lists the content of the first *CmContainer* which is not a Firm Security Box. Corresponds to **CmBoxPgm /qn1 /1**

CmBoxPgm /qb1 /1

List the content of a *CmContainer* with an index 0.

It does not make a difference whether the *CmContainer* holds an Firm Security Box entry or not. Corresponds to **CmBoxPgm /qn1:f /1**

CmBoxPgm.exe /qs1-1234 /1

List the content of a *CmContainer* with the serial number of 1-1234.

CmBoxPgm.exe /qn2,4:f /1

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 <i>CmContainer</i> 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 | /qnx[,y][:f] - Box Index Range |
| | Use this option to determine the index range of <i>CmContainer</i> to be programmed. Two indices (1, 2, 3,...), representing the range's lower and upper limit, may be specified.  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 <i>CmContainer</i> to be programmed. Expects a decimal value as argument. It is recommended to use both parameters to uniquely identify a <i>CmContainer</i> .  Must not be used together with options /qb or /qn . |
| Syntax | /qs<Serial Number>  /qs1-12345 or /qs2-12345 The mask parameter refers to the used CodeMeter® chip version. |
| Command | /pwd - Box password |
| | Changes the <i>CmContainer</i> password. |
| Syntax | /pwd "<old Password>"=<new password>" |
| Command | /r - Recursive Removal |
| | Removes every public entry or any entries whose removal is covered by an appropriate license entry in a connected FSB from the specified <i>CmContainer</i> .  This option works only with license models which base on transactions (adding and updating of Product Items - PaPu) since the Firm Item is deleted. In the case of extensive transactions of single <i>CmContainer</i> (updating Firm Item - Fa) this is not possible due to security reasons. |
| Syntax | /r  /q2, 4 /r Performs a cleanup for the <i>CmContainer</i> with the index range from 2 to 4, excluding FSBs. |
| Command | /rau - Remote Activation Update |
| | Executes the programming sequences stored within the specified Remote Activation Update file on the target <i>CmContainer</i> as far as applicable. |
| Syntax | /rau:<RaU file>" |

Programming examples

CmBoxPgm

lists the content of the first *CmContainer* which is not a Firm Security Box. Corre-

| | |
|-----------------------|---|
| | sponds to CmBoxPgm /qn1 /1 |
| CmBoxPgm /qb1 /1 | lists the content of the <i>CmContainer</i> with index 1. It does not matter whether the <i>CmContainer</i> holds a Firm Security Box entry or not. Corresponds to CmBoxPgm /qn1:f /1 |
| CmBoxPgm /qs1-1234 /1 | lists the content of the <i>CmContainer</i> with the serial number 1–1234. |
| CmBoxPgm /qn2,4:f /1 | lists the content of the <i>CmContainer</i> 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 |
| Syntax | Defines the Firm Codes to be used. Expects an unsigned decimal value as argument. |
| Command | /fac - Firm Access Counter (FAC) |
| Syntax | 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 | /fpta - Firm Precise Time, absolute |
| Syntax | 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 |
| Syntax | Adds the specified number of days to the current value of the Firm Precise Time. If the Firm Item doesn't exist yet, the current system time plus the specified offset will be set as Firm Precise Time. Expects an integer value greater than or equal to zero as argument. |
| |  If this Firm Item has not yet been created, the system time plus the specified number is used as Firm Precise Time. For example, /fptr1 corresponds to 1 day immediately starting. |
| Syntax | /fptr<number of days> |

| | |
|----------------|-----------------------------------|
| Command | /ft - Firm Item Text |
| Syntax | /ft : "<Text>" |
| Command | /fuc - Firm Update Counter |
| 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 <i>CmContainer</i> 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 <i>CmContainer</i> . 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 /1 /f206 /cu /1 | Lists the content of the <i>CmContainer</i> , 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 <i>CmContainer</i> including the specified status. |
|--|--|

| Command | /p - Product Code | | | | | | |
|---|--|---|------|--|-----------------|---|-----------|
| | <p>Defines the Product Code to be used. Expects an unsigned decimal value as argument.</p> <p> 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.</p> <p>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).</p> | | | | | | |
| Syntax | /p<value>[=<new value>][:<Feature Code> <reference>][,<TVB dep.>] | | | | | | |
| e.g. | <table border="1"> <tr> <td>Selection of first Product Items with Product Code 13</td><td>/p13</td></tr> <tr> <td>Selection of Product Item with Product Code 13, Feature Map=0x00000001</td><td>/p13:0x00000001</td></tr> <tr> <td>Selection of Product Item with Product Code 13, Product Item Reference = 16</td><td>/p13:[16]</td></tr> </table> | 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] |
| 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 | /pat - Activation Time | | | | | | |
| | <p>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).</p> | | | | | | |
| Syntax | /pat - [a<date> r<offset>][,<TVB dep.>] | | | | | | |
| | <p> Removes the PIO (-) or sets an (a)bolute or a (r)eative Activation Time.</p> | | | | | | |
| Command | /pata - Activation Time, absolute | | | | | | |
| | <p>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.</p> <p> If the time zone is omitted, the system's time zone will be used instead.</p> <p>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).</p> | | | | | | |
| Syntax | <p>/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.>]</p> | | | | | | |
| e.g. | Sets the Activation Time to December 31st, 2012, 1 second to midnight, UTC /pata2012Dec31,23:59:59UTC | | | | | | |
| Command | /patr - Activation Time, relative | | | | | | |
| | <p>Adds the specified number of days to the current Activation Time. Expects an integer value greater than or equal to zero as argument.</p> <p> 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.</p> | | | | | | |

| | |
|--|---|
| Command | /patr - Activation Time, relative |
| | 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 | <code>/patr<days>[,<TVB dep.>]</code> |
| 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 | <code>/pcoli-</code> <code>/pcoli:"<text>"[,<TVB dep.>]</code> |
| |  Deleting the PIO (-) or setting the specified text. |
| Command | /ped - Extended Protected Data |
| | Adds, updates or 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 | <code>/ped<extended type>- [:0x<hex data>][,<TVB dep.>]</code> |
| |  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. |
|  | <code>/ped0:0x75BCD15</code> Adds the decimal value 123456789 to the field (type) 0. <code>/ped2-</code> 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 | <code>/pet- [a<date> r<offset>][,<TVB dep.>]</code> |
| |  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. Expect 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). |

| Command | /peta - Expiration Time, absolute | |
|---------|--|---|
| Syntax | <pre>/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.>]</pre> | |
| 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 | |
| | <p>Adds the specified number of days to the current ExpirationTime. Expects an integer value greater than or equal to zero as argument.</p> <p> If this Firm Item does not exist yet, the current system time plus the specified offset will be set as Expiration Time. For example: /petr1 corresponds to 1 day immediately starting.</p> <p>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).</p> | |
| Syntax | /petr<days>[,<TVB dep.>] | |
| e.g. | Extends the Expiration Time by 30 days. | /petr30 |
| Command | /pfm - Feature Map | |
| | <p>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).</p> | |
| Syntax | <pre>/pfm-[<value>][,<TVB dep.>]</pre> <p> Deletes the PIO (-).</p> | |
| Command | /phd - Hidden Data | |
| | <p>Adds, updates or 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. 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).</p> | |
| Syntax | <p>Fills the Hidden Data PIO with user-defined data. /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>-</p> | |

| Command | /phd - Hidden Data |
|--|---|
|  | <p>/phd15 : 0x1122334455 Fills the field (type)15 of the Hidden 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</p> <p>/phd16 ,<acc. code> : r32 Fills the field (type)16 of the Hidden Data PIO with 32 bytes of random data.. The Access Code <acc. code> can be a text input or an input of 16 bytes in hexadecimal format..</p> |
| Command | /plq - License Quantity |
| Syntax | <p>Adds, updates or deletes the PIO License Quantity 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).</p> <p>/plq- [<counter>] [,<TVB dep.>] :w If the parameter w is specified, the license access on a Wide Area Network (WAN) is available. Note that you need a separate Firm Security Box (FSB) license entry is required you are able to receive by Wibu-Systems.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Removes the PIO (-). </div> |
|  | /plq15 adds 15 licenses to the PIO. |
| Command | /plt - Linger Time |
| Syntax | <p>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).</p> <p>/plt- [<seconds>] [,<TVB dep.>]</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Removes the PIO (-). </div> |
|  | /plt15 adds a Linger Time of 15 seconds to the PIO. |
| Command | /pmp - Maintenance Period |
| Syntax | <p>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).</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Requires CodeMeter® Firmware 1.18 or higher. </div> <p>/pmp- [d[<start date>],<end date>] i[<start value>,<end value>] [,<TVB dep.>]</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  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. </div> |

| | |
|---|--|
| Command | /pmpd – Maintenance Period (Date) |
| | <p>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.</p> <p> The start date may be omitted. In this case it is set to 2000-01-01, 00:00:00 UTC</p> <p>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).</p> <p> Requires CodeMeter® Firmware 1.18 or higher.</p> |
| Syntax | <code>/pmpd[<start time>[,<end time>[,<TVB dep.>]]</code> |
|  | <pre>/pmpd2011Jul01,00:00:00,2012Jun30,23:59:59 or /pmpd2011-07-01T00:00:00,2012-06-30T23:59:59</pre> <p>Sets a one-year Maintenance Period beginning with July 1st, 2011.</p> |
| Command | /pmpi – Maintenance Period (Integer) |
| | <p>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.</p> <p> The start date may be omitted. In this case it is set to a value of 0.</p> <p>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).</p> <p> Requires CodeMeter® Firmware 1.18 or higher.</p> |
| Syntax | <code>/pmpi[<start value>[,<end value>[,<TVB dep.>]]</code> |
|  | <pre>/pmpi394416000</pre> <p>Sets a Maintenance Period until July 1st, 2012. The difference of 4565 days to 1.1.2000 is 394416000 seconds.</p> |
| Command | /pnwc - Network License Counter |
| | <p> Deprecated, please use option /p1q instead (identical syntax).</p> |
| Command | /ppd - Protected Data |
| | <p>Adds, updates or 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).</p> |
| Syntax | <code>/ppd- [0x<hex data>][,<TVB dep.>]</code> |
| | <p> Deletes the PIO (-).</p> |
| Command | /psd - Secret Data |
| | <p>Adds, updates or deletes the PIO Secret Data of a Product Item.</p> |

| Command | /psd - Secret Data | |
|--|--|--|
| | <p>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).</p> | |
| Syntax | <p>Fills the Secret Data PIO with user-defined data. <code>/psd<ext. type>[<acc. code>][:0x<hex data>][,<TVB dep.>]</code> Fills the Secret Data PIO with <count> bytes of random data.. <code>/psd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>]</code> Removes the PIO (-) <code>/psd<ext. type>-</code></p> | |
|  | /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 | |
| | <p>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.</p> | |
| Syntax | /pt - /pt:<text>" |  Removes the PIO (-). |
|  | | |
| Command | /puc - Unit Counter | |
| | <p>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).</p> | |
| 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 | |
| | <p>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.</p> | |
| |  For a firmware version prior to 1.18 the maximum value is 16777215. | |
| | <p>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).</p> | |
| Syntax | /puca<value>[,<TVB dep.>] | |
|  | /puca25 | Sets the value of the Unit Counter PIO to the absolute value of 25. |

| | |
|----------------|---|
| Command | /pucr - Unit Counter, relative |
| | <p>Increments the Unit Counter of a Product Item by the specified amount. Expects a signed decimal value in the range [-2147483648, 2147483648] as argument for Firmware versions newer than 1.18..</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  For a firmware version prior to 1.18 the range is limited to [-16777215, 16777215]. </div> |
| 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 |
| | <p>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.</p> |
| Syntax | /pud- [0x<hex data>] |
| | <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Removes the PIO (-). </div> |
| e.g. | /pu- d0x112233445 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 |
| | <p>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.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Requires CodeMeter® Firmware Version 1.11 or higher. </div> |
| Syntax | /pup- [:<days>] [,<TVB dep.>] |
| e.g. | /pup:30 Removes the PIO (-). |
| Command | /pupa - Usage Period, absolute |
| | <p>Sets the length of the <i>Product Item's Usage Period</i> to the given number of <days>. Input as integer value greater than or equal to a value of null.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Requires CodeMeter® Firmware Version 1.11 or higher. </div> |
| Syntax | /pupa[:]<days>[,<TVB dep.>] |
| e.g. | /pupa:30 Removes the PIO (-). |
| Command | /pupr - Usage Period, rerelative |

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 | <code>/pupr[:]<days>[,<TVB dep.>]</code> |
|--------|--|

| | |
|------|--|
| e.g. | <code>/pupr:30</code> Removes the PIO (-). |
|------|--|

| Command | <code>/pwupidata - WUPI Data</code> | |
|---------|--|---|
| | 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. | |
| Syntax | <code>/pwupidata:[e<ext. type>][,b<block size>][,a<acc. code>]:<file></code> <code>/pwupidata:s<size>[,e<ext. type>][,b<block size>][,a<acc. code>][,f<fill byte>]</code> <code>/pwupidata[:s<size>[,e<ext. type>][,b<block size>]-</code> | Allocates WUPI Data storage, filled with the contents of file <file>. Allocates <size> bytes of WUPI Data storage, initialized with the value <fill byte>. Removes the WUPI Data storage. |

9.2.7 CmActLicense Options

This section describes operations related to the licensing system *CmActLicense*.

The following options are available:

| Command | <code>/lac - CmActLicense License Activation Code</code> |
|---------|--|
| | Use this option to calculate a <i>CmActLicense</i> activation code for activation by phone. Expects an installation identifier as argument. |
| Syntax | <code>/lac:<installation ID></code> |
| Command | <code>/laf - CmActLicense License Activation File</code> |
| | Use this option to set the path to the <i>CmActLicense</i> license activation request file and the license activation file. |
| Syntax | <code>/laf:<request file>\<activation file>\<</code> |
| Command | <code>/lbind - CmActLicense Binding Value</code> |
| | 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 <code>0x</code> as argument. |
| | This option is only supported in combination with the binding mode <u>/lfs:cus</u> ³⁴⁶ and <u>/laf</u> |

| | |
|----------------|--|
| Command | /lbind – CmActLicense Binding Value |
| Syntax |  ³⁴⁴ . The request file argument of option /laf must be omitted. |
| Command | /ldf – Display of CmActLicense License File |
| Syntax | /ldf:<file> |
| Command | /ldi – Display of CmActLicense-Installation ID |
| Syntax | /ldi:<installation ID> |
| Command | /lfs – CmActLicense Binding Scheme (License Feature Set) |
| | Use this option to set the <i>CmActLicense</i> Binding Schemes (License Feature Set). |
| | CodeMeter® SmartBind |
| | <i>CodeMeter® SmartBind</i> 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 <i>CodeMeter® SmartBind</i> 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, <i>CodeMeter® SmartBind</i> uses the tolerance level 2. If you like to change this setting please contact Wibu-Systems Support before you do so. |
| | also supports Binding Schemes which relate to fix or configurable hardware properties of the PC. |
| |  Wibu-Systems recommends to contact Wibu-Systems Support before you do so. |
| | Syntax: <code>/lfs:smart[:<tolerance level>]</code> |
| | CmActLicense with SmartBind for licenses in a VM (Virtual Machine) |
| | The behavior of <i>CmActLicense</i> with the binding scheme <i>SmartBind</i> for licenses in a VM is defined as follows: |
| | <ul style="list-style-type: none"> • If the VM is copied. i.e. the "I copied it" option has been selected, the license breaks. • If the VM is moved, i.e. the "I moved it" option has been selected, then the license remains intact in case of the same CPU types. <p>However, if the CPU types differs, the license also breaks except the tolerance level has been set to a value of "3" (loose).</p> |
| | 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> to define how restrictive the scheme is to be, i.e. how many properties need to remain unchanged. |

| Command | /lfs – CmActLicense Binding Scheme (License Feature Set) | |
|---|--|-------------|
| | Hardware Property | Description |
| 'b' | (B)IOS | |
| 'c' | (C)PU | |
| 'd' | (d)isk | |
| 'n' | (n)etwork adapter | |
| Syntax: | <code>/lfs:[b][c][d][n][:<count>]</code> | |
| Configurable Hardware Properties | Use one of the following other configurable hardware properties which cannot be combined. | |
| Binding Scheme | Description | |
| 'ip' | (IP) Address | |
| 'mid' | (m)achine (ID); includes the machine SID and the domain SID. | |
| 'non' | (non)e – no hardware binding | |
| 'ran' | (ran)dom | |
| 'ser' | Product- (Ser)ial number | |
| 'cus' | (cus)tom plugin; expects the plugin's name (up to 31 characters) as argument, valid characters are 'A'..'Z', 'a'..'z', '0'..'9', '_' | |
| Syntax: | <code>/lfs:ip mid non ran ser cus:<plugin name></code> | |
| Command | /lif – CmActLicense License Information File (file-based activation) | |
| | Use this option to set the path to the <i>CmActLicense</i> license information file. | |
| Syntax | <code>/lif:<license information file></code> | |
| Command | /lip – CmActLicense License Information File (activation by phone) | |
| | Use this option to set the path to the <i>CmActLicense</i> license information file. | |
| Syntax | <code>/lip:<license information file></code> | |
| Command | /lmrt – Minimum required CodeMeter Runtime version | |
| | Use this option to specify the minimum CodeMeter Runtime version that is required for using <i>CmActLicense</i> . As argument a version number is expected, e.g. '4.50'. The most recent version supported by <i>CmActLicense</i> is 4.30. | |
| Syntax | <code>/lmrt:<version>.<subversion></code> | |
| Command | /lopt – CmActLicense License Options | |
| | Use this option to specify <i>CmActLicense</i> license options. | |
| | Valid license option identifiers are: | |
| Flag | Description | |
| 'vm' | <i>CmActLicense</i> license can be used on a (V)irtual (M)achine | |

| Command | /lopt – CmActLicense License Options | | | | | | | | | | | | | | | | | |
|-------------------------|---|--------------------|-------------------------|--------------------|-----------|---|---------|------------|------------------|---------------------|------------|---------------|------------------|---------------------|--------|-----------|--------|-----------|
| | Flag | Description | | | | | | | | | | | | | | | | |
| Syntax | <code>'reimport'</code> CmActLicense activation file can be reimported any time. | | | | | | | | | | | | | | | | | |
| Command | /los – CmActLicense License Target Operating System | | | | | | | | | | | | | | | | | |
| | Use this option to specify on which operating system(s) the CmActLicense license can be used. | | | | | | | | | | | | | | | | | |
| |  This option is mandatory. | | | | | | | | | | | | | | | | | |
| | The following operating systems are supported: | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Operating System</th><th>Description</th></tr> </thead> <tbody> <tr> <td>'Win'</td><td>(Win)dows, all supported Windows versions, Windows 2000 or higher</td></tr> <tr> <td>'Mac'</td><td>(Mac) OS X</td></tr> <tr> <td>'Lin'</td><td>(Lin)ux</td></tr> <tr> <td>'Emb'</td><td>(Emb)eeded</td></tr> </tbody> </table> | | Operating System | Description | 'Win' | (Win)dows, all supported Windows versions, Windows 2000 or higher | 'Mac' | (Mac) OS X | 'Lin' | (Lin)ux | 'Emb' | (Emb)eeded | | | | | | |
| Operating System | Description | | | | | | | | | | | | | | | | | |
| 'Win' | (Win)dows, all supported Windows versions, Windows 2000 or higher | | | | | | | | | | | | | | | | | |
| 'Mac' | (Mac) OS X | | | | | | | | | | | | | | | | | |
| 'Lin' | (Lin)ux | | | | | | | | | | | | | | | | | |
| 'Emb' | (Emb)eeded | | | | | | | | | | | | | | | | | |
| | It is also possible to select specific OS versions: | | | | | | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Operating System</th><th>Description</th></tr> </thead> <tbody> <tr> <td>'Win2000'</td><td>Windows 2000</td></tr> <tr> <td>'WinXp'</td><td>Windows XP</td></tr> <tr> <td>'WinServer-2003'</td><td>Windows Server 2003</td></tr> <tr> <td>'WinVista'</td><td>Windows Vista</td></tr> <tr> <td>'WinServer-2008'</td><td>Windows Server 2008</td></tr> <tr> <td>'Win7'</td><td>Windows 7</td></tr> <tr> <td>'Win8'</td><td>Windows 8</td></tr> </tbody> </table> | | Operating System | Description | 'Win2000' | Windows 2000 | 'WinXp' | Windows XP | 'WinServer-2003' | Windows Server 2003 | 'WinVista' | Windows Vista | 'WinServer-2008' | Windows Server 2008 | 'Win7' | Windows 7 | 'Win8' | Windows 8 |
| Operating System | Description | | | | | | | | | | | | | | | | | |
| 'Win2000' | Windows 2000 | | | | | | | | | | | | | | | | | |
| 'WinXp' | Windows XP | | | | | | | | | | | | | | | | | |
| 'WinServer-2003' | Windows Server 2003 | | | | | | | | | | | | | | | | | |
| 'WinVista' | Windows Vista | | | | | | | | | | | | | | | | | |
| 'WinServer-2008' | Windows Server 2008 | | | | | | | | | | | | | | | | | |
| 'Win7' | Windows 7 | | | | | | | | | | | | | | | | | |
| 'Win8' | Windows 8 | | | | | | | | | | | | | | | | | |
| Syntax: | <code>/los:<OS version>[,<OS version>]</code> | | | | | | | | | | | | | | | | | |
| Command | /lpid – CmActLicense ID (CmAct ID) | | | | | | | | | | | | | | | | | |
| | Use this option to set the CmActLicense ID (CmAct ID). As argument a combination of four ASCII characters that represent the major part of the ID is expected. Additionally an unsigned number that represents the ID's minor part can be specified. | | | | | | | | | | | | | | | | | |
| Syntax | <code>/lpid:<major>[-<minor>]</code> | | | | | | | | | | | | | | | | | |
| | <pre>major = 'ABCD'; minor = 123 major = 'ABCD'; minor part omitted => minor = 0 /lpid:ABCD-123</pre> | | | | | | | | | | | | | | | | | |
| Command | /lpn – CmActLicense Name (CmAct Name) | | | | | | | | | | | | | | | | | |
| | This option sets the CmActLicense name (CmActLicense Name)which is displayed as name in <i>CodeMeter Control Center</i> . | | | | | | | | | | | | | | | | | |
| Syntax | <code>/lpn: "<name>"</code> | | | | | | | | | | | | | | | | | |

Programming examples

How do I program and update a *CmActLicense* license?

Creating and activating a PC-bound *CmActLicense* covers the following single steps:

- Creating a binding scheme
A binding scheme defines the hardware characteristics of a PC used for the binding. Here *CodeMeter* provided *SmartBind*[®] as an easy and at the same time secure way to uniquely bind licenses to a PC.
- Importing an empty 'virtual' *CmContainer* by the end user.
- 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 automated in *CodeMeter License Central*. The following description refers to using *CmBoxPgm*.

1. Creating the license information file by the ISV

The ISV creates an empty license container (LIF) (*.wbb) using the binding scheme *SmartBind*²⁸.

- a) Specify the Firm Code (in this case 5010, the evaluation code for *CmActLicense*).
- b) Specify a Product Code 14.
- c) Specify the file name of the license information file (in our example below "TemplateDisc.wbb").
- d) Specify the product name of the license information file (in our example below "Empty virtual CmContainer"). This name will display in *CodeMeter Control Center* instead of the name of the *CmContainer*.
- e) Specify a Product-ID (in our example 0001). This allows you to identify the matching license information file with *CodeMeter Core API*.
- f) Specify the binding scheme *SmartBind* with a tolerance level 2.
- g) Specify for which operating systems this license information file is to be used. You can specify either operating system families (Linux, Mac OS and Windows), or single special versions (e.g. Windows XP). In our example, we create a license information file to be used under Windows.

The respective *CmBoxPgm* command looks as follows:

```
CmBoxPgm /f5010 /p14 /ca /lif:"TemplateDisc.wbb" /lpn:"Empty virtual
CmContainer" /lpid:0001 /lfs:smart:2 /los:win
```

- h) The file "TemplateDisc.wbb" is now delivered to the end user.

2. Creating a license request file by the end user

- a) The end user imports the file by drag and dropping it onto *CodeMeter Control Center*. Alternatively, the license information file is imported via the "**File | Import license**" and the license is displayed in *CodeMeter Control Center*.
- b) The end user creates a license request file by selecting the virtual *CmContainer* to be activated and clicking the button "**Activate license**".

- c) The license request file created by *CmFAS Assistant* the end user sends to the ISV.
- 3.** Programming the license update file by the ISV
- From the license request file you received, you can now create the license update file which will activate the license on your customer's PC.
- For the most part, the *CmBoxPgm* parameters and options are identical to the ones used to produce the license information file. Some parameters must be identical. They are: Firm Code (/f...), Product-ID (/lpid:...), binding scheme (/lfs:...), and operating systems (/los:...).
- Firm Item Text (/ft:...) and the name of the virtual *CmContainer* (/lpn:...) may be changed. With /laf:... you specify the license request file, and separated by comma, the name of the license update file you wish to create.
- You may add or change any license entries and their respective properties. In the example for Product Code 14 only the Product Item Text has been supplemented, and License Quantity has been set to 1. License Quantity stands for the number of simultaneous use of licenses, especially as floating license in the network. Here is the syntax for the command line.
- The respective *CmBoxPgm* command looks as follows:
- ```
CmBoxPgm /f5010 /ft:"CodeMeterAct Demo Firm Item" /Cu /p14 /pt:"My First
CmActLicense license" /plql /ca /laf:"[Name of the license request file].Wi-
buCmRaC", "[Name of the license update file].WibuCmRaU" /lpn:"My first license" /
lpid:0001 /lfs:smart:2 /los:win
```
- However, you might also add, for example, a time limit of 30 days from initial start (/pup:30) or an Unit Counter of 10 units for decrementing (/puca10). Please optionally insert both before the /ca option.
- The respective *CmBoxPgm* command looks as follows:
- ```
CmBoxPgm /f5010 /ft:"CodeMeterAct Demo Firm Item" /Cu /p14 /pt:"My First
CmActLicense license" /pup:30 /puca30 /plql /ca /laf:"[Name of the license re-
quest file].WibuCmRaC", "[Name of the license update file].WibuCmRaU" /lpn:"My
first license" /lpid:0001 /lfs:smart:2 /los:win
```
- The created license update file is delivered to the end user.
- 4.** Importing and activating the license by the end user
- a)** The end user activates the license by selecting the virtual *CmContainer* to be activated and clicking the button "**Activate license**".
 - b)** The end user optionally sends a receipt to the ISV.
- 5.** Updating of own *CmActLicense* licenses
- Updating (i.e. adding, editing and deleting license entries or their properties) and reactivation (i.e. on changed binding information) takes place the same way the license was activated the first time. The customer creates a license request file. You create a related license update file. The customer imports this file, and optionally sends you back a receipt. The command line options are analogous to the first activation of the license. Optionally, you are able to add properties or license entries (other Product Codes).
-  Please note that on a license update, you cannot create the license update file on the PC to which the license is to be bound. However, this restriction is relevant only while testing because in practice you will create the license for your customer and not for yourself.

How do I program a Trial License?

In order to create a *CmActLicense* [Trial License](#)²⁹, 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 /F5010 /ft:"MyCompany" /cau /p2002 /pup90 /ca /laf:"UpdateTrialLicense.WibuCmRaU" /lpn:"Trial CmActLicense" /lfs:none /los:WIN /lpid:0001
```

Description

A *CmActLicense* license container with a F(irm Code) 5010 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) "Upda-teTrialLicense.WibuCmRaU" with the *CmActLicense* name (/lpn) "Trial CmActLicense" covering a binding scheme (/lfs) "None" for WIN(dows) operating systems (/los) and a *CmActLicenseID* (/lpid) of 0001.

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.

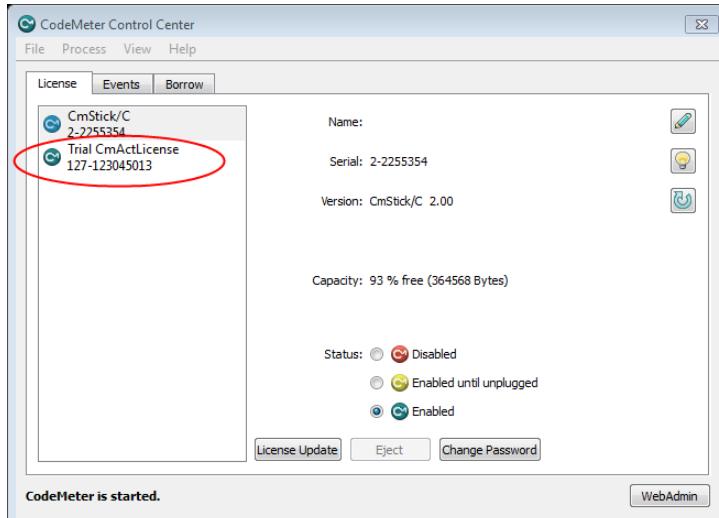
A Trial License cannot be updated and imported only once, i.e. the option /reimport is not allowed to be set.

Output

A message like the one below is returned indicating the successful creation and the license activation on file *UpdateTrialLicense.WibuCmRaU* is created in the user directory (%Users%).

```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 127-123045006, FC=5010
*** Add Product Item, CmContainer 127-123045006, 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](#)²⁹ 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 /F5010 /ft:"MyCompany" /cau /p2002 /ca /laf:"Update-ProtectionOnly-License.WibuCmRaU" /lpn:"Protection Only CmActLicense" /lfs:none /los:WIN /lpid:0001
```

Description

A *CmActLicense* license container with a *F*(irm Code) 5010 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" for *WIN*(dows) operating systems (/los) and a *CmActLicenseID* (/lpid) of 0001.

Optionally, you may also allow the use on virtual machines (/lopt:vm) and any number of file re-import (/reimport).

Output

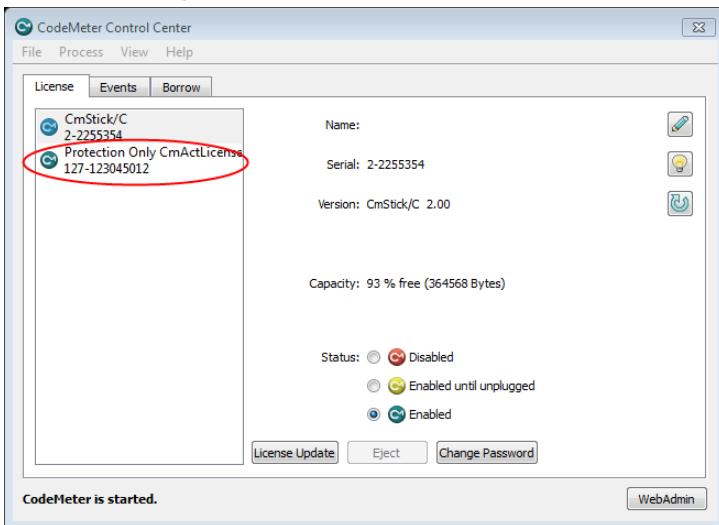
A message like the one below is returned indicating the successful creation and the license activation file *Update-ProtectionOnly-License.WibuCmRaU* is created in the user directory (%Users%\)..

```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 127-123045009, FC=5010
*** Add Product Item, CmContainer 127-123045009, FC=5010, 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 License Borrowing Options

This section describes available options referring to License Borrowing entries. These options are in addition to the Product Item Options and to be used equivalently as a part of a Product Item command.
Borrowing-related commands are structured in the following way:

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

The following options are available.

| Command | /bls - License Borrowing Server | | | | | | | | | | |
|--|---|----------|--|--|--|---------|---------------------------------|------|--|------|------------------------------------|
| | Use this option to add, update or remove a borrow license client entry. | | | | | | | | | | |
| Syntax | <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">i</td> <td style="padding: 5px;">Requires CodeMeter® firmware 1.10 or higher.</td> </tr> <tr> <td style="padding: 5px;">/bls[:<cm ca>,<fc>,<pc>,<fm>,<lqClient>,<duration> [,serverID]]</td> <td style="padding: 5px;">Licensing system of the client license (<i>CmDongle</i>, <i>CmActLicense</i>)</td> </tr> <tr> <td style="padding: 5px;"><cm ca></td> <td style="padding: 5px;">Firm Code of the client license</td> </tr> <tr> <td style="padding: 5px;"><fc></td> <td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;"><pc></td> <td style="padding: 5px;">Product Code of the client license</td> </tr> </table> | i | Requires CodeMeter® firmware 1.10 or higher. | /bls[:<cm ca>,<fc>,<pc>,<fm>,<lqClient>,<duration> [,serverID]] | Licensing system of the client license (<i>CmDongle</i> , <i>CmActLicense</i>) | <cm ca> | Firm Code of the client license | <fc> | | <pc> | Product Code of the client license |
| i | Requires CodeMeter® firmware 1.10 or higher. | | | | | | | | | | |
| /bls[:<cm ca>,<fc>,<pc>,<fm>,<lqClient>,<duration> [,serverID]] | Licensing system of the client license (<i>CmDongle</i> , <i>CmActLicense</i>) | | | | | | | | | | |
| <cm ca> | Firm Code of the client license | | | | | | | | | | |
| <fc> | | | | | | | | | | | |
| <pc> | Product Code of the client license | | | | | | | | | | |

| Command | /bls - License Borrowing Server | | | | | | | | |
|--------------|--|------|-----------------------------------|------------|---|------------|--|--------------|---|
| | <table border="1"> <tr> <td><fm></td><td>Feature Map of the client license</td></tr> <tr> <td><lqClient></td><td>Number of the maximum borrowable client licenses (License Quantity)</td></tr> <tr> <td><duration></td><td>Period how long at most the license can be borrowed (in minutes)</td></tr> <tr> <td>[,serverID]</td><td>ServerID of the server This ID has a length of 8 bytes and should follow the notation: 0x<hex data>.</td></tr> </table> <p> Up to a maximum of 4 of such entries for a single server license can be programmed.</p> | <fm> | Feature Map of the client license | <lqClient> | Number of the maximum borrowable client licenses (License Quantity) | <duration> | Period how long at most the license can be borrowed (in minutes) | [,serverID] | ServerID of the server This ID has a length of 8 bytes and should follow the notation: 0x<hex data>. |
| <fm> | Feature Map of the client license | | | | | | | | |
| <lqClient> | Number of the maximum borrowable client licenses (License Quantity) | | | | | | | | |
| <duration> | Period how long at most the license can be borrowed (in minutes) | | | | | | | | |
| [,serverID] | ServerID of the server This ID has a length of 8 bytes and should follow the notation: 0x<hex data>. | | | | | | | | |
| Command | /blc - License Borrowing Client | | | | | | | | |
| | <p>Use this option to add, update or remove a borrow license server entry.</p> <p> Requires <i>CodeMeter®</i> firmware 1.10 or higher.</p> <p>As arguments a maximum of four parameter blocks separated by a colon may be specified. Each parameter block consists of the identifier of the client license's licensing system, its Firm Code, its Product Code, and Feature Map, the number of licenses that can be checked out, the maximum check-out period measured in minutes and a server ID are expected to follow.</p> | | | | | | | | |

| | |
|--------|--|
| Syntax | /blc:[cm ca],<fc>,<pc> [,serverID] [cm ca] Licensing system of the server license (<i>CmDongle</i> , <i>CmActLicense</i>) <fc> Firm Code of the server license <pc> Product Code of the server license [,serverID] ServerID of the server This ID has a length of 8 bytes and should follow the notation: 0x<hex data>. |
|--------|--|

Programming example

A software vendor offers his customer to provide mobile licenses as part of his 50 licenses, i.e. licensed application also run when not connected to the license server. This is implemented using the license borrowing feature. Altogether 10 licenses are preprogrammed for mobile use. 5 of the licenses are issued for *CmDongle* not bound to specific PCs, while the remaining 5 licenses are bound to specific laptops using *CmActLicense*.

1. First of all, a server *CmDongle* has to be preprogrammed.

For the protected application (Firm Code 10 and Product Code 1000) 10 licenses (/plq) are prepared for borrowing. Then using the command /bls the borrowable server licenses are preprogrammed. Here 5 *CmDongle* and 5 *CmActLicense* licenses are defined as borrowable for 8 hours (480). For *CmDongles* with the license 10:1000, for *CmActLicense* license 5010:1000.

Command:

```
CmBoxPgm.exe /qb1 /f10 /ft:"License Borrow-Server" /cau /p1000 /pt:" License Borrowing with CmDongle and CmActLicense" /plq10 /bls:cm,10,1000,0,5,480,0x12345678:ca,5010,1000,0,5,28800,0x12345678 /ca
```

2. Then 5 clients for *CmDongle* are prepared.

For each of the 5 *CmDongles* the command /blc provides for the borrowable client license of the protected application with the Firm Code 10 and Product Code 1000.

Command:

```
CmBoxPgm.exe -qb1 -f10 -ca -p1000 -blc:cm,10,1000,0x12345678 -pt:"License
```

Borrowing Client CmDongle" -ca

3. In the case of the software-based variant *CmActLicense* you first have to program the preactivated license information files (LIFs).

a) In the license information file *CmAct.wbb* you define the required binding scheme (here, for example, a binding to the hard disk serial number

`[/lfs:D:1]` and the target operating system `[/los:win]`.

This file you will send to your customer.

Command:

```
CmBoxPgm.exe /f5010 /p1000 /plq5 /ca /lif:"CmAct.wbb" /lpn:"Pre-activated CmActLicense" /lpid:0001 /lfs:D:1 /los:win
```

b) Then your customer will drag & drop the file *CmAct.wbb* into *CodeMeter Control Center* and will create a license request file (here the file "LicenseRequest.WibuCmRaC") using the *CmFAS Assistant*. This file will be sent to you.

In the next step you program a license update file (here the file "Activation.WibuCmRaU") which your customer then import using the *CmFAS Assistant*.

At the same time using the command `/blc` the protected application with the Firm Code 5010 and Product Code 1000 will be made available as borrowable client license.

Command:

```
CmBoxPgm.exe -f5010 -p1000 -blc:cm,10,1000,0x12345678 -pt:"CmAct Borrowing License Client" -ca /laf:"LicenseRequest.WibuCmRaC","Activation.WibuCmRaU" /lpn:"Borrow License" /lpid:0001 /lfs:D:1 /los:win
```

Borrowing and returning of a license itself is done via the "["Borrow"](#)" tab in ["CodeMeter Control Center"](#)  ⁴²¹. In ["CodeMeter WebAdmin"](#)  ⁴³⁵ the license allocation is ["displayed"](#)  ⁴⁶⁰ number and maximum borrowing time can be configured.

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 |
|---------|---|
| | <p>This option initiates a FSB command sequence. Expects the Firm Code the FSB entry refers to as argument.</p> |
| Syntax | <code>/fsb<Firm Code></code> |
| Command | /fk - Firm Key |
| | <p>Use this option to specify a new Firm Key (32 Bytes).</p> <p>Please act with extreme caution when using this option! When changing the existing Firm Key you deeply interfere in encryption and programming processes!</p> <p> Because then all future encryption operations and all <i>CmContainer</i> programming will refer to this "new" Firm Key ! "New" encrypted applications will not run with "old" programmed <i>CmContainer</i>!</p> |

| Command | /fk - Firm Key |
|---------|--|
| | 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 Enabling Options

This section describes the Enabling-related operations. Basic operations are the creation, modification or deletion of Enabling Blocks.

Enabling build up as follows:

```
/f<Firm Code> [...] /e[<index>][:<type>] [<Enabling Options>] <Main Command>
```

The following options are available:

| Command | /e - Enabling | | | | | | | | | | | | |
|--------------------------------|---|-------------|------------------------------------|----------------|---------------------------------------|----------------|---------------------------------------|--------------------------|---|--------------------------------|--|--------|--|
| | This option defines the Enabling Block to be programmed. Supported arguments are the Enabling Block's index and type. Actually, the types Simple PIN (sp) and Time PIN (tp) are supported. | | | | | | | | | | | | |
| Syntax | /e[<index>] : [sp tp] | | | | | | | | | | | | |
| Command | /eac - Access Code | | | | | | | | | | | | |
| | Use this option to specify or change the Access Codes of an Enabling Block. The data may be given as password string enclosed in double quote characters or as hexadecimal data, at least 2 bytes, 16 bytes at maximum. | | | | | | | | | | | | |
| |  If the Firm Key will serve as Access Code the token 'fk' has to be specified instead. | | | | | | | | | | | | |
| Syntax | /eac:<access code>[=<new accesscode>] /eac:<access code> = "<text>" /eac:<access code> = 0x<hex data> /eac:<access code> = fk | | | | | | | | | | | | |
| Command | /eatt - Attach Enabling Block | | | | | | | | | | | | |
| | Use this option to attach an Enabling Blocks to a Firm Item or Product Item. | | | | | | | | | | | | |
| Syntax | /eatt<Firm Code>[,<Product Code>[,<Feature Code> <product item reference>]]:<Enable Level>,<Disable Level>[:req+ -] <table border="0"> <tr> <td><Firm Code></td> <td>Refers to the Firm Code of a block</td> </tr> <tr> <td><Product Code></td> <td>Refers to the Product Code of a block</td> </tr> <tr> <td><Feature Code></td> <td>Refers to the Feature Code of a block</td> </tr> <tr> <td><product item reference></td> <td>When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l).</td> </tr> <tr> <td><Enable Level>,<Disable Level></td> <td>Valid levels comprise: locate (loc), read (read), encrypt (enc), unituse (uu), or modify (mod).</td> </tr> <tr> <td>req+ -</td> <td>Setting the required flag serves to avoid potential conflicts when activating or deactivating. Especially when several attachment targets exist.</td> </tr> </table> | <Firm Code> | Refers to the Firm Code of a block | <Product Code> | Refers to the Product Code of a block | <Feature Code> | Refers to the Feature Code of a block | <product item reference> | When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l). | <Enable Level>,<Disable Level> | Valid levels comprise: locate (loc), read (read), encrypt (enc), unituse (uu), or modify (mod). | req+ - | Setting the required flag serves to avoid potential conflicts when activating or deactivating. Especially when several attachment targets exist. |
| <Firm Code> | Refers to the Firm Code of a block | | | | | | | | | | | | |
| <Product Code> | Refers to the Product Code of a block | | | | | | | | | | | | |
| <Feature Code> | Refers to the Feature Code of a block | | | | | | | | | | | | |
| <product item reference> | When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l). | | | | | | | | | | | | |
| <Enable Level>,<Disable Level> | Valid levels comprise: locate (loc), read (read), encrypt (enc), unituse (uu), or modify (mod). | | | | | | | | | | | | |
| req+ - | Setting the required flag serves to avoid potential conflicts when activating or deactivating. Especially when several attachment targets exist. | | | | | | | | | | | | |

| | | | | | | | | | |
|--|--|-------------|--------------------------------------|----------------|---|----------------|---|--------------------------|---|
| Command | /eatt - Attach Enabling Block | | | | | | | | |
| | <p>If at least one required flag set in the case of several attachment targets, a logical AND conjunction defines that all settings with a required flag must match before a defined operation is performed for a complete <i>CmContainer</i>, an Item level, or a license entry.</p> <p>This is the default setting since Firmware Version 1.18. On attaching  an Enabling Blocks using a lookup table entry the Required Flag is set by default.</p> <p>Although you are able to set the flag to non required, this has no effect since in the default setting a NonRequired Flags is ignored by a logical OR conjunction if at least one required flag exists. This due to the global enabling valid for the complete <i>CmContainer</i>. If you wish to change this global enabling, please contact Wibu-Systems Support.</p> | | | | | | | | |
| Command | /edet - Detach Enabling Block | | | | | | | | |
| | <p>Use this option to detach an Enabling Blocks from a Firm Item or Product Item.</p> | | | | | | | | |
| Syntax | <pre>/edet<Firm Code>[,<Product Code>[,<Feature Code> <product item reference>]]</pre> <table> <tr> <td><Firm Code></td><td>Refers to the Firm Code of the block</td></tr> <tr> <td><Product Code></td><td>Refers to the Product Code of the block</td></tr> <tr> <td><Feature Code></td><td>Refers to the Feature Code of the block</td></tr> <tr> <td><product item reference></td><td>When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l).</td></tr> </table> | <Firm Code> | Refers to the Firm Code of the block | <Product Code> | Refers to the Product Code of the block | <Feature Code> | Refers to the Feature Code of the block | <product item reference> | When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l). |
| <Firm Code> | Refers to the Firm Code of the block | | | | | | | | |
| <Product Code> | Refers to the Product Code of the block | | | | | | | | |
| <Feature Code> | Refers to the Feature Code of the block | | | | | | | | |
| <product item reference> | When using the Product Item Reference you obtain an exact addressing when Product Codes occur several times. For example, the Product Item Reference is issued when using the option list (/l). | | | | | | | | |
| Command | /edt - Disable Time | | | | | | | | |
| | <p>Use this option to specify a Disable Time of an Enabling Block.</p> <p> This feature is not support Enabling Block of the type Simple PIN.</p> | | | | | | | | |
| Syntax | /edt | | | | | | | | |
| Command | /edta - Disable Time, absolute | | | | | | | | |
| | <p>Either sets a Disable Time of an Enabling Block or suspends the Disable Time. Specify a date followed by an optional time and time zone argument. If the time zone is omitted, the system's time zone will be used instead.</p> <p> Using the parameter none suspends the Disable Time.</p> | | | | | | | | |
| Syntax | <pre>/edta<YYYY><Month><DD>[,<SS>:<MM>:<SS>[PST MST CST EST UTC CET EET]] or according to ISO-8601: /edta<YYYY>-<MM>-<DD>[T<hh>:<mm>:<ss>[Z] [±hh:mm or ±hhmm] [±hh] /edta:none</pre> | | | | | | | | |
|  | <p>Sets the Disable Time to December 31st, 2012, 1 second to midnight, UTC</p> <p>/edta2011Dec31,23:59:59UTC /edta2011-12-31T23:59:59Z</p> | | | | | | | | |
| | <p>Suspends the Disable Time</p> <p>/edta:none</p> | | | | | | | | |

| | |
|----------------|---|
| Command | /edtr - Disable Time, relative |
| | <p>Adds the specified number of days to the current value of the Disable Time of the Enabling Block. Expects an integer value greater than or equal to zero as argument.</p> <p> If the Enabling Block doesn't exist yet, the current system time plus the specified offset will be set as Disable Time. For example: /edtr1 corresponds 1 day immediately starting.</p> |
| Syntax | /edtr<number of days> |
| Command | /em - Enabling Mode |
| | <p>Use this option to set the activation status (mode) of the Enabling Block. Valid mode arguments are (d)isabled or (e)nabled.</p> <p> Additionally, (t)emporary enabling can be switched on (+) or off (-).</p> <p> For a firmware version prior to 1.18 the use of temporary enabling is limited to the Implicit Firm Item. When the Enabling Block locates in the Implicit Firm Item (IFI), additionally, the activation can be temporarily (t) activated (+) or deactivated (-).</p> |
| Syntax | /em: [d e][,][t+ -] |
| Command | /et - Text |
| | <p>Use this option to set the Text of an Enabling Block. Accepts a character string (up to 256 characters) enclosed in double quote characters as argument.</p> <p> This feature is not supported by Simple PIN Enabling Blocks.</p> |
| Syntax | /et: "<text>" |

9.2.11 Special Commands

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

| | |
|----------------|--|
| Command | /bkp - Backup File |
| | <p>Enables the backup file viewer mode. expects a <i>CodeMeter®</i> backup file.</p> <p> Only allowed in combination with list option /1.</p> |
| Syntax | /bkp:\ "<Backup Datei>\" |
| Command | /crac - Create Remote Activation Context File (*.WibuCmRaC) |
| | <p>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 <i>CmContainer</i> 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.</p> |

| | | | |
|--|--|--|---|
| Command | /crac - Create Remote Activation Context File (*.WibuCmRaC) | | |
| |  This option cannot be used in the Remote Activation mode. | | |
| Syntax | <code>/crac[:<*.WibuCmRaC file> <*.WibuCmRaC target directory>]</code> | | |
| Command | /ra - Remote Activation | | |
| | Calculates the Remote Programming sequences for the specified operations. The contents of the target <i>CmContainer</i> will be read from the specified Remote Activation Context file (=*.WibuCmRaC) or Remote Activation Modified Context file (=*.WibuCmRaC). The generated programming sequences will be written into a Remote Activation Update file (=*.Wi-buCmRaU). | | |
| Syntax | <code>/ra:<*.WibuCmRaC file> <*.WibuCmRaM file>[,<*.WibuCmRaU file>[,<*.WibuRaM file>]]</code> | | |
| | <table border="1"> <tr> <td><code>/ra:MyCmContainer</code> <code>/ra:Context-WibuCmRaC,Context-WibuCmRaM,Context-WibuCmRaU</code></td> <td>Establishing and writing the specified programming sequence</td> </tr> </table> | <code>/ra:MyCmContainer</code> <code>/ra:Context-WibuCmRaC,Context-WibuCmRaM,Context-WibuCmRaU</code> | Establishing and writing the specified programming sequence |
| <code>/ra:MyCmContainer</code> <code>/ra:Context-WibuCmRaC,Context-WibuCmRaM,Context-WibuCmRaU</code> | Establishing and writing the specified programming sequence | | |
| Command | /rcl - Cleanup Registry (<i>CmDongle</i> only) | | |
| | Deletes the CodeMeter® related Windows registry entries of the chosen categories. The supported <i>CmDongle</i> form factors comprise: <i>CmStick</i> (c), <i>CmStick/M</i> (m), <i>Removable Media</i> (r). | | |
| |  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. | | |
| Syntax | <code>/rcl[:cmr]</code> | | |
| Command | /sqd - Sequence Dump | | |
| | Dumps the generated programming sequences. | | |
| Syntax | <code>/sqd</code> | | |
| Command | /log - Logging | | |
| | Enables logging. Expects the path to a log file as argument. | | |
| |  If the log file is not specified a default name is used. 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 | <code>/log "[logfile]"[+]</code> | | |
| Command | /? - Help | | |
| | Issues further help on desired topics. | | |
| |  If no topics are specified, the complete help list is issued. | | |
| Syntax | <code>/?[<topic> <option>]</code> | | |

| | |
|----------------|---|
| Command | /v - Verbose Mode |
| | Activates the detailed display mode. |
| Syntax | /v |
| Command | /val - Validation Mode |
| | <p>Activates the validation mode. In this mode a <i>CmContainer</i> returns a confirmation sequence after each successful programming operation. The received data is validated with the Firm Security Box.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  By default, this mode is deactivated increasing the performance. Using this option reactivates the validation routine. </div> |
| Syntax | /val |

Programming examples

```
CmBoxPgm /qs1-1234 /f206 /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 206 in *CmContainer* 1-1234.

```
CmBoxPgm /qs1-1234 /f206 /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 /f206 /p2001 /pet /cd
```

Deleting the Expiration Time of the Product Item 2001.

```
CmBoxPgm /ra:1-1234.WibuCmRaC,1-1234.WibuCmRaU,1-1234.WibuCmRaM/f206 /p2008 /ca
```

Adding the Product Item 2008 per remote programming. Besides the remote update file, also a modified remote context file is created allowing for later reprogramming.

```
CmBoxPgm /qs1-1234 /f206 /p2001 /plq5 /ca
```

Setting the License Quantity to 5 for Product Item 2001 on Firm Item 206.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.

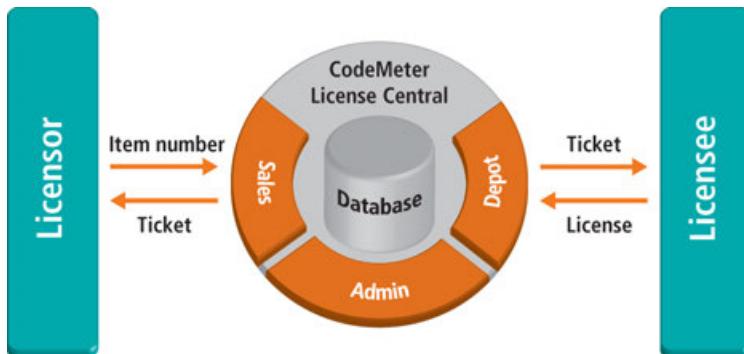


Figure 186: License Collection by Licensee

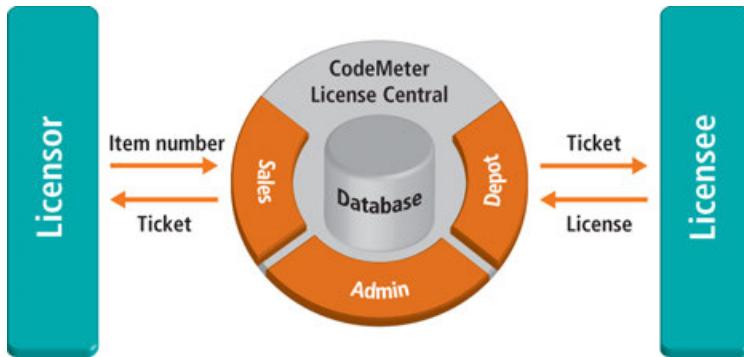


Figure 187: License Collection by Licensor

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*.

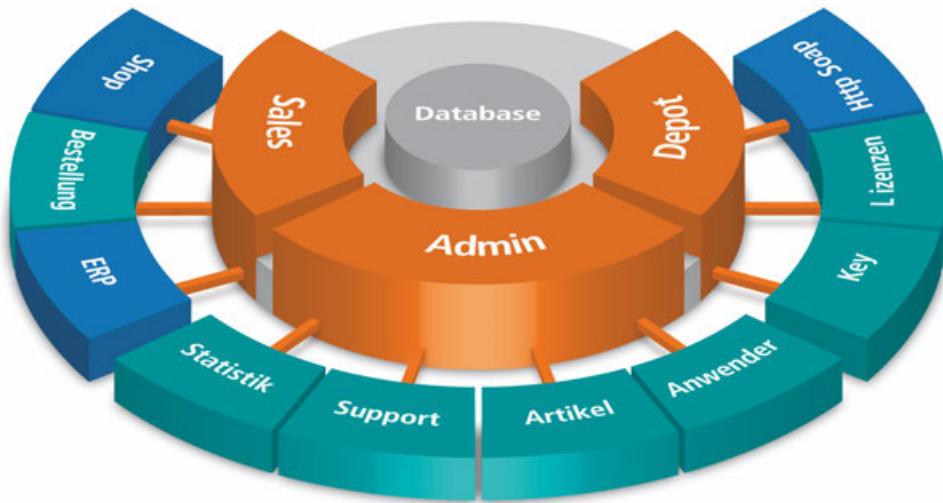


Figure 188: *CodeMeter License Central - Interfaces and Functions*

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.

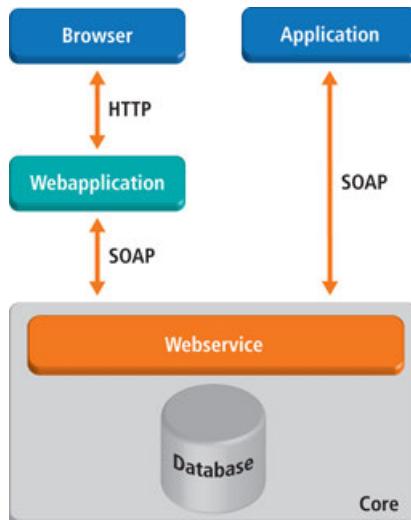


Figure 189: SOAP Access to *CodeMeter License Central*

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 <i>CodeMeter License Central</i> 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 <i>CodeMeter License Central</i> . The advantage in this case: all required components are already installed, and database management is not required. |
| Small Network/ Intranet | Here the <i>Desktop Edition</i> of <i>CodeMeter License Central</i> is installed on a server and the staff is able to access <i>CodeMeter License Central</i> using a browser. The advantage in this case: all staff is accessing a central database. |
| Online Accessibility | Here the <i>Edition</i> Edition of <i>CodeMeter License Central</i> 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 <i>Edition</i> Edition of <i>CodeMeter License Central</i> is installed on a server. Via a Connector located in the DMZ the Online Shop and <i>CodeMeter License Central</i> 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 <i>Internet</i> or <i>Edition</i> of <i>CodeMeter License Central</i> is installed on a server. The ERP/CRM system calls an internal Connector which processes the data and forwards it to <i>CodeMeter License Central</i> . 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, i.e. the *.WibuCmRaC file (License Request File).

***.WibuCmRaC file - License Request**

The creation of a *.WibuCmRaC file is bound to the physical ownership of a *CmDongle*. On creation the Firm Code o 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 *.WibuCmRaC file 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](#)^[420].

***.WibuCmRaU file - License Update**

On the basis of this *.WibuCmRaC file, you as licensor, are able to generate a so-called update (*.WibuCmRaU) file, 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 *.WibuCmRaU file 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 *.WibuCmRaU file is invalid.

This is of special importance, for example, when the *.WibuCmRaU file 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.

***.WibuCmRaM file - Modified Context File**

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

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

The following figure illustrates this process.

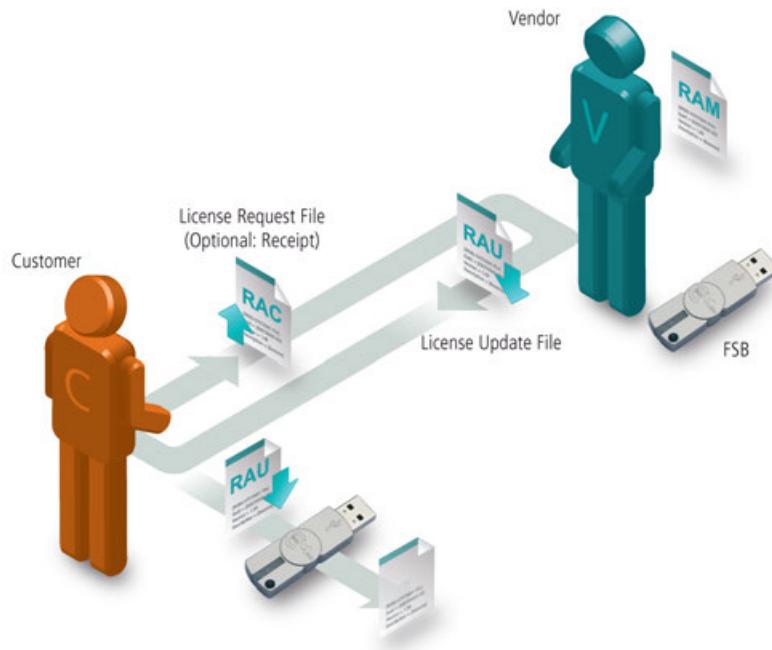


Figure 190: *CmFAS - File-based Remote Update CmDongle*

***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 (*.wibuCmRaC- or license request file) the customer has to import an empty license container he receives by the vendor. This LIF file (License Information File) in the *.wbb (WIBU Binary) format holds information on the [binding scheme](#)²⁸ and [additional activation options](#)²⁹ 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 license update file (*.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 file) is not created.

The following figure illustrates this process.

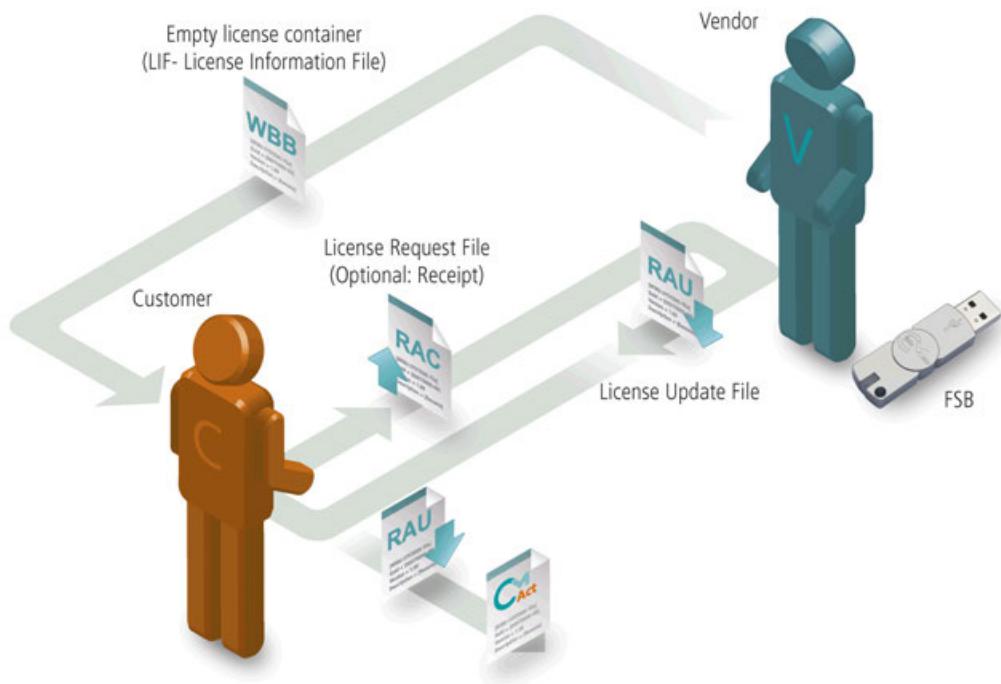


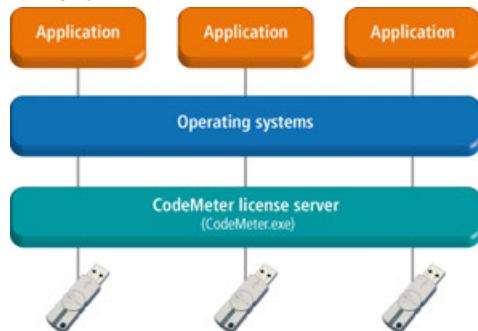
Figure 191: *CmFAS - File-based Remote Update CmActLicense*

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.



i 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

i 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.

i 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:



Mac

CodeMeter Runtime Kit (Mac OS X starting with 10.6), for PowerPC and Intel processors



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

DEB packages, e.g. for Debian, Ubuntu

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 - mere driver installer for systems without GUI



Sun Solaris

CodeMeter Runtime for SPARC

CodeMeter 64-bit extension for SPARCV9 (starting with Version 4.10)

CodeMeter Runtime for i386

CodeMeter 64-bit extension for AMD64 (starting with Version 4.10)

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 ³⁷⁴ 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](#)³⁷⁴ are available which comprise files, registry entries and settings of specific runtime components. Setup developer are able to use them for own installer.

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 in 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](#)⁴⁷² 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.msn | 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 in 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* holds, 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).

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](#)³⁷⁶, [directed installing of features](#)³⁷⁷ and [using of central configuration parameter on integration of merge modules](#)³⁷⁷ 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.

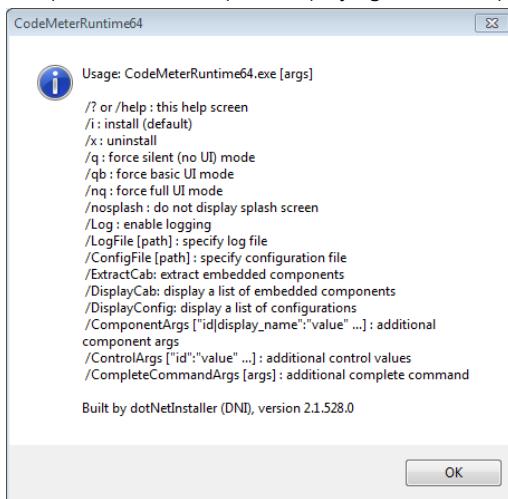


Figure 192: CodeMeterRuntime.exe - Commandline options

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**.

The commandline input

```
CodeMeterRuntime.exe /ComponentArgs "*" ":" /qn"
```

allows a "silent" installation without user interaction.

The commandline input

```
CodeMeterRuntime.exe /ComponentArgs "*" ":" /qn REINSTALLMODE=omusv  
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 <code>CmRuntimeMerger</code> module and the following secondary features |
| DotNET_Modules | holds the <code>WibuCmNet.msn</code> module and holds the file <code>wibucmnet.dll</code> , the language files. and the policy files |
| WibuShellExtension | holds the <code>ShellExtensionMerger</code> module |
| User_Help | holds the <code>CmUserHelp</code> module |



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 <code>CmRuntimeMerger</code> module and the following secondary features |
| DotNET_Modules | corresponds to the <code>WibuCmNet.msn</code> module and holds the file <code>wibucmnet.dll</code> , the language files. and the policy files |
| WibuShellExtension | holds the <code>ShellExtensionMerger</code> module |



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 | <ul style="list-style-type: none"> Preventing the start of <i>CodeMeter Control Center</i> at the end of the installation Disabling the <i>CodeMeter Control Center</i> entry in the auto start directory |
| run | Disabling the <i>CodeMeter Control Center</i> entry in the auto start directory |
| auto | Preventing the start of <i>CodeMeter Control Center</i> at the end of the installation |
| all | <ul style="list-style-type: none"> Start of <i>CodeMeter Control Center</i> at the end of the installation Enabling the <i>CodeMeter Control Center</i> 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 (<i>CodeMeter Control Center</i> , <i>User Help</i> , <i>CmDust</i> etc.)  Please note that the <i>CmDust</i> entry in the start menu no longer exists. Creating the log file then alternative must be triggered using the commandline tool <i>cmu</i> . |
| yes | Creating all shortcuts (<i>CodeMeter Control Center</i> , <i>User Help</i> , <i>CmDust</i> etc.)  If parameter PROP_MAKESC is not set, it corresponds to value yes. |

e.g. 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 <i>User Help</i> . |
| yes | Disabling the <i>User Help</i> 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 |
|--------------------------|--|
| <i>CodeMeter.exe</i> | <i>CodeMeter License Server</i> |
| <i>CodeMeter.l1**</i> | Language files for <i>CodeMeter License Server</i> |
| <i>CodeMeterCC.exe</i> | <i>CodeMeter Control Center</i> including support tool <i>CmDust</i> . |
| <i>CodeMeterCC_**.qm</i> | Language files for <i>CodeMeter Control Center</i> |
| <i>CodeMeter**.wbb</i> | <i>CodeMeter WebAdmin</i> |
| <i>WibuCm32.dll</i> | <i>CodeMeter®</i> runtime library (from %Windows%/system32) |
| <i>WibuCm32.1**</i> | Language files for the runtime library (from %Windows%/system32) |

| |
|--|
|  For mobile installation add to the <i>CodeMeter®</i> runtime a <i>CodeMeter.ini</i> file on the <i>CmDongle</i> . Then all settings are read from and written to the <i>CodeMeter.ini</i> 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.exe* in the same directory where *CodeMeter.ini* 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.

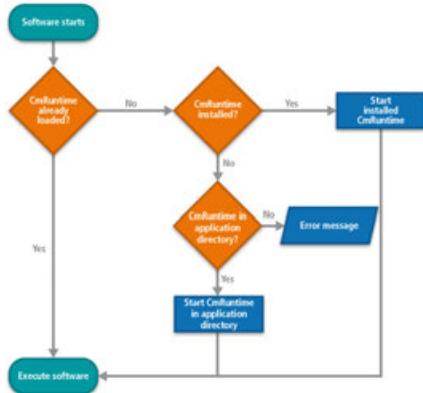


Figure 193: Behaviour CodeMeter License Server

UseMobileHandling

Starting with version 4.0, processes to exit *CodeMeter License Server* are automated. Enter "UseMobileHandling=1" into the [codemeter.ini](#)³⁷⁹ 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
UDPPacketingTime=1000
UDPPacketingTime=20
ApiCommunicationMode=2
IsNetworkServer=0
NetworkAccessFsB=0
NetworkPort=22350
NetworkTimeout=100
MaxMessageLen=67108864
BindAddress=0.0.0.0
UseUDPSA=1
CmAIDisabled=0
UseMobileHandling=1
  
```

Figure 194: 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](#)³⁷⁹ file the entry "ApiCommunicationMode=2".



Wibu-Systems recommends this setting on mobile installation. For further questions contact Wibu-Systems support.

10.4 CodeMeter 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 <i>CodeMeter License Server</i> Can be implemented as service with option /i if privileges are sufficient. |
| CodeMeter.l** | optional | Language files for <i>CodeMeter.exe</i> If no language file is installed the default language English is available. |
| CodeMeterCC.exe | recommended | Executable of <i>CodeMeter Control Center</i> |
| CodeMeterCC**.qm | optional | Language files for <i>CodeMeter Control Center</i> If no language file is installed the default language English is available. |
| cmu32(64).exe | recommended | Executable of <i>cmu</i> commandline program including support tool <i>CmDust</i> . If not installed commandline-based <i>CodeMeter Control Center</i> functions are not available. If not installed an access to information gathered by support application <i>CmDust</i> is not possible which limits support assistance. |
| CodeMeterXX.wbb | recommended | <i>CodeMeter WebAdmin</i> localized in several languages. |
| WibuCm32(64).dll | recommended | Includes <i>CodeMeter® API</i> functions, e.g. support application <i>CmDust</i> . The default installation path is [%\Windows\System32]. |
| WibuCm32(64).lXX | optional | Language files for the <i>WibuCm32(64).dll</i> ; 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 <i>CodeMeter License Central</i> . |
| WibuCmTrigger32(64).lXX | optional | Language files for the <i>WibuCmTrigger32(64).dll</i> . 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).

11.2 Enabling

The *CodeMeter®* feature Enabling allows you by using an access code to activate or deactivate the complete *CmContainer*, or single Firm Items or license entries in the *CmContainer*.

If the Enabling refers to the Implicit Firm Item (IFI) level, the complete *CmContainer* can be activated or deactivated. However, you also able to implement the Enabling feature for complete Firm Item levels or license entries, i.e. Product Items.

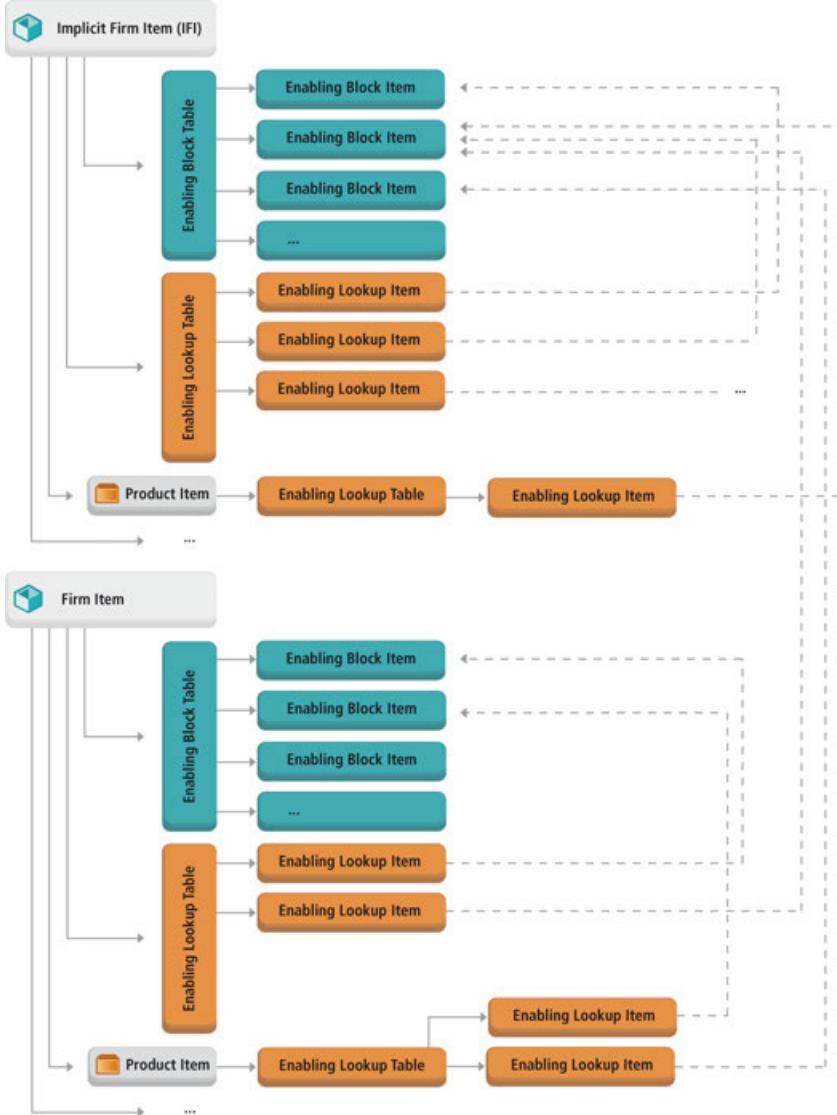
Additionally, you have the option to temporarily define the activation: then temporary Enabling allows the access to the *CmDongle*, single Firm Item levels or license entries as long the *CmContainer* is connected to the PC and the PC is powered on.



Note that you are able to activate or deactivate only the Implicit Firm Item level, your own Firm Item level, and your license entries. In reverse, other licensors cannot apply enabling to your Firm Item level and license entries.

The controlled enabling or disabling comprises the interaction of several constituent parts:

- on/off switches (Enabling Blocks) and
- mapping (Lookup) between Enabling Blocks, Firm Item level or license entries.

Figure 195: 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).

11.2.1.1 Enabling Access Code

The access code authenticates the access to an Enabling Block.

The Access Key (sequence of characters) free to choose and required to change the other elements of an Enabling Block. It is also required when an Enabling Block is changed.

The Access Key is converted by a hash algorithm into a 16 bytes sequence, and the result is called Enabling Access Code (EAC). This code is directly saved to the Enabling Block. Also the direct input of the 16 bytes Enabling Access Code is possible.

For each operation changing the settings in an Enabling Block the user is required to re-enter the Access Key. This is converted by the same hash algorithm to the Enabling Access Code, and compared to the Enabling Access Code saved in the *CmContainer*.

11.2.1.2 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](#)^{b394}). 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`.

11.2.1.3 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](#)⁴¹⁷.

Beginning with *CodeMeter®* Version 4.30 and Firmware Version 1.18 the Temporary Enabling feature applies to all Enabling Blocks.

Up to the *CodeMeter®* Version 4.30 and Firmware Version 1.18 the Temporary Enabling feature for an Enabling Blocks is exclusively limited to the level of the Implicit Firm Item (IFI). However, for applying the Temporary Enabling feature you are also able to attach a license container or license entry to Enabling Blocks located in the IFI.

The attachment is possible only in this direction. From within the IFI license container you cannot attach to Enabling Blocks located somewhere else in the *CmDongle*.

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.

11.2.1.4 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.

11.2.2 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.



An Enabling Lookup Table can locate at the levels of the Implicit Firm Items (IFI), Firm Items, and license entries.



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).

11.2.2.1 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 <i>CmContainer</i> . |
| 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 <i>CmContainer</i> . |
| 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.

11.2.2.2 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  355 | Create, edit and delete Enabling Blocks |
| Core API  310 | Enabling options |
| Programming API  311 | Call corresponding classes |

11.2.3 Enabling Example

The following small examples illustrates the general concept of Enabling.

Requirements

Using *AxProtector* we protect a basic editor application which features a new function to change the font via an menu item. This function is to be provided to selected users only with no time restrictions; the other users are to be able to use the basic editor as usual.

On using the new function for changing the font a dialog is to display. In it the input of the correct password is required to activate (*enabled*) the license. The password corresponds to the Enabling Access Code. On next PC reboot and again using the new function the dialog is to open again.

This example is designed along the lines of the Second Sample help file you find after installing the *CodeMeter®* Development Kit in the user directory "%\Documents\Public Documents\WIBU-SYSTEMS".

Solution

The solution is the Temporary Enabling of the new function. For this two separate license entries are programmed: the basic editor application has Firm Code 10, Product Code 201000, and a Feature Code 1; the new function Firm Code 10, Product Code 201001, and also a Feature Code 1.

Implementing Temporary Enabling requires first to create an Enabling Block in the license container with the Firm Code 10. In a second step you attach this Enabling Block to the Lookup of the Firm Item 10. In both cases use the commandline tool *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](#)¹⁷ you find *CmBoxPgm* at the usual locations. See also [Enabling options in CmBoxPgm](#)¹⁸.

1. For creating the Enabling Blocks enter the following commandline addressing the *CmDongle* and the Firm Item 10.

```
cmboxpgm.exe -qs<serial number> -f10 -e:tp -eac:"MyAccess" -et:"FontApp" -edta:none -em:d,t+ -ca
```

The following table lists the options used and their meaning.

| Option | Description |
|--------|--|
| -e:tp | Creating an access type Time PIN (tp). In the case you create an access type Simple PIN instead, you cannot specify an additional description or a Disable Time as usage period for the Enabling feature. |
| -eac: | Input of the Enabling Access Code. Here "MyAccess". |
| -et: | Input of the Enabling Text describing the Enabling Block. Here "FontApp". |
| -edta: | Input of the Disable Time. Here none for no time-based restrictions of the Enabling feature. You can also specify here an expiration date. |
| -em: | Input of the Enabling Mode. Here d,t+ for the temporary Enabling feature. |
| -ca | Creates (adds) the Enabling Block. |

2. For attaching the Enabling Block to the Lookup table of Firm Item 10 enter the following commandline addressing the *CmDongle* and the Firm Item 10:

```
cmboxpgm.exe -qs<Serial number> -f10 -e0:tp -eac:"MyAccess" -eat-t10,201001,1:mod,loc:req+ -ca
```

The following table lists the options used and their meaning.

| Option | Description |
|--------|--|
| -e0:tp | Addressing the Enabling Block with the Index 0 of access type Time PIN (tp). Which index you use here depends on the number and sequence of other Enabling Blocks eventually created. |
| -eac: | Input of the Enabling Access Code. Here "MyAccess" as the access code you specified when you created the Enabling Block. |
| -eatt | 10,201001,1 Specifies the license entry of the new function 'Font'. In the case you address the IFI, the parameters would correspond to a value of 0 or |

| Option | Description |
|----------|---|
| | the Firm Code in the case you like to attach to a Firm Item. |
| :mod,loc | When <u>activated</u> the Enabling Level has the privilege modify (mod). When <u>deactivated</u> the Enabling Level has the privilege to read the Firm and Product Codes only (loc). |
| :req+ | In the case this Enabling Block is attached to other Lookup tables which locate at other levels (IFI, Firm Item, Product Item), the use of the Required Flag ensures that the Enabling Block is only activated when the respective defined settings of the Enabling Blocks and the attachments do not conflict. |
| -ca | Attaches the Enabling Block to the Lookup table. |

3. For displaying the *CmContainer* content using *CmBoxPgm* after you have created the Enabling Block and attached it to the Lookup table, please type the following commandline:

```
cmboxpgm.exe -qs<Serial number> -f10 -e0:tp -eac:"MyAccess" -eat-
t10,201001,1:mod,loc:req+ -ca
```

The resulting lines contain the following lines:

```
Firm·Code·10·at·[17], ·Box·Based, ·Individual·Key¶
- ·Firm·Access·Counter¶
· ·Data: 65535¶
- ·Firm·Update·Counter¶
·· ·Data: ·0-12 ·(12)¶
- ·Firm·Item·Text¶
·· ·Data: ·(0 ·characters)¶
··"¶
- ·Firm·Precise·Time¶
·· ·Data: ·2011-02-10·12:01:04 ·(UTC)¶
* ·Enable·Block·Table¶
·0[-T]: ·TimePin, ·16 ·bytes ·Access ·Code¶
·· ······Disable·Time ·= ·(never)¶
·· ······Text ·(13 ·character(s)) : ·"FontAnwendung"¶
- ·No ·Enable ·Lookup ·Table ·exists¶
** ·Product ·Code ·201000 ·at ·[17], ·dependencies ·= ·dsu¶
- ·No ·Enable ·Lookup ·Table ·exists¶
·· ·· Feature ·Map ·dependencies ·= ·dsu¶
·· ·· 0x000000001¶
* ·Product ·Code ·201001 ·at ·[16], ·dependencies ·= ·dsu¶
- ·Enable ·Lookup ·Table¶
·· ·Loc[0] ·- ·Required, ·Valid ·- ·E:Modify ·(7) , ·D:Locate ·(0)¶
·· ·· Feature ·Map ·dependencies ·= ·dsu¶
·· ·· 0x000000001¶
```

The same information on the successful creation and attachment you also receive by using the commandline tool *cmu*.

Open *cmu* in the directory %\Program Files%\CodeMeter\Runtime\bin by the command *cmu[32].exe*. Alternatively, execute *cmu* using the system menu item "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**". For the operating systems Mac OS, Linux and Sun this command is provided by the lookup..

Please type in the following commandline:

Cmu32.exe --enabling

The resulting lines contain the following lines:

```
- cmstick with serial Number 2-506478 and version 1.18
* FC=0(IFI)
1 EnableBlock: +-----+ |Index 00: "Default" (TimePin) enabled No DisableTime |
|-----+
1 Item attached: +-----+ |status | Level | Index | required? | enable level | disable level |
+-----+
| - | 7 | 00 (IFI) | yes | modify (7) | locate (0) |
+-----+
IFI-Level = modify
+-* PC=0 (Ref=16)
+-* PC=1000 (Ref=17)

* FC=10003
+-* PC=1 (Ref=16)

* FC=10
1 EnableBlock: +-----+ |Index 00: "FontAnwendung" (TimePin) temp enabled No DisableTime |
|-----+
1 Item attached: +-----+ |status | Level | Index | required? | enable level | disable level |
+-----+
| - | 0 | 00 | yes | modify (7) | locate (0) |
+-----+
PI-Level = locate
```

The dialog prompting for the password input (Enabling Access Code) when using the new function can be implemented by individually programming the UserMessage in *AxProtector*. Then a password dialog displays



11.3 Using 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.

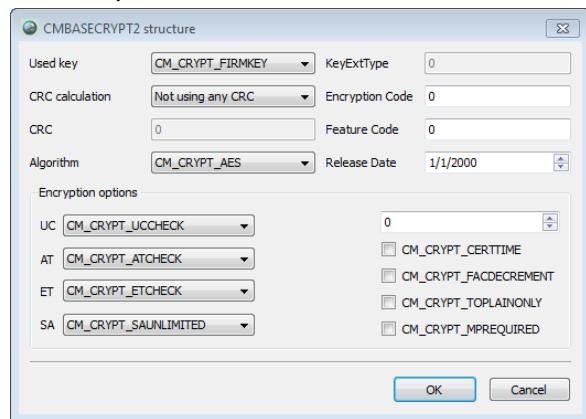


Figure 196: Encryption Alternatives

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 197: 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). This 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*³⁵⁶, 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.

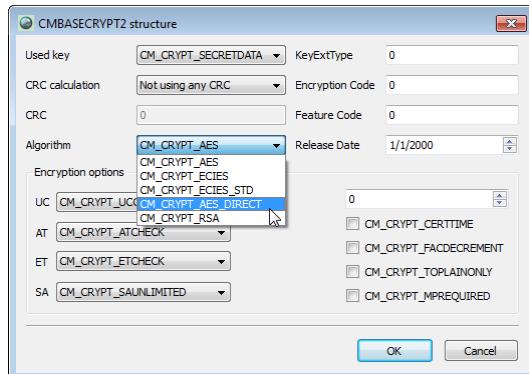


Figure 198: AES direct for Secret Data, Hidden Data direct encryption

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.

[CodeMeter API Guide](#)³¹³ provides you the API commands and function blocks required: [Authentication API](#)³¹⁰, [Encryption API](#)³¹⁰, [Blocks](#)³⁰⁹ for executing various encryption and decryption operations.

Please contact Wibu-Systems support for further questions.

11.4 Time 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 "**Content | CmContainer**".

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, Mac OS, 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 there-

fore 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):

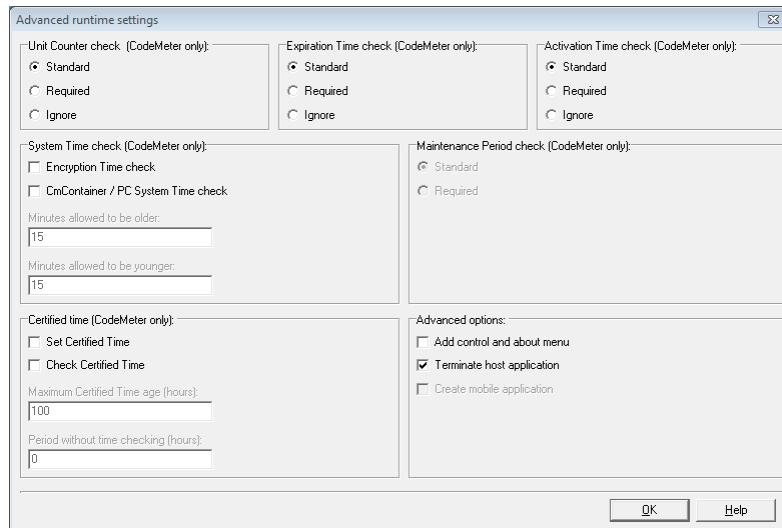


Figure 199: *AxProtector* - "Advanced Runtime Settings"

Times in *CodeMeter WebAdmin*

In *CodeMeter WebAdmin*, you see the *CodeMeter®* System Time, PC System Time, and the Certified Time.

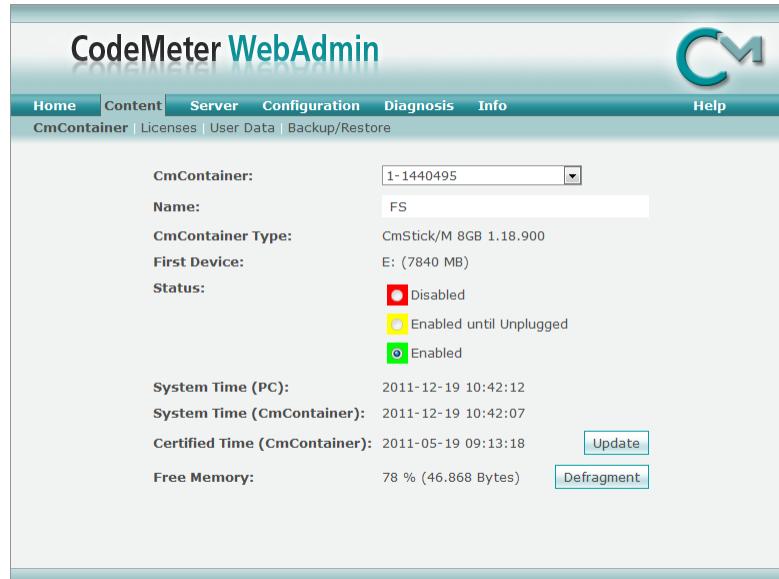


Figure 200: CodeMeter WebAdmin – "Content | CmContainer"

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.

11.5 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. It can be programmed to any other value.



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 ***CmCrypt*** provides the ***Fac_Decrement*** option which decrements the FAC by a value of 1.

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.

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.

11.6 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.

11.7 CodeMeter in a Wide Area Network (WAN)

By default, *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](#)³⁹⁹ using CmWAN and describe the necessary steps required for *CodeMeter®* sided [implementing](#)⁴⁰⁰.

11.7.1 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.

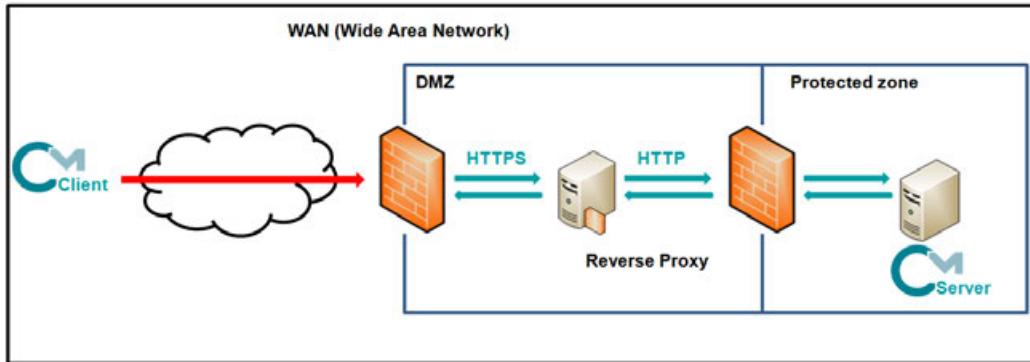
At the communication level, the reverse proxy forwards the HTTPS request as HTTP request to the *CodeMeter License Server* on the server. Conversely, the reverse proxy returns the HTTP response of the *CodeMeter License Server* on the server back to the *CodeMeter®* clients secured by HTTPS.

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.

Communication level:**Authentication level (HTTPS):**

| Client user: | Reverse Proxy |
|---|--|
| <ul style="list-style-type: none"> checks server certificate proves the identity to the reverse proxy using "username" and "password" | <ul style="list-style-type: none"> delivers server certificate proves the identity of the server to the client |

Figure 201: 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.

i If testing and using a self-created test certificate at the reverser, please note that you import this certificate as root certificate 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 <i>CodeMeter License Server</i> Minimum Version 5.0 |
| transforming from HTTPS to HTTP and vice versa support of authentication tasks | |

11.7.2 CodeMeter-sided Implementation

For using CmWAN you have to configure CodeMeter® in the following areas and the following tools:

- [license programming](#)⁴⁰¹ (*CmBoxPgm*, *CodeMeter API Guide*)
- [application encryption](#)⁴⁰⁴ (*AxProtector*)
- [CmWAN network communication](#)⁴⁰⁴ (*CodeMeter WebAdmin*; registry or *Server.ini* entries)

11.7.2.1 Programming of licenses

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](#)³⁴⁰ w 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>:w
```

/[CmContainer] addresses the *CmContainer* to be programmed (see [here](#)³³³)

/f [...] / specifies the license entry (Firm Code/Product Code)

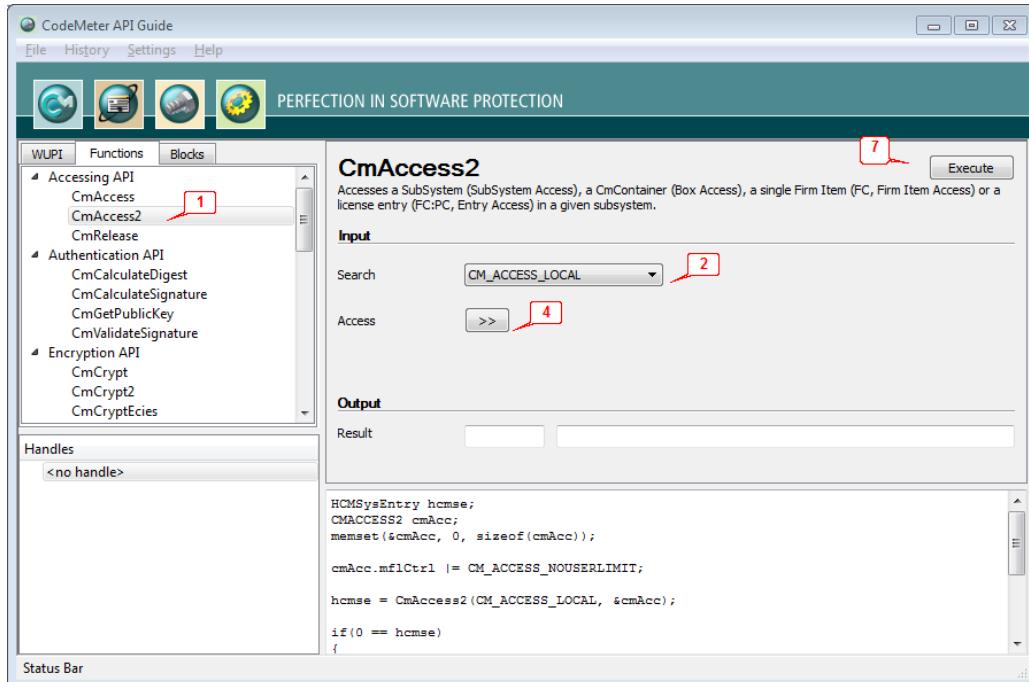
p[...]

Programming the license access using *CodeMeter API Guide*

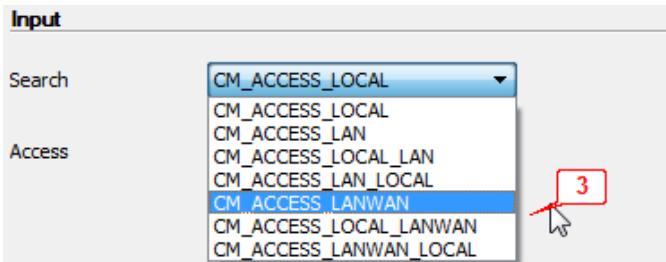
In order to use the required license access via the **CmAccess2** structure using the tool [CodeMeter API Guide](#)³¹³, open the *CodeMeter API Guide* via the start menu item "**CodeMeter | Tools | CodeMeter API Guide**" and then proceed as follows:

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.

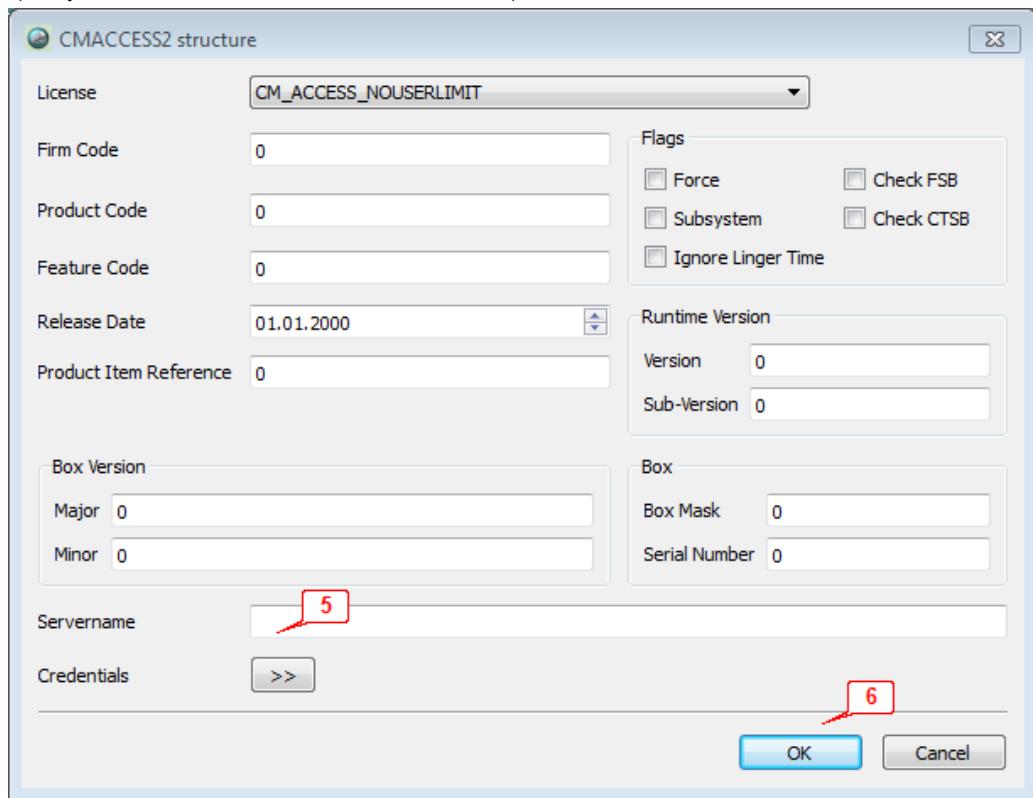
- CM_ACCESS_LANWAN_LOCAL:

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.

11.7.2.2 Encrypting the application to be protected

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](#)²⁷⁰ 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`²⁷² 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.

11.7.2.3 Configuring CmWAN network communication

For configuring the *CmWAN* network communication two alternative ways are provided: either by using [CodeMeter WebAdmin](#)⁴⁰⁵ or by configuring the [profiling](#)⁴⁰⁵.



Please note that using profile settings is necessary only, if *CodeMeter License Server* should not run.

11.7.2.3.1 CodeMeter WebAdmin Configuration

For setting up *CodeMeter*® in a WAN, please proceed as follows:

Configure Server

1. Start *CodeMeter WebAdmin* (see [here](#)⁴³⁸).
2. Navigate to the page "[Configuration | Server](#)"⁴⁴⁵.
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](#)⁴¹⁶. For non-Windows operating systems see [here](#)⁴¹¹.

Configure Server Search List

5. Navigate to the page "[Configuration | Network](#)"⁴⁴³.
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.

- 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](#)⁴¹⁶. For non-Windows operating systems see [here](#)⁴¹¹.

11.7.2.3.2 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 |

| Operating system | Registry / Server Entry |
|------------------|---|
| Windows | %Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini |
| Mac OS | /Library/Preferences/com.wibu.CodeMeter.Server.ini |
| Linux | /etc/wibu/CodeMeter/Server.ini |
| Solaris | /etc/opt/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.

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. 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](#)⁴⁸²).

 *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®* Runtime Server is installed as service (Windows) or as deamon (Linux, Mac) 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®* Runtime Server should not be active, it can be [manually started or stopped](#)⁴⁸¹. 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 |
|  Mac OS | [Programs CodeMeter CodeMeter Control Center] | CodeMeterMacX |
|  Linux | [Applications System CodeMeter Control Center] or [Applications Accessories CodeMeter Control Center] | CodeMeterLin |
|  Sun Solaris | /opt/CodeMeter/CodeMeterCC] | CodeMeterSun |

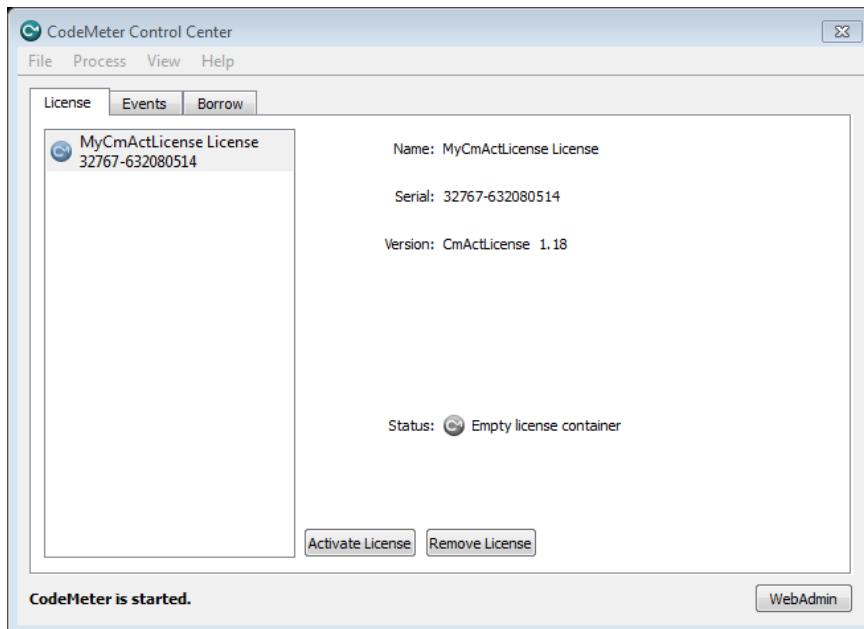
Activating *CmActLicense* licenses

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.

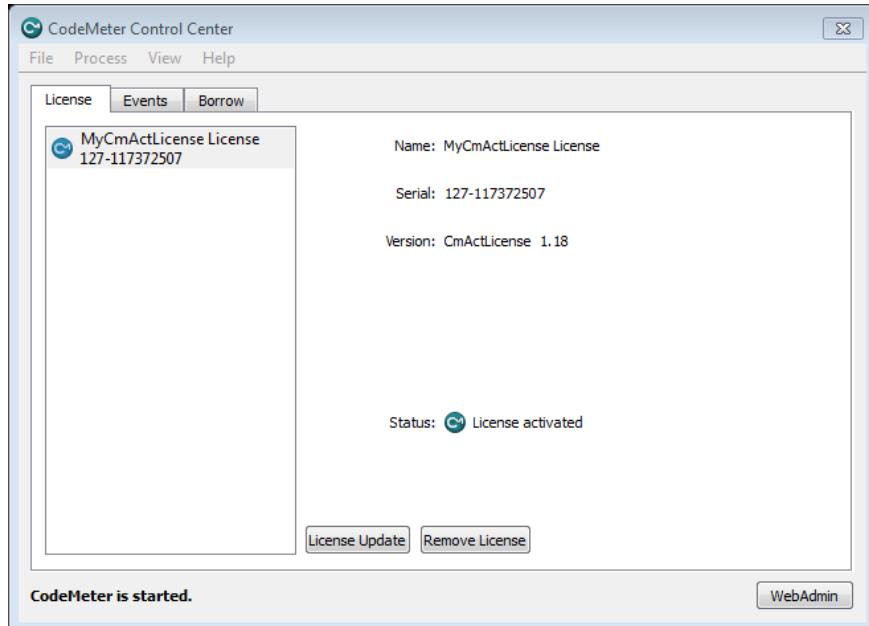
Before you are able to activate *CmActLicense* licenses for your PC you require a separate file you obtain from your software vendor. This licenses information file corresponds to an empty license container. It serves to collect hardware properties of your PC as a kind of 'finger print' for the subsequent 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](#)⁴²⁷) 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.wbb, 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 <code>CmDust-Result.log</code> created with CmDust ⁴⁷² .. |
| Telephone | Contact our <i>CodeMeter®</i> 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 CodeMeter 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, Mac) 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] |
|  Mac OS | [Programs - CodeMeter - CodeMeter Control Center] |
|  Linux | [Applications - System - CodeMeter Control Center] or [Applications - Accessories - CodeMeter Control Center] |
|  Sun Solaris | [/opt/CodeMeter/CodeMeterCC] |



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 | <ol style="list-style-type: none">1. Navigate via "Windows Control Panel Administrative Tools Services" to CodeMeter Runtime Server.2. Right mouse-click and 'Start' or 'Stop' the service. Alternatively, use the "Action" menu of CodeMeter Control Center. |

| Operating system | Description |
|---|--|
|  Mac OS | <p>1. Navigate via "System preferences Other" to the CodeMeter® icon.</p>  <p>2. Click the CodeMeter® Icon. The CodeMeter dialog displays</p>  <p>3. Click the "Stop Service" or "Start Service" button to stop or start the service.</p> |
|  Linux | <p>1. Call the following script with 'sudo' root privileges to stop the service: <code>etc/init.d/codemeter stop</code></p> <p>2. Call the following script with 'sudo' root privileges to start the service: <code>etc/init.d/codemeter start</code></p> |

| Operating system | Description |
|---|--|
|  Sun Solaris | <ol style="list-style-type: none"> Call the following command as 'root' to stop the service: %> svcadm disable codemeter Call the following command as 'root' to start the service: %> svcadm enable codemeter Call the following command as 'root' to disable the service until next boot: %> svcadm disable -t codemeter |

 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®.

12.2.1 Structure and Navigation

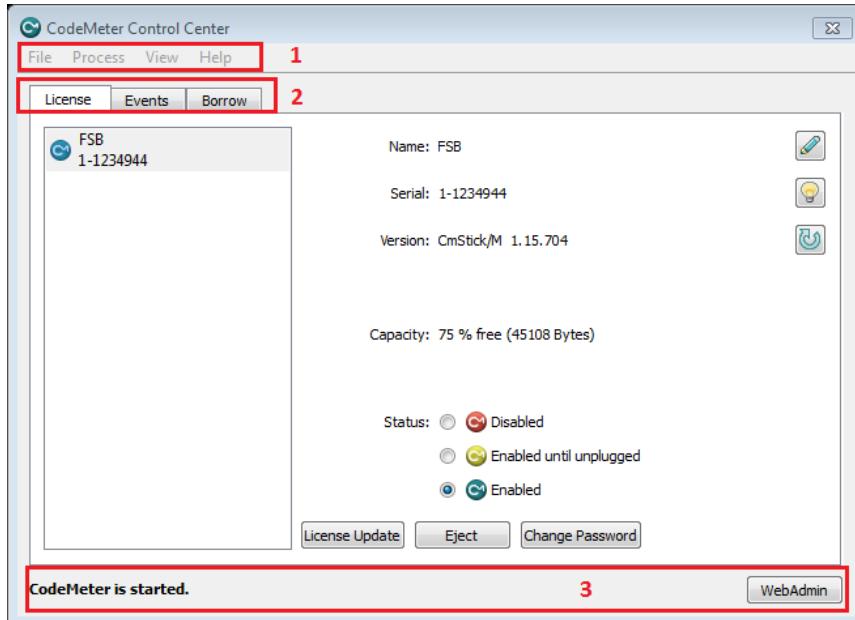


Figure 202: *CodeMeter Control Center - Overview*

The *CodeMeter Control Center* user interface is divided in three separate areas:

- [menu bar](#) (1)
- Tab areas (2)
- [Status and Open CodeMeter WebAdmin](#) (3).

Starting CodeMeter Control Center

You access and start CodeMeter Control Center in several ways:

| Open | |
|--|--|
| <ul style="list-style-type: none"> Double-click on the CodeMeter®  or  symbols in the info area of the Windows task bar | |
| <ul style="list-style-type: none"> 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 CmDongle Password. |
| Help | Opens the CodeMeter® help. |
| About | Shows general information on CodeMeter® components. |
| Quit | Exits but not shuts down the service CodeMeter License Server. |
| <ul style="list-style-type: none"> Navigation by the "Start All Programs CodeMeter Control Center" system menu. | |

In the info area of the Windows task bar, different colors of the CodeMeter® symbols represent different status conditions of connected CmContainer.

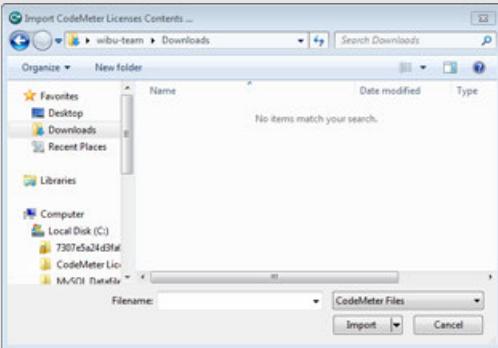
| Color | Status |
|---|--|
| Grey  | No CmContainer is connected, or CodeMeter License Server is not started. |
| Green  | An activated CmContainer is connected. |
| Blue  double | Several CmContainer are connected and activated until disconnected.. |
| Yellow  | A CmDongle is connected and activated until it is disconnected. |
| Red  | A deactivated CmContainer is connected. |

Figure 203: CodeMeter® Symbols Windows Task Bar

12.2.2 Menu Bar

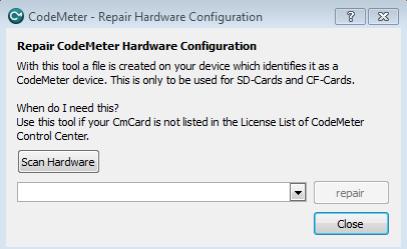
File Menu

| Element | Description |
|----------------|---|
| Import license | <p>In order to import license contents using CodeMeter Control Center, proceed as follows:</p> <ol style="list-style-type: none"> Select the  "File Import License..." item. 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. |

| Element | Description |
|--|--|
| |  <p>Figure 204: CodeMeter Control Center - Import Licenses</p> |
| | <p>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.</p> |
| WebAdmin  | Opens CodeMeter WebAdmin in the default Internet browser. Alternatively, press the key combination <CTRL>+W. |
| Logging  | <p>Saves all CodeMeter® events to a log file. Alternatively, press the key combination <CTRL>+L.</p> <p> When you activate the logging, this also affects the logging display in CodeMeter WebAdmin on the "Diagnosis Logfile" page.</p> <p>On Windows operating systems this log file is stored to the directory %\Program Files%\CodeMeter\Logs.</p> <p> This log file is especially important for trouble shooting.</p> |
| Preferences | Opens CodeMeter WebAdmin and is defaulted on the page where you are able to apply network settings. |
| Exit  | <p>Exits CodeMeter Control Center. Alternatively, press the key combination <CTRL>+Q.</p> <p> The serviceCodeMeter License Server however is not shut down.</p> |

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 |

| Element | Description |
|---|---|
|  Start CodeMeter Service  | <p>refreshed.</p> <p>Starts the service <i>CodeMeter License Server</i>.</p> |
| |  Use this menu item if <i>CodeMeter License Server</i> has been stopped before, for example, when you made changes on the network settings in <i>CodeMeter WebAdmin</i> which require the restart of the service. |
| |  When you have administrator privileges under Windows you can also manage the <i>CodeMeter License Server</i> service by setting the desktop management (System Settings Management Services). |
| Repair Hardware Configuration  | <p>Repairs the hardware configuration of the <i>CmDongle</i> form factors SD Card and CF Cards. This tool is required if the <i>CmCard</i> hardware is not listed in the license list of <i>CodeMeter Control Center</i>.</p>  |
| Stop CodeMeter Service  | <p>Stops the service <i>CodeMeter License Server</i>.</p> |

View Menu

| Element | Description |
|--|---|
| Hide Window | <p>Minimizes and hides the <i>CodeMeter Control Center</i> window back into the info area of the Windows task bar. Alternatively, press the key combination <CTRL>+M.</p> |
| Refresh | <p>Refreshes the display of all connected <i>CmContainer</i>. Alternatively, press the key <F5>.</p> |
| Zoom in | <p>Enlarges the display in the Events tab area. Alternatively, press the key combination <CTRL>++.</p> |
| Zoom out | <p>Scales down the display in the Events tab area. Alternatively, press the key combination <CTRL>+-.</p> |
| Copy Event Content | <p>Copies the event actions in the Events tab area to the clipboard. Alternatively, press the key combination <CTRL>+C.</p> |
| Clear Event Content <input type="checkbox"/> | <p>Deletes the event actions in the Events tab area. Alternatively, press the key combination <ALT>+C.</p> |
| Show all connected CmContainer  | <p>Shows all connected <i>CmContainer</i> including details in the Events tab area. Alternatively, press the key combination <ALT>+S.</p> |
| List all open Handles  | <p>Shows all open handles in the Events tab area. Handles work as references for the developer for further programming.</p> |

| Element | Description |
|--|---|
|  Show all available License Entries | Shows all <i>CmContainer</i> 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. |

Help Menu

| Element | Description |
|---|--|
|  Help | Opens the <i>CodeMeter®</i> online help. Here you access the help files on <i>CodeMeter License Server</i> and <i>CodeMeter Control Center</i> . |
|  Register CmDongle | Opens the secure website https://my.codemeter.com to register <i>CmDongles</i> . |
|  About | Informs on the started <i>CodeMeter Control Center</i> version. |

12.2.3 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](#)⁴²⁵.

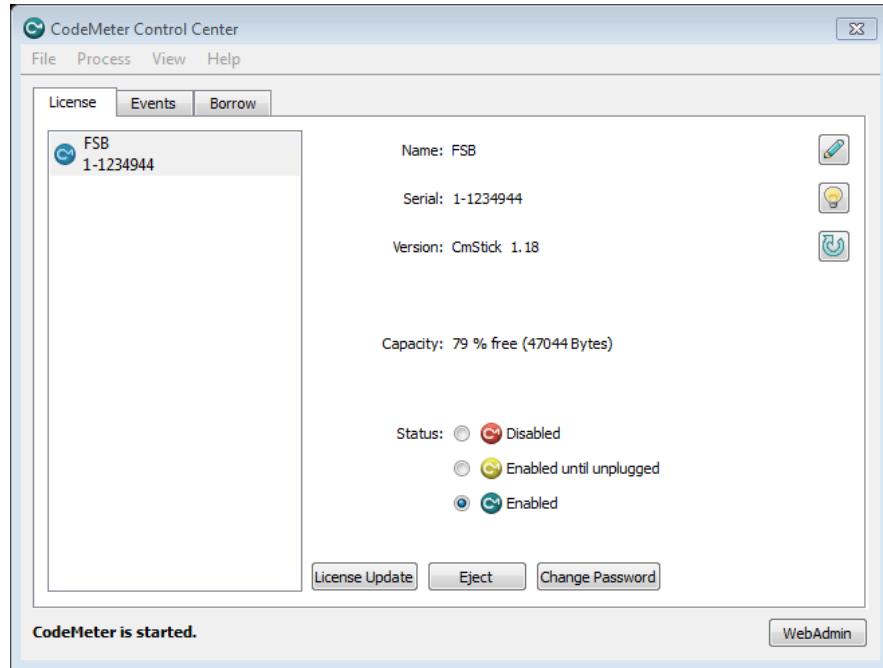
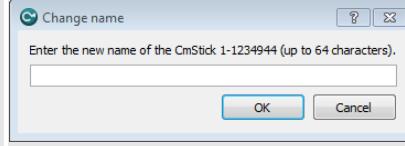


Figure 205: CodeMeter Control Center License Tab

| Element | Description |
|---------|---|
| | Changes and displays the name of the selected <i>CmContainer</i> . In the subsequent dialog you can edit the name.  A small dialog box titled 'Change name' with the instruction 'Enter the new name of the CmStick 1-1234944 (up to 64 characters)' and a text input field. It has 'OK' and 'Cancel' buttons. |
| | Flashes the LEDs of the selected <i>CmStick</i> . This eases the identification of a <i>CmStick</i> , if several <i>CmSticks</i> are connected. |
| | Updates the firmware of the selected <i>CmDongles</i> . 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 <i>CodeMeter Control Center</i> automatically connects to the Firmware Update Server of Wibu-Systems. You are prompted to enter your <i>CmDongle</i> Password in order to confirm |

| Element | Description | | | | | | | | |
|---|--|-------|--------|---|---|---|---|---|---|
| | <p>this action.</p> <p> The update may take a couple of minutes. You <u>must not</u> remove the <i>CmDongle</i> before this process is finished. Otherwise, irreparable damage of the <i>CodeMeter® SmartCard Chip</i> may occur.</p> | | | | | | | | |
| Capacity | <p>Informs on the capacity of the <i>CodeMeter® SmartCard Chip</i> of a selected <i>CmDongle</i>. The capacity is displayed in percent format, and by number of absolute bytes.</p> <p> Please note that this value tells nothings about the memory allocation of an eventual flash memory of a <i>CmDongle</i>.</p> | | | | | | | | |
| Status | <p>The status group informs on the activation status of the selected <i>CmDongle</i>.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="background-color: #00AEEF; color: white;">Color</th><th style="background-color: #00AEEF; color: white;">Status</th></tr> </thead> <tbody> <tr> <td></td><td>The connected <i>CmContainer</i> is disabled. No licensed application can use license information in the <i>CmContainer</i>.</td></tr> <tr> <td></td><td>The <i>CmDongle</i> is enabled as long as it is connected. If the <i>CmDongle</i> is removed from the PC, automatically the licensed access by applications is deactivated.</td></tr> <tr> <td></td><td>The <i>CmContainer</i> is fully enabled. In the case of a <i>CmDongle</i>, the licensed access of applications is still featured even if the <i>CmDongle</i> is removed.</td></tr> </tbody> </table> <p> Wibu-Systems recommends the activation status "Enabled until plugged out". This ensures that even when a <i>CmDongle</i> is lost, unauthorized access to the licenses and personal data in the <i>CmDongle</i> is not possible.</p> | Color | Status |  | The connected <i>CmContainer</i> is disabled. No licensed application can use license information in the <i>CmContainer</i> . |  | The <i>CmDongle</i> is enabled as long as it is connected. If the <i>CmDongle</i> is removed from the PC, automatically the licensed access by applications is deactivated. |  | The <i>CmContainer</i> is fully enabled. In the case of a <i>CmDongle</i> , the licensed access of applications is still featured even if the <i>CmDongle</i> is removed. |
| Color | Status | | | | | | | | |
|  | The connected <i>CmContainer</i> is disabled. No licensed application can use license information in the <i>CmContainer</i> . | | | | | | | | |
|  | The <i>CmDongle</i> is enabled as long as it is connected. If the <i>CmDongle</i> is removed from the PC, automatically the licensed access by applications is deactivated. | | | | | | | | |
|  | The <i>CmContainer</i> is fully enabled. In the case of a <i>CmDongle</i> , the licensed access of applications is still featured even if the <i>CmDongle</i> is removed. | | | | | | | | |

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.



Figure 207: *CodeMeter Control Center* - Enter Password

3. Click the "OK" button to confirm the status change.

| Element | Description |
|------------------------|---|
| License Update | <p>Click this button to request new, or update existing licenses for selected <i>CmContainer</i>. The <i>CodeMeter Field Activation Service (CmFAS) Assistant</i>  opens.</p>  |
| Eject | <p>Click this button to disconnect the selected <i>CmDongle</i>. The <i>CmDongle</i> logs off from the operating system, and can be safely removed from the PC.</p> |
| Change Password | <p>Click this button to change the password of the selected <i>CmDongle</i>. In the following "Change Password" dialog please complete the respective fields.</p>  <p>Figure 208: CodeMeter Control Center - CmFAS Assistant</p> <p>Figure 209: CodeMeter Control Center - Change Password</p> <ol style="list-style-type: none"> 1. Enter in the "Old Password" field the currently used <i>CmDongle</i> password. 2. Enter in the "New Password" field the new desired <i>CmDongle</i> password. 3. Re-enter in the "Retype Password" field the new desired <i>CmDongle</i> password. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i If you forgot the <i>CmDongle</i> password, you have the option to set a new <i>CmDongle</i> password by using the <i>CmDongle</i> Master Password. </div> <ol style="list-style-type: none"> 4. Click the "OK" button to confirm your input. 5. Activate the "Input Master Password" option and specify your <i>CmDongle</i> Mas- |

| Element | Description |
|---------|---|
| | <p>Master Password in the "Old Password" field.</p> <p>A Master Password you have received when you registered at the website my.codemeter.com.</p> <p>In order to register, use the "Help Register CmDongle" menu item.</p> <p>A registration bears several advantages and serves to provide security when using <i>CodeMeter®</i>. Only when you are registered loosing the own password can be remedied by requesting a Master Password.</p> |

12.2.4 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 | ...](#)"⁴¹⁶ menu item.

You log the content for the event view using the "[File | Logfile](#)"⁴¹⁵ menu item.

12.2.5 Borrowing Tab

This tab informs on borrowable licenses as a feature of *CodeMeter®* [license borrowing](#)⁵⁵. 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.

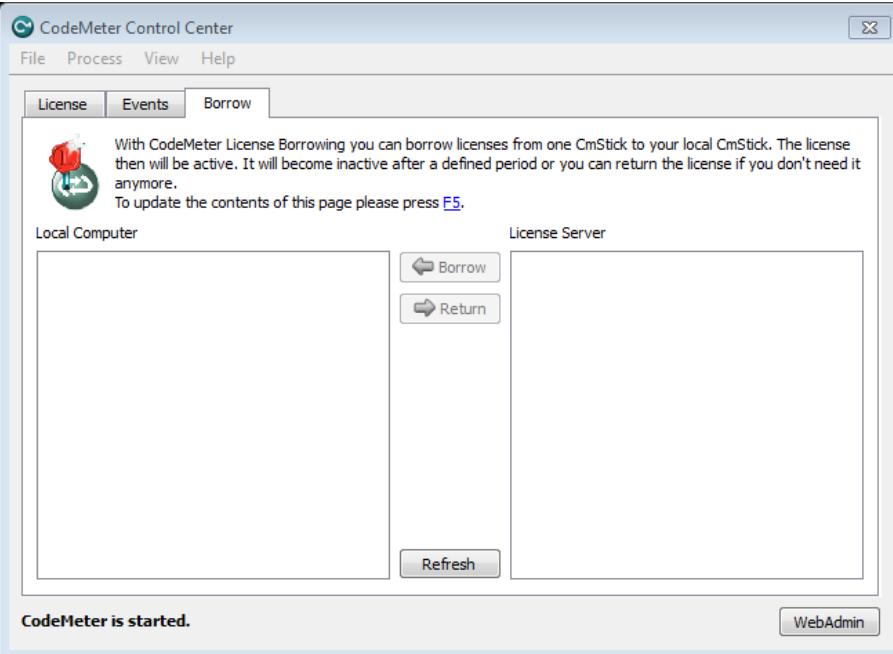


Figure 210: *CodeMeter Control Center - Borrowing Tab*

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.

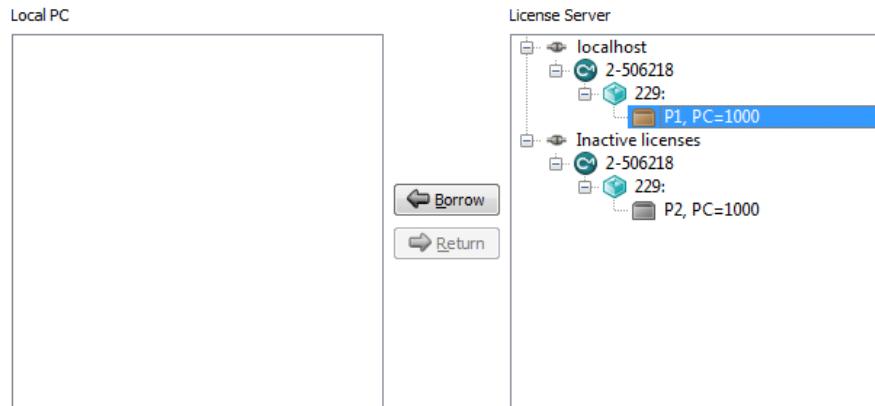


Figure 211: *CodeMeter Control Center - Borrow Licenses*

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.

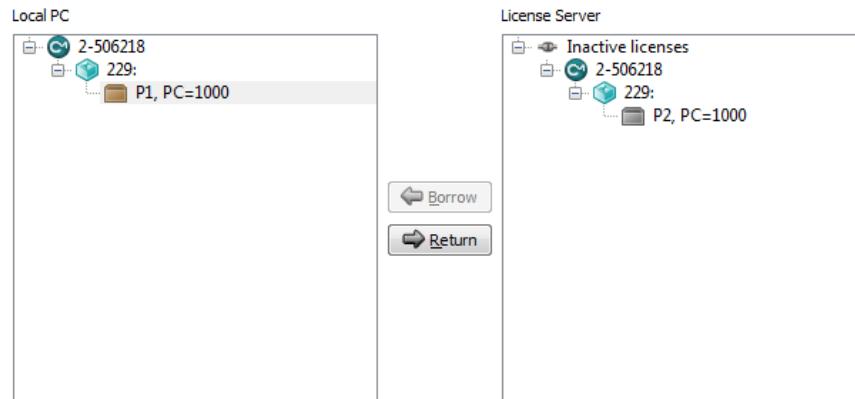


Figure 212: *CodeMeter Control Center - Return Licenses*



For refreshing the display of the tab press the key **<F5>** or the "**Refresh**" button.

12.2.6 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.

12.3 Importing and Updating Licenses

The [CmFAS Assistant](#)⁴²⁵ 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*.

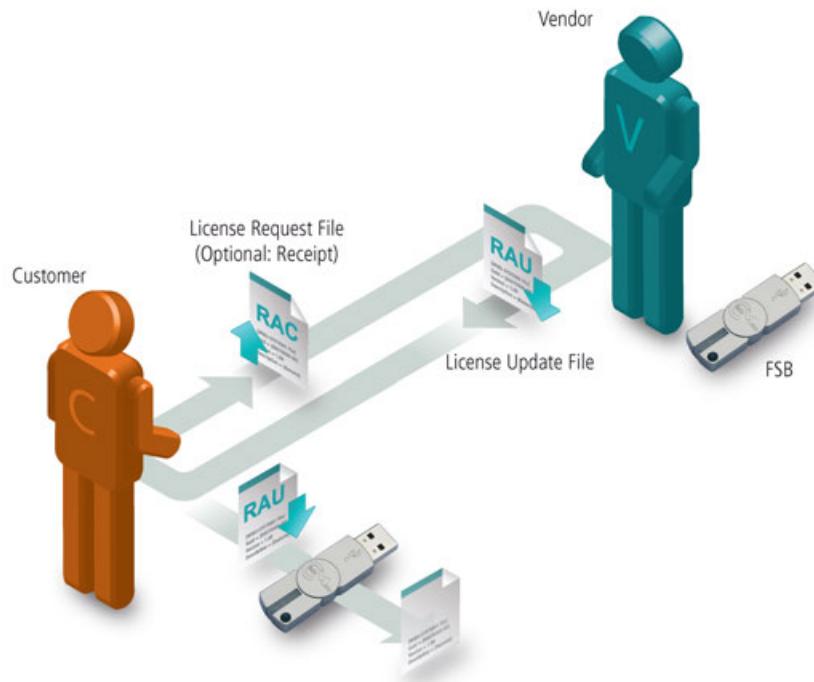


Figure 213: *CmFAS - File-based Remote Update*

12.3.1 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.

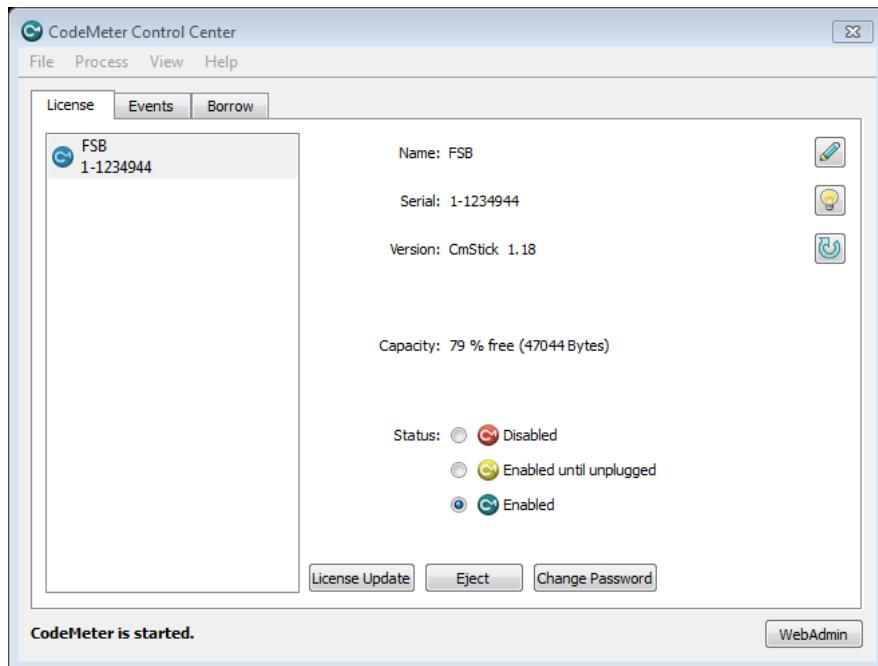


Figure 214: License Update - *CodeMeter Control Center*

The *CodeMeter Field Activation* (CmFAS) Assistant opens with a welcome dialog.



Figure 215: *CmFAS Assistant*

3. Click the "**Next**" button.

12.3.1.1 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.

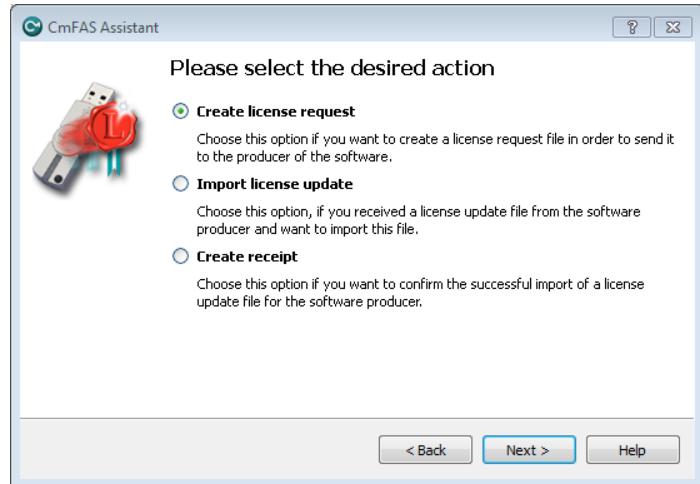


Figure 216: CmFAS - Create License Request

12.3.1.1.1 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.

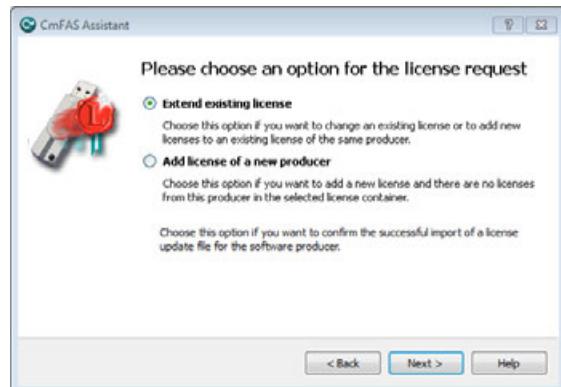


Figure 217: CmFAS – Extend existing License

When you extend an existing license, select the software vendor (producer) for which you want to create a license request. After your selection click the "Next" button.



Figure 218: *CmFAS - License Extension - Select Producer*

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 219: *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.

12.3.1.1.2 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 producer. Select "**Add license of a new producer**" and click the "**Next**" button.

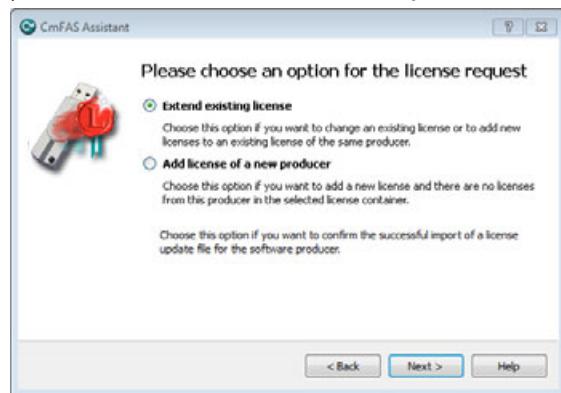


Figure 220: *CmFAS – New License*

In the next dialog, specify the Firm Code you received by the software vendor, and click the "**Next**" button.



Figure 221: *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.

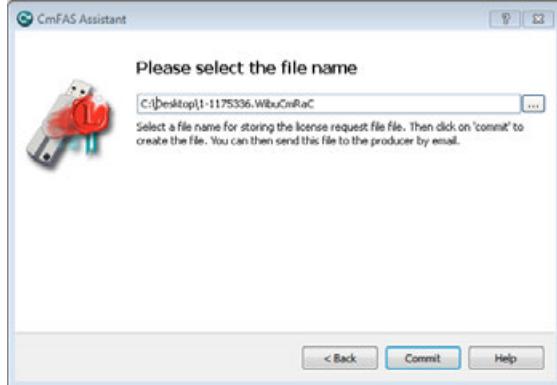


Figure 222: *CmFAS - Save File*

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.

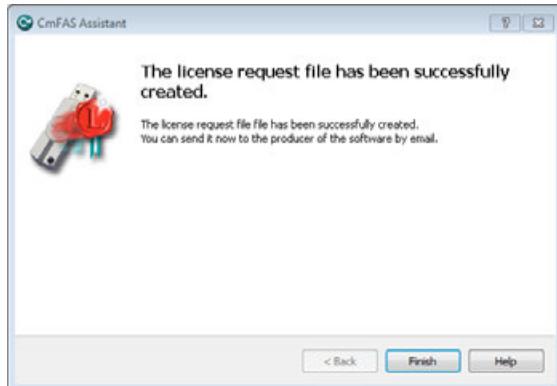


Figure 223: *CmFAS - Receipt*

12.3.1.2 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.

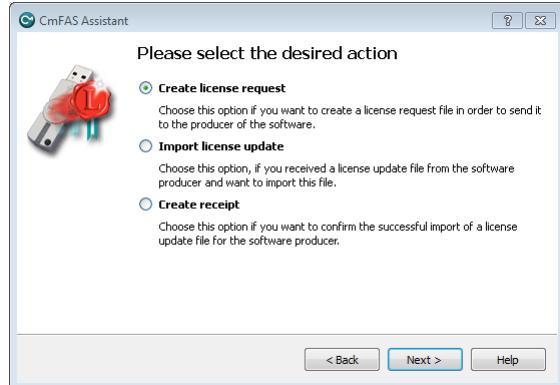


Figure 224: CmFAS - Import License Update

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.

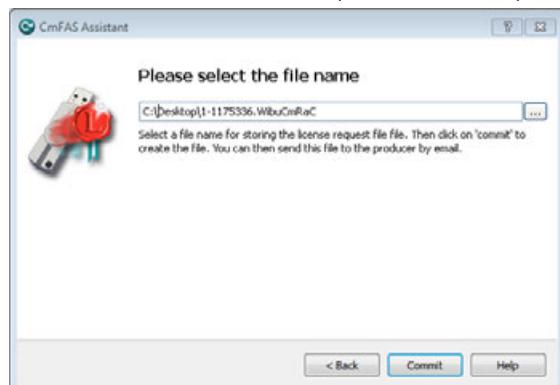


Figure 225: CmFAS - License Update - Save File

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.

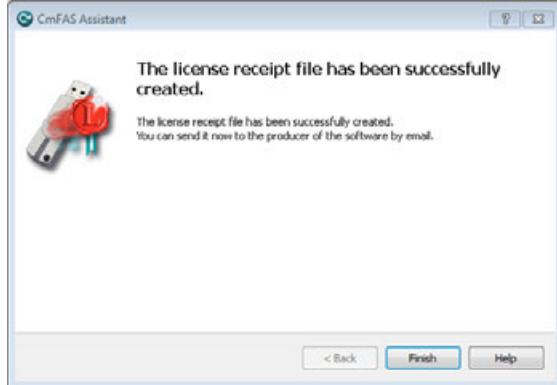


Figure 226: CmFAS - License Update - Receipt

12.3.1.3 Create Receipt

In the start menu, select the option "**Create Receipt**", then click the "**Next**" button.

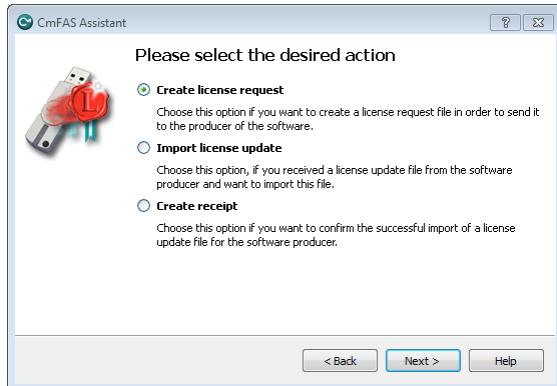


Figure 227: CmFAS - Create Receipt

In the next dialog, select the software-vendor you want to send the receipt to, then click the "**Next**" button.

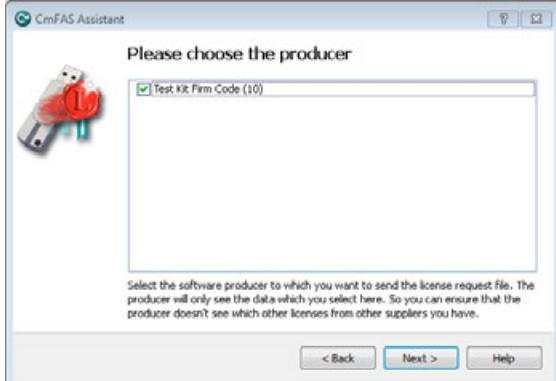


Figure 228: *CmFAS - Create Receipt - Producer*

Save the receipt file using the "**Commit**" button and send it to the software vendor.

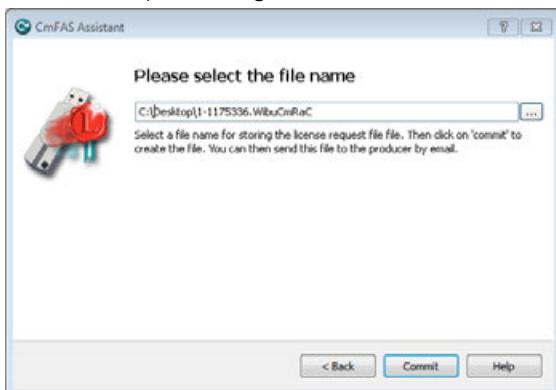


Figure 229: *CmFAS - Create Receipt - Save File*

The successful creation of the receipt file is confirmed in the next dialog. Click on the "**Finish**" button to complete this process.

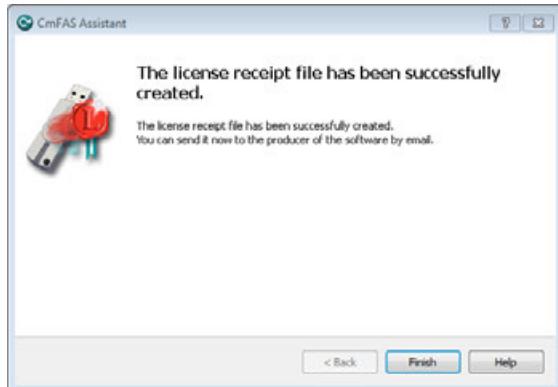


Figure 230: *CmFAS - Create Receipt - Receipt*

12.4 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**⁴³⁸: host, *CmContainer*
- **configuration**⁴⁴²: use as network server, proxy settings, access protection, remote access, time server, backup
- **display**: display of all existing licenses locally⁴⁵⁴ and on the network⁴⁵⁹, view of license conditions, session information
- **management**: management of network licenses by manual allocation of licenses
- **diagnosis**: logging
- **backup**.

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 modu- |

| Term | Description |
|--------------------------|--|
| Code Version | le/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 <i>CmDongle</i> Password. This license container is identified by the number of "0". |
| License Quantity | Informs on the total number of licenses available for a license. |
| 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 <u>once</u> 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

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.

12.4.1 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 (127.0.0.1).



However, an actual connection to the Internet is not established.

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.

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 |
| Mac OS | /Library/Preferences/com.wibu.CodeMeter.Server.ini |
| Linux | /etc/wibu/CodeMeter/Server.ini |
| Solaris | /etc/opt/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: <ul style="list-style-type: none">• Windows: IPv6, IPv4; Shared Memory• Linux/Mac: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.

12.4.2 Starting CodeMeter WebAdmin

CodeMeter WebAdmin is a web-based tool to be displayed with each standard Internet browser. The following table shows existing start options.

| Operating System | Start |
|---|--|
|  Windows | <ul style="list-style-type: none"> • via <i>CodeMeter®</i> symbol in the task bar (right mouse-click) and selection of 'WebAdmin' item. • via the 'WebAdmin' option in <i>CodeMeter Control Center</i> • directly in your Internet browser when typing in the URLs: http://localhost:22350 or http://127.0.0.1:22350. |
|  Mac OS / Linux | <ul style="list-style-type: none"> • via <i>CodeMeter®</i> in the task bar (right mouse-click) and selection of 'WebAdmin' item. • via the 'WebAdmin' option in <i>CodeMeter Control Center</i> • directly in your Internet browser when typing in the URLs: http://localhost:22350 or http://127.0.0.1:22350. |

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:22350> or <http://127.0.0.1:22350> directly in the address field of your Internet browser.

12.4.3 Status Information

Here you obtain first information on connected *CmContainer*:

- general information
- information on *CmContainer*

12.4.3.1 General Information

The "**Home**" page displays general status information on your PC, *CodeMeter License Server* and *CodeMeter WebAdmin*.

The screenshot shows the "CodeMeter WebAdmin" interface. At the top, there is a navigation bar with tabs: Home, Content, Server, Configuration, Diagnosis, Info, and Help. To the right of the tabs is a logo consisting of a stylized 'C' and 'M'. Below the navigation bar, the main content area displays various system details:

| | |
|-------------------|---|
| Host Name: | fs1.wibu.local |
| IP Address: | 192.168.139.128 |
| Operating System: | Microsoft Windows 7 Enterprise Edition, 64-bit (build 7600) |
| Server Startup: | 19.Dez.2011 10:23:12 |
| Runtime Version: | 4.40 |
| Server Version: | Version 4.40 vom 16.Dez.2011 (Build 687) |
| WebAdmin Version: | Version 4.40 of Dec/16/2011 |

Figure 231: *CodeMeter WebAdmin – "Home"*

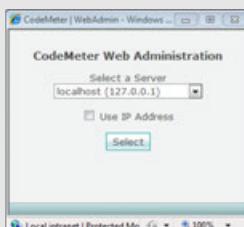
| Element | Description |
|------------------|---|
| Host Name | The " Host Name " button shows the name of the actual PC on which the service <i>CodeMeter License Server</i> is started. A search request using the port 22350 is sent to the network. For changing the host, please proceed as follows: <ol style="list-style-type: none">1. Click the "DNS-Name" button of the host. The "CodeMeter Web Administration" dialog opens.2. Use the "Select a Server" dropdown control to select another PC on which <i>CodeMeter®</i> is also |

Figure 232: *CodeMeter WebAdmin – "Home | Host Name"*

| Element | Description |
|--------------------------|---|
| | <p>started and the service <i>CodeMeter License Server</i> runs.</p> <p>3. Activate the Use IP Address option to use the network address of the found PC.</p> <p>4. Click the "Select" button to use the selected PC.</p> |
| IP Address | Shows information on the network address in use. |
| Operating System | Shows information on the operating system in use. |
| Server Start Time | Shows information on the start time of the server. |
| Runtime Version | Shows information on the <i>CodeMeter®</i> runtime in use. |
| Server Version | Shows information on the <i>CodeMeter®</i> version on the server. |
| WebAdmin Version | Shows information on the <i>CodeMeter WebAdmin</i> version in use. |

12.4.3.2 Information on CmContainer

The "Content | CmContainer" page displays information on selected *CmContainer*.

CodeMeter WebAdmin

Home Content Server Configuration Diagnosis Info Help

CmContainer | Licenses | User Data | Backup/Restore

CmContainer: 1-1440495

Name: FS

CmContainer Type: CmStick/M 8GB 1.18.900

First Device: E: (7840 MB)

Status: Disabled
 Enabled until Unplugged
 Enabled

System Time (PC): 2011-12-19 10:42:12

System Time (CmContainer): 2011-12-19 10:42:07

Certified Time (CmContainer): 2011-05-19 09:13:18 **Update**

Free Memory: 78 % (46.868 Bytes) **Defragment**

Figure 233: CodeMeter WebAdmin – "Content | CmContainer"

| Element | Description |
|-------------------------|--|
| CmContainer | Select the CmContainer on which the information is to be displayed. If several <i>CmContainer</i> are connected, select the desired <i>CmContainer</i> using the drop-down control. |
| Name | Shows the Name of the selected <i>CmContainer</i> . If you want to change the name of your <i>CmContainer</i> , use <i>CodeMeter Control Center</i> . |
| CmContainer Type | Shows the Type of the selected <i>CmContainer</i> . |
| First Device | Shows the drive information of the selected <i>CmDongle</i> . |
| Status | Shows the current activation status of the selected <i>CmContainer</i> . The following status settings are displayed: <ul style="list-style-type: none">• Disabled: The connected <i>CmContainer</i> is deactivated and not usable by any |

| Element | Description |
|-------------------------------------|--|
| | <p>application.</p> <ul style="list-style-type: none"> • Enabled until Unplugged: The <i>CmDongle</i> is activated as long as it is connected and supplied by electrical energy. After removed from the PC the <i>CmDongle</i> is automatically deactivated. • Enabled: The <i>CmContainer</i> is fully activated. If a <i>CmDongles</i> is removed, the license access is still possible after plugout. <p>You change the activation status of a <i>CmContainer</i> using CodeMeter Control Center⁴¹⁴.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Wibu-Systems recommends the activation status "Enabled until Unplugged" when using <i>CmDongles</i>. This ensures that even when a <i>CmDongles</i> is lost, unauthorized access to the licenses and personal data in the <i>CmDongle</i> is lost, unauthorized access to the licenses and personal data is not possible. </div> |
| System Time (PC) | Shows the System Time (local time on the PC) when the service <i>CodeMeter License Server</i> has started. |
| System Time (CmContainer) | Shows the saved System Time (internal time) of the <i>CmContainer</i> . These two system times may differ due to the pending synchronization process. |
| Certified Time (CmContainer) | <p>Shows the Certified Time saved in the <i>CmContainer</i>. In order to update the Certified Time of your <i>CmContainer</i> using a <i>CodeMeter® Time Server</i>, click the "Update" button. This action is confirmed by a dialog.</p> <div data-bbox="387 812 871 1003" style="border: 1px solid #ccc; padding: 10px; width: fit-content; margin: auto;"> <p>Message from webpage</p> <p> This will update all Timestamps on the CmContainer</p> <p style="text-align: center;"><input type="button" value="OK"/> <input type="button" value="Cancel"/></p> </div> |
| | Figure 234: <i>CodeMeter WebAdmin - Update Certified Time</i> |
| Free Memory | Shows the Free Memory of the SmartCard chips of a <i>CmDongle</i> , i.e. how much space is available for the programming of additional license entries. |
| Defragment | Click the " Defragment " button to defragment the memory of the <i>CmDongle</i> chip. |

12.4.4 Configuration

Here you configure settings for network server status, proxy, access control, remote access, time server and backup.

12.4.4.1 Network

In order to set up CodeMeter® in a network environment use the "**Configuration | Network**" page.

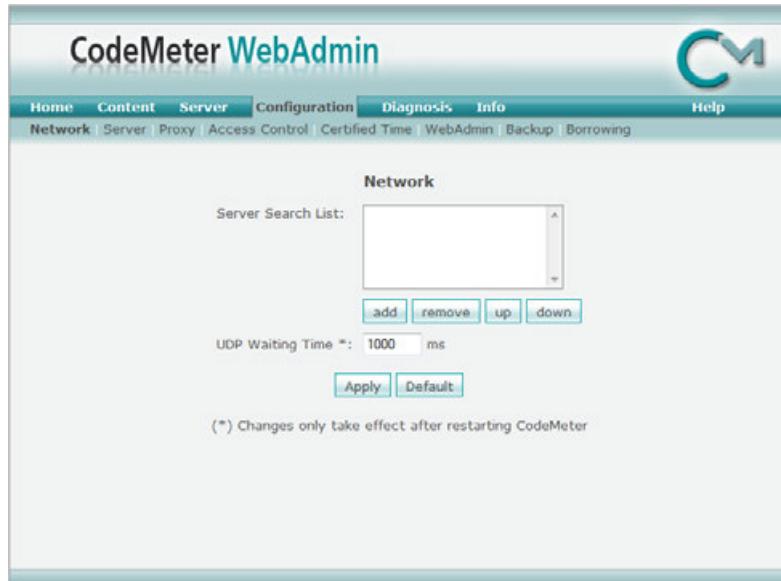


Figure 235: *CodeMeter WebAdmin – "Configuration | Network"*

| Element | Description | |
|---|--|--|
| Server Search List | Use a Server Search List to define access to and order of CodeMeter® network LAN and WAN (Wide Area Network) servers. 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. You save the changes you made by using the " Apply " button. | |
| <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> You set back the settings of the server search list using the "Default" button. </div> Alternatively, you are also able to set the Server Search List using the configuration files CodeMeter.ini or Server.ini. The tabel below shows you where to find the respective files. | | |
| Operating System | Configuration File | |
| Windows | %Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini | |
| Mac OS | \Library\Preferences\com.wibu.CodeMeter.Server.ini | |
| Linux | \etc\wibu\CodeMeter\Server.ini | |
| Solaris | \etc\opt\CodeMeter\Server.ini | |

In the separate section [ServerSearchList] define the server as the example 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](#)⁴¹⁶. For non-Windows operating systems see [here](#)⁴¹¹.

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):

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.

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.

12.4.4.2 Server

In order to set up CodeMeter® in a network and/or a wide area network (WAN) use the "**Configuration | Server**" page.



Figure 236: CodeMeter WebAdmin – "Configuration | Server"

| Element | Description |
|---------------------------|---|
| Bind Adresse | Select the Network Address to which the service <i>CodeMeter License Server</i> is to be bound. <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> Primarily, this is required when the PC has several network cards (virtual adapter) and is to act as a network server providing its licenses on the network. </div> |
| Run Network Server | Activate the Run Network Server option to use the PC as <i>CodeMeter®</i> network server. Then this PC provides its <i>CodeMeter®</i> licenses on the network using the service <i>CodeMeter License Server</i> . |
| Network Port | Specify a Network Port . By default, the port 22350 is used for the <i>CodeMeter®</i> communication. This network port is registered at IANA (Internet Assigned Numbers Authority) and uniquely assigned for the <i>CodeMeter®</i> communication. <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> You are able to customize this port value. However, make sure that all <i>CodeMeter License Server</i> use this port when a <i>CodeMeter®</i> protected application is to be used on the network. </div> |
| Run CmWAN Server | Activate the Run CmWAN Server option to use the PC in a wide area network (WAN) and allow license accesses. |
| CmWAN Port | Specify a CmWAN Port . By default, the port 22351 is used for the <i>CodeMeter®</i> |

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.

You save the changes you made by using the "**Apply**" button. You set back the settings of the server search list using the "**Default**" button..

When you define server 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](#)⁴¹⁶. For non-Windows operating systems see [here](#)⁴¹¹.

12.4.4.3 Proxy Settings

On the "**Configuration | Proxy**" page you define settings when using a proxy server.

Figure 237: CodeMeter WebAdmin - "Configuration | Proxy"

| Element | Description |
|----------------------|---|
| Proxy Support | Activate this option for the support of a proxy server. When you use a proxy Server , specify here the IP address or the DNS name, and the Port number of the proxy server. |

| Element | Description |
|-----------------------|---|
| |  You require a proxy server when you apply Certified Time updates, or acquired product via an online shop. |
| Authentication | Activate this option for a required proxy server authentication. Specify the proxy user and password for the proxy server. |

If the selection of several *CodeMeter®* client PCs in *CodeMeter WebAdmin* is not possible, then try the following remedial action:

1. Exclude the related *CodeMeter License Server* from proxy use.
2. Type in the IP addresses or DNS name of those *CodeMeter®* client-PCs into the proxy exception list of the Internet Explorer: [Tools-Internet Options.. | Connections | Lan Settings | Advanced | Exceptions]

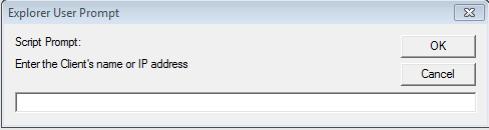
12.4.4.4 Access Control

On the "**Configuration | Access Control**" page you define settings managing the client access to *CodeMeter License Server*.



Figure 238: *CodeMeter WebAdmin* - "Configuration | Access Control"

| Element | Description |
|----------------|--|
| Clients | Shows a list of all client PCs which have the privilege to use <i>CodeMeter License Server</i> , i.e. to allocate a license. |

| Element | Description |
|-------------------|--|
| | <p> When this list is empty, each <i>CodeMeter</i>® client on the network is able to use <i>CodeMeter License Server</i>. This is the default setting.</p> <p>To add a new client to the client list, please proceed as follows:</p> <ol style="list-style-type: none"> 1. Click the "add" button. A prompt dialog displays.  <ol style="list-style-type: none"> 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. <p>To remove a client from the list, please proceed as follows:</p> <ol style="list-style-type: none"> 1. Click the "remove" button. The PC is now removed from the client list |
| Access FSB | <p>When you own a <i>CodeMeter</i>® 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 <i>CmContainer</i> or automatically protect applications.</p> <p> This option makes sense only for <i>CodeMeter</i>® licensee with an individual <i>CodeMeter</i>® Firm Code.</p> <p>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.</p> <p> When you define access settings, in some cases, this requires the restart of the <i>CodeMeter</i>® service. However, you do not have to eject or deactivate the <i>CmContainer</i>. After you specified the settings you are able to stop and then restart the <i>CodeMeter</i>® service in CodeMeter Control Center⁴¹⁶. For non-Windows operating systems see here⁴¹¹.</p> |

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. 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 |
|  Mac OS | /Library/Preferences/com.wibu.CodeMeter.Server.ini |
|  Linux | /etc/wibu/CodeMeter/Server.ini. |
|  Solaris | /etc/opt/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 Mac OS, Linux and Sun, you must stop the service *CodeMeter License Server* before. Otherwise, changes you have been made do not apply.

| Parameter | Description |
|--|---|
| Client<index>=&ltSubnetz>[,<serial>[,FC[,PC]]]] (Whitelist) | <p>Whitelist:</p> <p>These parameters hold the IP addresses of client PCs on the network which have the privilege to access the local <i>CodeMeter License Server</i>. When the IP address of a client is not on this list, the access is denied.</p> <p>If no whitelist exists, no other restrictions apply. The specification of subnets is possible.</p> <p>The syntax is as follows:</p> <p><code>Client<index>=&ltSubnetz>[,<serial>[,FC[,PC]]]</code></p> <p>The serial number has to follow the pattern MaskByte-Serial Number (e.g. 1-1179681).</p> <p>Example:</p> <p><code>Client1=192.168.0.0/24,1-123456,10,13</code></p> <p>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).</p> <p>The serial number, FC, and PC are optional.</p> <p> This whitelist corresponds to the client list in <i>CodeMeter WebAdmin</i>.</p> |
| Client<index>=&ltSubnetz>[,<serial>[,FC[,PC]]]] [SZ, optional] | <p>Blacklist:</p> <p>These parameters hold the IP addresses of client PCs on the network which have no privilege to access the local <i>CodeMeter License Server</i>. When an IP address of a client is on this list, the access is denied.</p> <p>If no blacklist exists, no other restrictions apply.</p> <p>The syntax is as follows:</p> <p><code>Client<index>=&ltSubnetz>[,<serial>[,FC[,PC]]]</code></p> <p>The serial number has to follow the pattern MaskByte-Serial Number (e.g. 1-1179681).</p> <p>Example:</p> <p><code>Client1=192.168.0.0/24,1-123456,10,13</code></p> <p>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).</p> <p>The serial number, FC, and PC are optional.</p> |

12.4.4.5 Certified Time

On the "Configuration | Certified Time" page you define settings for the CodeMeter® Time Server.

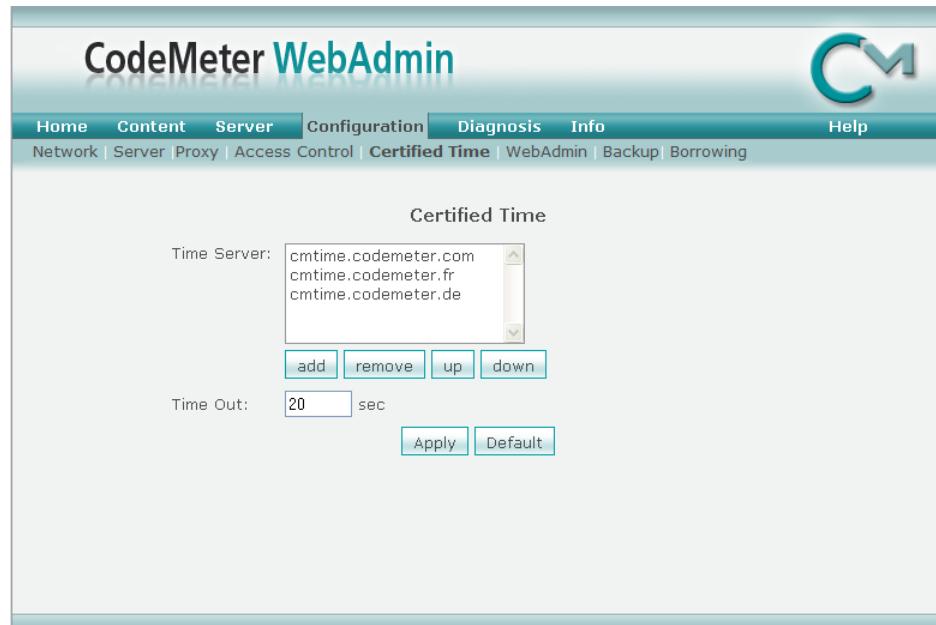


Figure 239: *CodeMeter WebAdmin - "Configuration | Time Server"*

| 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. Click the "Apply" button to save the changes you have made. By a previous click on the "Default" button you save the default settings. |

12.4.4.6 WebAdmin

On the "Configuration | WebAdmin" page you define settings to manage the remote access to CodeMeter WebAdmin and to customize the user interface language.

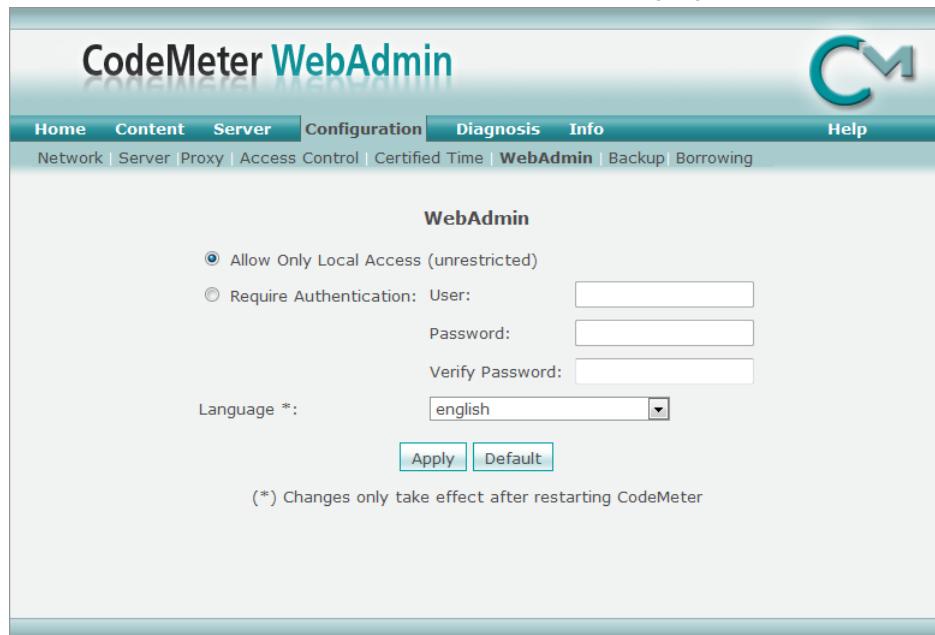


Figure 240: CodeMeter WebAdmin - "Configuration | WebAdmin"

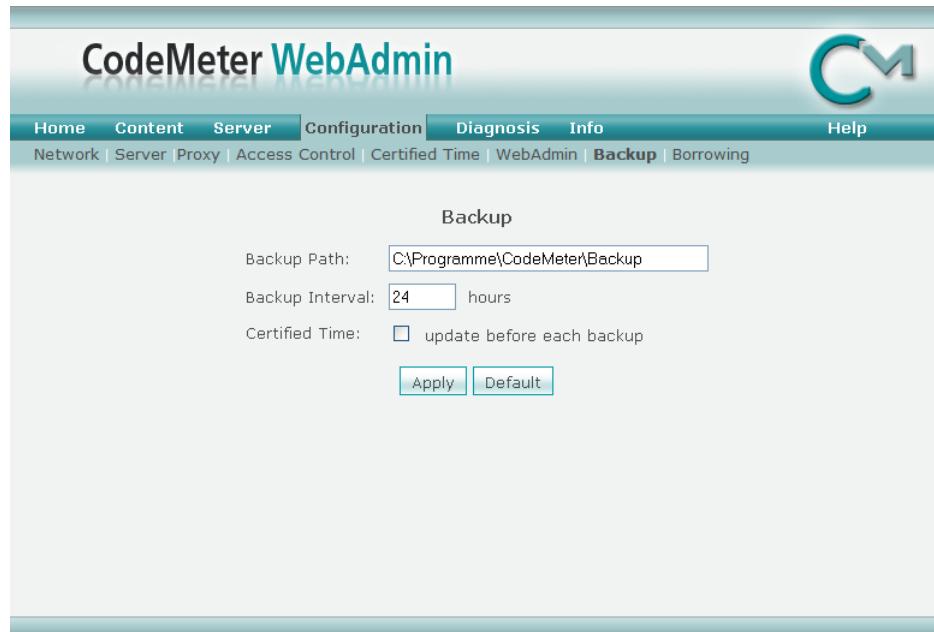
| Element | Description |
|---|---|
| Allow Only Local Access (unrestricted) | Activate this option to allow unrestricted local access to CodeMeter WebAdmin. |
| Require Authentication | Activate this option to enable remote write-access to CodeMeter WebAdmin. This allows a client to access the server via HTTP. This requires an authentication. Please complete the necessary authentication data in the fields User , Password and Verify Password . |
| Language | Customize the user interface language of CodeMeter WebAdmin using this dropdown control. You can select from the following languages: German, English, French, Italian, Japanese and Chinese. Click the " Apply " button to save the changes you have made. By a previous click on the " Default " button you save the default settings. Remote read access is featured and English set as default language. |

 Setting the remote access, in some cases, 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](#)

| Element | Description |
|---------|--|
| |  ⁴¹⁶ . For non-Windows operating systems see here  ⁴¹¹ . |

12.4.4.7 Backup

On the "Configuration | Backup" page you define settings for the location and intervals of *CmDongle* data backups.



The screenshot shows the 'Configuration | Backup' page of the CodeMeter WebAdmin interface. At the top, there's a navigation bar with links for Home, Content, Server, Configuration (which is highlighted in blue), Diagnosis, Info, Help, and several system status links. The main content area is titled 'Backup'. It contains three configuration fields: 'Backup Path' with the value 'C:\Programme\CodeMeter\Backup', 'Backup Interval' set to '24 hours', and a checkbox for 'Certified Time' which is unchecked. Below these fields are two buttons: 'Apply' and 'Default'.

Figure 241: CodeMeter WebAdmin - "Configuration | Backup"

| Element | Description |
|------------------------|--|
| Backup Path | Specify in the Backup Path field the location where the backup file of the <i>CmDongle</i> 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 <i>CmDongle</i> at any time. |

| Element | Description |
|-----------------------|---|
| 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 " Default " button you save the default settings. |

12.4.4.8 Borrowing

On the "**Configuration | Borrowing**" page you define entry-specific settings of a borrowed license. These settings overwrite the original programmed settings of the borrowed license.

The screenshot shows the 'CodeMeter WebAdmin' interface with the 'Borrowing' tab selected. The main section is titled 'License Borrowing'. It includes a checkbox for 'Overwrite Entry Settings' which is checked. Below it are three input fields: 'Maximum Borrow Duration' (set to 'Minutes'), 'Maximum Borrow Quantity' (set to 'Licenses'), and 'Server Identification' (a dropdown menu set to 'Server Name'). At the bottom are two buttons: 'Apply' and 'Default'.

Figure 242: *CodeMeter WebAdmin - "Configuration | Borrowing"*

For setting the borrowing parameter, please proceed as follows:

1. Activate the option "**Overwrite Entry Settings**" to allow overwriting the original license condition of the borrowed license.
2. Specify in the "**Maximum Borrow Duration**" field the maximum time in minutes how long the license is to be borrowable.
3. Specify in the "**Maximum Borrow Quantity**" field the maximum number of borrowed licenses to be borrowed.
4. Select in the "**Server Identifizierung**" field how to identify the server either as Server Name or IP Address.
5. Click the "**Apply**" button to save the changes you have made. By a previous click on the "**Default**"

button you save the default settings.

12.4.5 License Display

CodeMeter WebAdmin displays information on [local](#)⁴⁵⁴ and [network licenses](#)⁴⁵⁹.

12.4.5.1 Local Licenses

The "Content | Licenses" page displays you all local licenses saved in a selected *CmContainer* or in all connected *CmContainer*.

Use the **CmContainer** dropdown control to select the desired or all *CmContainer*.

The screenshot shows the 'Content | Licenses' page of the CodeMeter WebAdmin interface. At the top, there's a navigation bar with links for Home, Content, Server, Configuration, Diagnosis, Info, Help, and a dropdown for CmContainer (set to 1-1440495). Below the navigation is a breadcrumb trail: CmContainer | Licenses | User Data | Backup/Restore. The main content area displays two tables of local licenses:

| 10 Vendor 1 | | | | | |
|---------------|------------------------|--------------|---------------------|-----------------|------------------|
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 10 | Word Processing | 200 | n/a | n/a | 10 |
| 13 | Calculation Processing | 400 | 2010-03-02 17:48:22 | n/a | 20 |
| 14 | Charts Processing | 200 | n/a | n/a | 23 |

| 228 Vendor 2 | | | | | |
|----------------|------------------|--------------|---------------------|---------------------|------------------|
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 67 | Print Processing | 1000 | n/a | 2009-03-02 17:48:45 | 50 |
| 1000 | Fax Add-on | n/a | 2009-04-13 11:50:05 | [borrowed] | 1 |

| 100003 Bundling Articles | | | | | |
|----------------------------|----------------|--------------|-----------------|-----------------|------------------|
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 1 | SecuriKey Lite | n/a | n/a | n/a | 1 |

Figure 243: CodeMeter WebAdmin - "Content | Licenses"

The display of local licenses is ordered by different licensors. A licensor is uniquely identified by number value, the Firm Code, and a name. For example, in the figure above this is the Firm Code "10" 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 the Test Firm Code is used, the following message text displays.
CodeMeter Evaluation License - not for commercial use!

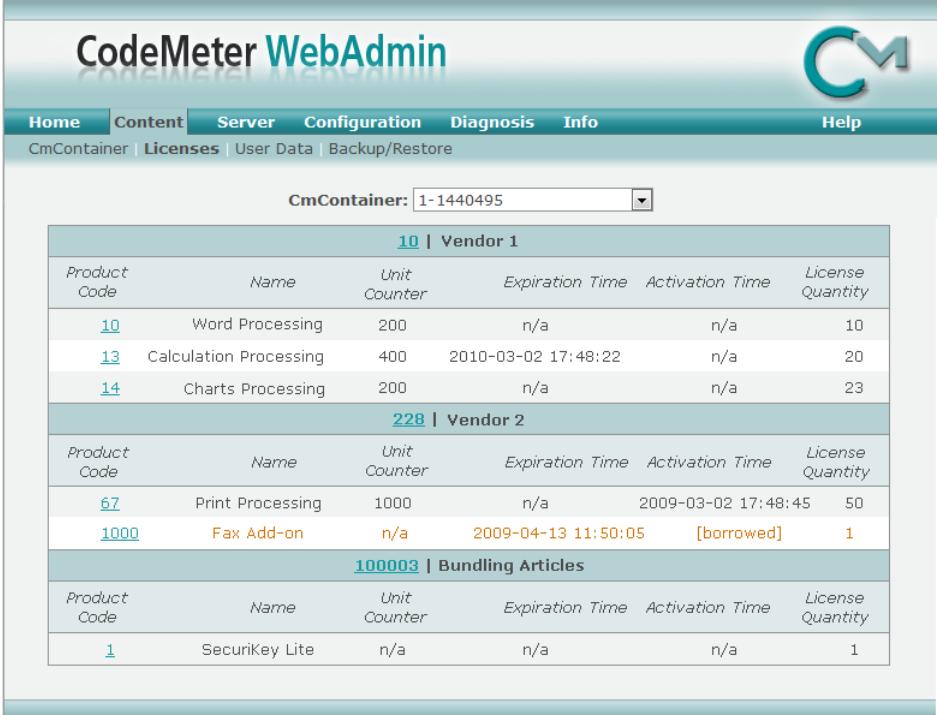
In the figure above, this is, first, the product "Print Processing" with the Product Code of 67, and, second, the product "Fax Add-on" with a Product Code of 1000 which is borrowed as a local license up to the end of the borrowing period. In addition, you obtain [further information](#)⁴³⁵ on existing Unit Counter, Expiration Time, Activation Time, and License Quantity.

Click on the highlighted [Firm Code](#)⁴⁴⁵ entry for the display of more detailed information on the license conditions of products by a specific vendor.

Click on the highlighted [Product Code](#)⁴⁴⁶, entry for the display of more detailed information on the license conditions of products by a specific vendor.

12.4.5.1.1 Licensor Information (ISV)

In the following figure you see all licenses provided by the licensor Vendor 1. Additional [information](#)⁴³⁵ comprise Product Code, CmContainer serial number, Name, Unit Counter, Activation Time, Expiration Time, License Quantity, and Feature Map.



The screenshot shows the CodeMeter WebAdmin interface with the following data:

| 10 Vendor 1 | | | | | |
|----------------------------|------------------------|--------------|---------------------|---------------------|------------------|
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 10 | Word Processing | 200 | n/a | n/a | 10 |
| 13 | Calculation Processing | 400 | 2010-03-02 17:48:22 | n/a | 20 |
| 14 | Charts Processing | 200 | n/a | n/a | 23 |
| 228 Vendor 2 | | | | | |
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 67 | Print Processing | 1000 | n/a | 2009-03-02 17:48:45 | 50 |
| 1000 | Fax Add-on | n/a | 2009-04-13 11:50:05 | [borrowed] | 1 |
| 100003 Bundling Articles | | | | | |
| Product Code | Name | Unit Counter | Expiration Time | Activation Time | License Quantity |
| 1 | SecuriKey Lite | n/a | n/a | n/a | 1 |

Figure 244: CodeMeter WebAdmin - "Content | Licenses - Firm Item"

455

12.4.5.1.2 Product Information

In the following figure you see all license information on the product with a Product Code "13" of the licensor at the Firm Item level with a Firm Code "10".

CodeMeter WebAdmin

Home
Content
Server
Configuration
Diagnosis
Info
Help

CmContainer |
Licenses | User Data | Backup/Restore

Product Item Details

Product Item 228:1000 of CmContainer 1-1123634

| Product Item Option | Type | Size (Bytes) | Dependencies | Value |
|-------------------------|------|--------------|-----------------------|---|
| Text | | 22 | | Fax Add-on |
| Feature Map | | 4 | data, serial, counter | 0000 0000 0000 0000 0000 0000 0000 0000 (0x0) |
| Units | | 4 | data, serial, counter | 0 |
| Activation Time | | 4 | data, serial, counter | 2008-04-03 13:09:32 |
| Expiration Time | | 4 | data, serial, counter | 2009-09-13 13:09:32 |
| Usage Period | | 8 | data, serial, counter | 0 days - Activation Time n/a |
| License Quantity | | 4 | data, serial, counter | local |
| License Information | | 10 | | Borrowed license |
| User Data | | 10 | | 0x00 |
| Protected Data | | 20 | data, serial, counter | 0x00 |
| Disable Time | 130 | 64 | data, serial, counter | 2009-04-13 11:50:05 |
| Borrow Server | 131 | 20 | | 192.168.0.134 (1-1123622) |
| Borrow Server Entry | 130 | 64 | data, serial, counter | CodeMeter 228:1200 |
| Borrow SID | 130 | 64 | data, serial, counter | 0x0000000000000001 |
| Extended Protected Data | 0 | 30 | data, serial, counter | 0x00 |
| Hidden Data | 23 | 50 | data, serial, counter | <hidden> |
| Secret Data | 34 | 60 | data, serial, counter | <secret> |

Figure 245: CodeMeter WebAdmin - "Content | Licenses - Product Item"

| 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 ⁴³⁵ 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 <i>CmContainer</i> 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 <i>CmContainer</i> . |
| 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.

12.4.5.2 User Data

The "Content | User Data" page shows you detailed information on products (licenses) the owner of the *CmDongle* Password. This license container is identified by a number value of "0".

Use the **CmContainer** dropdown control to select the desired *CmDongle* in order to see "your" licenses. Navigation and entry structures are analog to the [display of local licenses](#)⁴⁶¹.

The screenshot shows the CodeMeter WebAdmin interface. At the top, there's a navigation bar with tabs: Home, Content (which is selected), Server, Configuration, Diagnosis, Info, and Help. Below the navigation bar, there are links: CmContainer | Licenses | User Data | Backup/Restore. A search bar labeled "CmContainer: 1-1440495" is present. The main content area has a title "0 | User Data". A table displays user data with the following columns: ProductCode, Name, Unit Counter, Expiration Time, Activation Time, License, and Quantity. There are two rows of data:

| ProductCode | Name | Unit Counter | Expiration Time | Activation Time | License | Quantity |
|----------------------|------|--------------|-----------------|-----------------|---------|----------|
| 1000 | - | n/a | n/a | n/a | 1 | 1 |

Figure 246: CodeMeter WebAdmin - "Content | User data"

12.4.6 License Display on the Network

The "Server | ..." pages show you information on existing network licenses and their current allocation.



Network licenses on a *CmContainer* are addressable by other PCs only, if *CodeMeter License Server* has been started as network server.

The display of network licenses is divided in two categories:

- ordered by licensor and licenses (cluster).
- ordered by users of licenses (user).

12.4.6.1 Cluster - Licenses summarized

The "Server | Cluster" page shows all existing network licenses and their allocation ordered by licensors and related licenses.

The screenshot displays the 'Available Network Licenses at 'fs1.wibu.local'' section of the CodeMeter WebAdmin interface. The table has the following columns: Product Code, Name, Feature Map, Licenses, Status (User Limit (Borrowed), No User Limit, Exclusive, Shared, Free), and Details. The data is grouped by vendor and article number.

| Product Code | Name | Feature Map | Licenses | User Limit (Borrowed) | No User Limit | Exclusive | Shared | Free | |
|-----------------------------------|------------------------|-------------|----------|-----------------------|---------------|-----------|--------|------|-------------------------|
| 10 Vendor 1 | | | | | | | | | |
| 10 | Word Processing | 0x2 | 10 | 0 (-) | 1 | 1 | 0 | 9 | Details |
| 13 | Chart Processing | - | 20 | 5 (-) | 3 | 0 | 1 | 14 | Details |
| 14 | Print Processing | 0x6 | 23 | 2 (1) | 2 | 0 | 1 | 20 | Details |
| 228 Vendor 2 | | | | | | | | | |
| 67 | Calculation Processing | 0x8 | 50 | 1 (-) | 0 | 0 | 0 | 49 | Details |
| 100003 Bundling Articles | | | | | | | | | |
| 1 | SecuriKey Lite | 0x1 | 1 | 0 (-) | 0 | 0 | 0 | 1 | Details |

Information last updated on 16.Jun.2009 10:22:48

Figure 247: CodeMeter WebAdmin – "Server | Cluster"

Besides the describing information on **Product Code**, **Name**, and **Feature Map**, the column **Licenses** shows the respective total number of available network licenses.

Shared and Free Licenses

In addition, the **Status** group structures the licenses according to access modes (**User Limit**, **No User Limit**, **Exclusive**, **Shared**) and shows available **free licenses** ⁴³⁵.

Borrowed Licenses

Moreover, the **Status** group also shows you whether and if so which licenses in what quantity are locally borrowed from the license server.

In the figure above you see that of the total of 20 licenses of Vendor 1 for "Chart Processing" 14 licenses are free and available. Altogether 9 instances of the application access licenses but only 6 are counted since 3 accesses are of access status **No User Limit** allocating no additional licenses.

Click on the "[Details](#)" button to obtain detailed information on the license allocation.

12.4.6.1.1 Session Details

The following figure shows you detailed information on the license allocation.

| ID | Client | Client Process ID | Application Information | Access Mode | First Access | Last Access | Action |
|----|---------------|-------------------|-------------------------|---------------|---------------------|---------------------|------------------------|
| 27 | 192.168.0.134 | 760 | Charts | No User Limit | 2009-02-27 09:13:30 | 2009-03-03 08:11:40 | Cancel |
| 28 | 192.168.0.134 | 760 | Charts | No User Limit | 2009-03-02 08:12:23 | 2009-03-03 08:12:23 | Cancel |
| 43 | 192.168.0.134 | 760 | [1-1123634] | User Limit | 2009-04-02 15:39:22 | 2009-03-03 08:14:38 | Cancel |
| 44 | 192.168.0.134 | 760 | Charts | User Limit | 2009-02-27 08:17:05 | 2009-03-03 08:14:39 | Cancel |
| 45 | 192.168.0.134 | 760 | Charts | Station Share | 2009-02-27 08:14:48 | 2009-03-03 08:14:48 | Cancel |
| 55 | 192.168.0.134 | 760 | Charts | User Limit | 2009-02-27 08:14:48 | 2009-03-03 08:17:05 | Cancel |
| 57 | 192.168.0.33 | 764 | Charts | Exclusive | 2009-03-03 08:17:36 | 2009-03-03 08:17:36 | Cancel |

Information last updated on 27.Feb.2009 11:10:57

Figure 248: CodeMeter WebAdmin – "Server | Cluster - Details"

For example, in the figure above you see:

- the licenses for the application derive from the licensor with the Firm Code 10 and describe the product with the Product Code 14 as a module (0x6 as Feature Map).
- the licenses are stored in the **CmContainer** with the mask and serial number 1-1123634.
- in total 2 clients, identified by **ID**, **Client** (192.168.0.134 and 192.168.0.33) and **Client Process ID** columns, access the application "Chart Processing".
- 5 licenses of the 23 available licenses in total are allocated, 18 licenses are free.

- Client 192.168.0.33 exclusively allocates 1 license, client 192.168.0.134 uses the application in 6 instances but allocates only 4 licenses due to the **Access Mode** column.
- Client 192.168.0.134 has borrowed a license on the *CmContainer [1–1123634]* valid until April 12th, 2009.
- Client 192.168.0.33 for the first time accessed the application (**First** and **Last Access** columns are of same date).
- Client 192.168.0.134, according to the **First Access** column, previously accessed the application.

Cancel

Clicking the "**Cancel**" button 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.

12.4.6.2 Current User

The "Server | User" page shows you all existing network licenses ordered by users actually logged on (clients).

The screenshot shows the 'CodeMeter WebAdmin' interface with the 'Server' tab selected in the navigation bar. Below the navigation bar, there is a sub-menu with 'Cluster | User'. The main content area displays a table of license allocations:

| CmContainer | Firm Item | Product Item | Client | Access Mode | |
|-------------|---------------|---------------------------|---------------|---------------|-------------------------|
| 1-1123634 | 10: Vendor 1 | 10: Word Processing | 192.168.0.33 | No User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 10: Word Processing | 192.168.0.134 | Exclusive | Details |
| 1-1123634 | 10: Vendor 1 | 13: CalculationProcessing | 192.168.0.134 | No User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 13: CalculationProcessing | 192.168.0.134 | User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 13: CalculationProcessing | 192.168.0.134 | User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 13: CalculationProcessing | 192.168.0.134 | Station Share | Details |
| 1-1123634 | 10: Vendor 1 | 13: CalculationProcessing | 192.168.0.134 | Station Share | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | No User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | No User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | Station Share | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.134 | User Limit | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 192.168.0.33 | Exclusive | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 127.0.0.1 | Station Share | Details |
| 1-1123634 | 10: Vendor 1 | 14: Charts Processing | 127.0.0.1 | Station Share | Details |
| 1-1123634 | 228: Vendor 2 | 67: Print Processing | 192.168.0.33 | User Limit | Details |

Information last updated on 19.Dez.2011 11:35:56

Figure 249: CodeMeter WebAdmin - "Server | User"

Here you obtain all [describing information](#)⁴³⁵ on the **CmContainer**, licensor (**Firm Item**), license (**Product Item**), and **Access Mode**.

Session Details

Click on the "**Details**" button to obtain detailed information on the allocation of the license.

License Details CmContainer 1-1123622

| | |
|-------|---------------|
| Entry | 10 : 14 (0x6) |
| Free | 18 |
| Total | 23 |

| ID | Client | Client Process ID | Application Information | Access Mode | First Access | Last Access | Action |
|----|---------------|-------------------|-------------------------|---------------|---------------------|---------------------|------------------------|
| 27 | 192.168.0.134 | 760 | Charts | No User Limit | 2009-02-27 09:13:30 | 2009-03-03 08:11:40 | Cancel |
| 28 | 192.168.0.134 | 760 | Charts | No User Limit | 2009-03-02 08:12:23 | 2009-03-03 08:12:23 | Cancel |
| 43 | 192.168.0.134 | 760 | Charts | User Limit | 2009-03-03 08:14:38 | 2009-03-03 08:14:38 | Cancel |
| 44 | 192.168.0.134 | 760 | Charts | User Limit | 2009-03-03 08:14:39 | 2009-03-03 08:14:39 | Cancel |
| 45 | 192.168.0.134 | 760 | Charts | Station Share | 2009-03-03 08:14:48 | 2009-03-03 08:14:48 | Cancel |
| 55 | 192.168.0.134 | 760 | Charts | User Limit | 2009-03-03 08:17:05 | 2009-03-03 08:17:05 | Cancel |
| 57 | 192.168.0.33 | 764 | Charts | Exclusive | 2009-03-03 08:17:36 | 2009-03-03 08:17:36 | Cancel |

Information last updated on 27.Feb.2009 11:10:57

Figure 250: CodeMeter WebAdmin - "Server | User - Details"

Click the "[Details](#)"⁴⁶¹ button to obtain detailed information on the allocation of the license.

12.4.6.3 License Tracking

The "[Server | 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 | |
| Mac OS | /Library/Preferences/com.wibu.CodeMeter.Server.ini | |
| Linux | /etc/wibu/CodeMeter/Server.ini | |
| Solaris | /etc/opt/CodeMeter/Server.ini | |
| There exist two relevant profiling entries for <i>License Tracking</i> . | | |
| Entry | Property | Value |
| LogLicenseTracking | [DWord] [0 ; 1] | <p>i Default value is 0 and Logging for License Tracking is disabled.</p> |
| LogLicenseTrackingPath | [SZ] | <p><path></p> <p>i 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.</p> |
| i Please note that changed settings will take effect only after restarting <i>CodeMeter License Server</i> . | | |

On the basis of selectable 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 log file using the field "**Select logging period**".

Select logging period

2013-10-10T15:03 - 2013-10-10T15:06

Reload

Click the "**Reload**" button to update the logging period entries.

2. Select the license to be tracked using the field "**Select license**".

Select license**Create report**

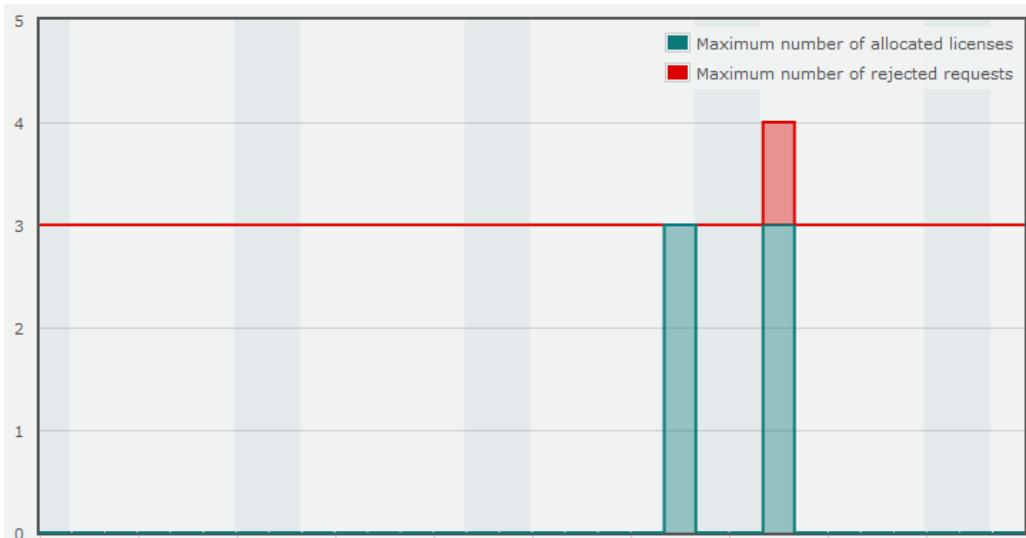
3. Click the button "**Create report**".

The separate area **Navigation**:

Navigation**View mode:** Hour**From:** 2013-09-20T10:00**To:** 2013-09-20T11:00

- 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.

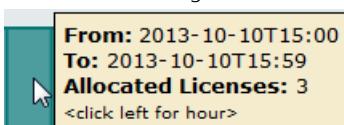
- Move over a colored bar to open an over-layered dialog for information display.



- 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.

- Move over a bar again to switch to the view mode **Hour**.



- 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. |

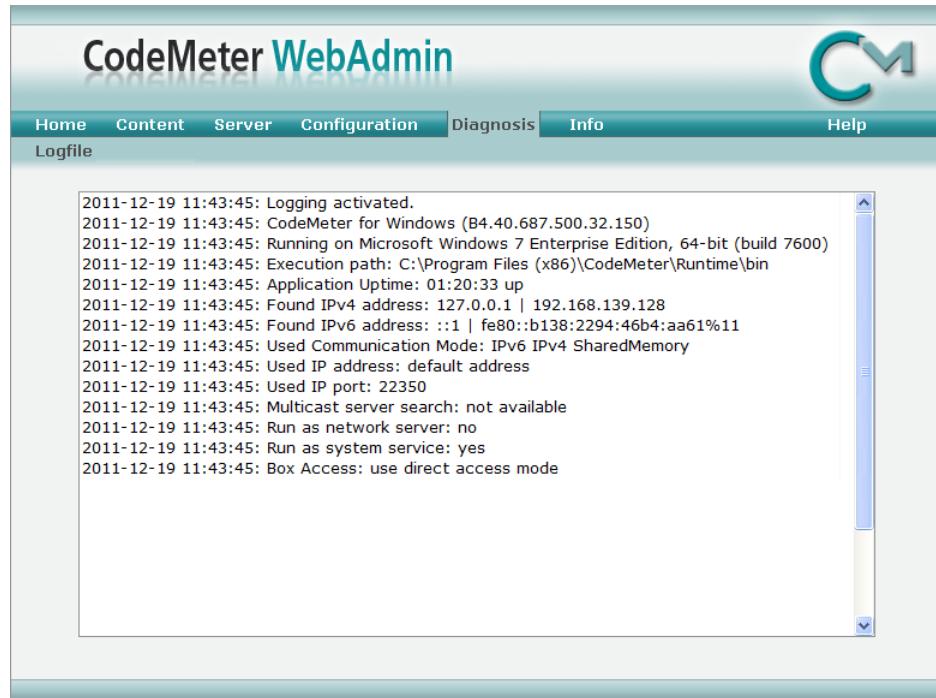
12.4.7 Diagnosis

LogFile

The "**Diagnosis | Logfile**" page allows you to log all processes related to the *CodeMeter License Server* service. This provides information which supports you in detecting eventually occurring errors.

 CodeMeter WebAdmin displays a protocol only if you previously [activated](#)⁴¹⁵ this function in CodeMeter Control Center.

There you find further information on how to save the log file.



The screenshot shows the 'CodeMeter WebAdmin' interface with a teal header bar containing 'Home', 'Content', 'Server', 'Configuration', 'Diagnosis' (which is highlighted in yellow), 'Info', and 'Help'. Below the header is a sub-menu with 'Logfile'. The main content area is titled 'Diagnosis | Logfile' and contains a scrollable text box displaying a log of system events from December 19, 2011, at 11:43:45. The log entries include details about the application's startup, network configuration, and access mode.

| |
|---|
| 2011-12-19 11:43:45: Logging activated. |
| 2011-12-19 11:43:45: CodeMeter for Windows (B4.40.687.500.32.150) |
| 2011-12-19 11:43:45: Running on Microsoft Windows 7 Enterprise Edition, 64-bit (build 7600) |
| 2011-12-19 11:43:45: Execution path: C:\Program Files (x86)\CodeMeter\Runtime\bin |
| 2011-12-19 11:43:45: Application Uptime: 01:20:33 up |
| 2011-12-19 11:43:45: Found IPv4 address: 127.0.0.1 192.168.139.128 |
| 2011-12-19 11:43:45: Found IPv6 address: ::1 fe80::b138:2294%46b4:aa61%11 |
| 2011-12-19 11:43:45: Used Communication Mode: IPv6 IPv4 SharedMemory |
| 2011-12-19 11:43:45: Used IP address: default address |
| 2011-12-19 11:43:45: Used IP port: 22350 |
| 2011-12-19 11:43:45: Multicast server search: not available |
| 2011-12-19 11:43:45: Run as network server: no |
| 2011-12-19 11:43:45: Run as system service: yes |
| 2011-12-19 11:43:45: Box Access: use direct access mode |

Figure 251: CodeMeter WebAdmin - "Diagnosis | Logfile"

12.4.8 Backup/Restore

The **"Content | Backup/Restore"** page allows you to save personal data located in your *CmDongle*, and restore them in the *CmDongle*.

 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".

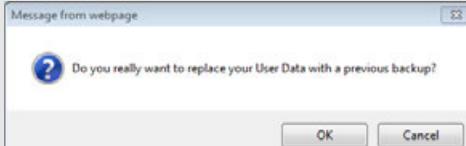
In order to restore licenses which do not locate in the personal area (Firm Item levels unequal to Firm Code "0"), please contact Wibu Support.

The screenshot shows the CodeMeter WebAdmin interface with the following details:

- Header:** Home, Content (selected), Server, Configuration, Diagnosis, Info, Help.
- Sub-Header:** CmContainer | Licenses | User Data | Backup/Restore.
- Form Fields:** CmContainer: 1-1440495 (dropdown).
- Text Area:** You can write all license data into a backup file. Includes a "Backup now" button and a "Last Backup: 2011-12-19 10:53:39" timestamp.
- Text Area:** You can restore your personal license data (including CM Password Manager) here. Includes a "Browse..." button and a "Restore" button.
- Information:** For information how to restore the license data inside a Firm Code not equal 0, contact our [Support](#).

Figure 252: CodeMeter WebAdmin - "Content | Backup/Restore"

| Element | Description |
|-------------------------|---|
| CmDongle | Click the CmDongle dropdown control to select the desired <i>CmDongle</i> for which the backup or restore is to apply. |
| Backup now | <ol style="list-style-type: none"> Click the "Backup now" button to apply an instant backup of your personal <i>CmDongle</i> data (user data). In addition, the time of the Last Backup is displayed. Confirm the following dialog to create the backup file.  |
| Browse / Restore | <ol style="list-style-type: none"> Click the "Browse" button to select the backup copy which is to be restored. The location of the backup file displays. Click the "Restore" button to start the restoring process. Confirm the following dialog and click the "OK" button. |



i 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.



i You are also able to import the saved data into another *CmDongle*. Please note, however, that the second *CmDongle* must have the same password !

12.4.9 Info

The "Info" page displays an overview of products and important Wibu-Systems addresses.

12.4.10 Help

The "Help" page can be reached from each page, and provides context-sensitive help on *CodeMeter WebAdmin*.

12.5 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* (**CodeMeter Enduser Support Tool**) 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

You open *CmDust* using the "Start | All Programs | CodeMeter | Tools" menu item. 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](#)⁴⁷⁵ to create a log file. For analyses this file can be sent to Wibu-Systems.

CmDust on Mac OS

For Mac OS you create the *CmDust* file using the [cmu](#)⁴⁷⁴ 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, Sun

For the operating systems Linux and Sun you create the *CmDust* file using the [cmu](#)⁴⁷⁴ 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 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"
```

12.6 CMU - 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 import of license request and update files (Remote

Context and Update files, *.WibuRaC; *.WibuRaU)

Call *cmu* in the directory %\Program Files%\CodeMeter\Runtime\bin using the command *cmu[32].exe*.

Alternatively, on Windows call *cmu* by the system menu item "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**".

For the operating systems Mac OS, Linux and Sun 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 | list the complete network license information. |
| /c <FI> or --context <FI> | creates a license request for an license update via CmFAS ⁴²⁷ 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 ⁴³¹ 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 --firm-code 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 . |

| Command | Description |
|-------------------------------|---|
| --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 <i>CmContainer</i> and the server name using the options --serial and --server . |
| --borrowlist | lists the borrowed licenses for the client and the server. |
| --enabling | lists the enabling stati of all connected <i>CmContainer</i> . Combined with the command -x you can also display additional enabling information of the <i>CmContainer</i> content. |
| --create-io | is used in combination with the option --file and makes sense only when using the hardware form factors <i>CmCard/SD</i> or <i>CmCard/CF</i> . A new <i>codemtr.io</i> file is created. Please call this command only if the <i>codemtr.io</i> file is deleted. |
| --detect-proxy | detects the proxy settings of the system. When options are omitted the standard output is used (stdout). The option --write saves the settings using the <i>CodeMeter®</i> profiling. |
| --delete-cmact-license | deletes a <i>CmActLicense</i> license you specify using the command --serial .  Once you delete a <i>CmActLicense</i> license it cannot be restored.. |

The following list shows all existing *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 <i>CmContainer</i> 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 <i>CodeMeter</i> 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 <i>CodeMeter®</i> profiling. These settings are written only if no proxy has been previously set in the profiling. For 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 <i>CodeMeter®</i> profiling. |
| --show-config-disk | Shows the current settings of removable/fixed drives or of the type of the defined Master Boot Record (MBR). |

| Options | Description |
|--|---|
| | This option concerns the behavior of virtual flash memory partitions. Use only for <i>CmStick</i> and <i>CmStick/M</i> . |
| --set-config-disk <parameter> | Allows to define a special behavior of virtual flash memory partitions, e.g. drive settings, boot code or activations (<i>CmDongle</i> only). |
| |  Please note that replugging of the <i>CmDongle</i> is required. |
| Description | Parameter |
| Drive settings | RemovableDisk,LocalDisk |
| Boot Code | Int18Boot,ZeroBoot,LoopBoot,SwapBoot,VbrBoot |
| Activation | ActivePartition,InactivePartition |
| FAT | |
| USB-Communication Device Class | HidCommunication; MsdCommunication |
| --check-cm-integrity | Allows to check the <i>CodeMeter®</i> signature. |

Application examples

| Action | Parameter |
|---|--|
| Displaying <i>cmu</i> options | <i>Cmu[32].exe -h</i> |
| Creating a <i>CodeMeter®</i> Remote Activation Context file (here:1-1040870.WibuCmRaC) for the Firm Code 10 (Firm Item level) | <i>Cmu[32].exe -c10 -f1-140870.WibuCmRaC</i> |
| Importing a <i>CodeMeter®</i> Remote Activation Update file (here:1-1040870.WibuCmRaU) --> reprograms the connected <i>CmContainer</i> | <i>Cmu[32].exe -i -f1-1040870.WibuCmRaU</i> |
| Showing the versions of current <i>CodeMeter®</i> components. | <i>cmu32 --version</i> |
| Listing all available <i>CodeMeter</i> network license server and if existing all related licenses. | <i>cmu32 --list-server --list-content</i> |
| Starting 100 simple tests. The tests are executed only for the <i>CmContainer</i> specified by the serial number of 1-233232. | <i>cmu32 --test 100 --serial 1-233232</i> |
| Changing the enabling status to "temporarily enabled" for the <i>CmContainer</i> 1-2345 by using the <i>CodeMeter®</i> password "SECRET". | <i>cmu32 --enable2 --serial 1-2345 --password SECRET</i> |

12.7 CodeMeter 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 are 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.

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](#)⁴⁷⁸
- [introduce definitions and value ranges used in the logfile](#)⁴⁷⁹
- [describe single logfile entry types](#)⁴⁸⁰

12.7.1 Requirements

Using the CodeMeter® feature License Tracking requires at least CodeMeter License Server Version 4.50.

12.7.2 Configuration

The logfile history needs to be enabled with CodeMeter License Server. This has to be done by activating it directly in the CodeMeter® profiling environment.

12.7.2.1 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 |
| Mac OS | /Library/Preferences/com.wibu.CodeMeter.Server.ini |
| Linux | /etc/wibu/CodeMeter/Server.ini |
| Solaris | /etc/opt/CodeMeter/Server.ini |

There exist two relevant profiling entries for *License Tracking*.

| Entry | Property | Value |
|------------------------|----------|--|
| LogLicenseTracking | [DWord] | [0;1]  Default value 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 changed settings will take effect only after restarting CodeMeter License Server.

LogFile Rotation

Currently, there is no logfile rotation implemented.



Currently, each time *CodeMeter License Server* is started, a new logfile with a timestamp is created and completed by respective licensing data.

12.7.3 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](#)
 480
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.

12.7.3.1 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 |
| borrow id | string The <borrow id> is derived as <mask>-<serial number>-<firm code>-<enabling block index>. All values refer to the borrowing client. |
| 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] |
| 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.\" "
```

12.7.4 Entry Types

The *CodeMeter®* license tracking logfile knows the following listed entry types.

12.7.4.1 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 |

12.7.4.2 License Entry

| | |
|--------------|---|
| Entry type | License entry |
| Description | The License entry describes an existing license. |
| Writing time | All License entries are written to the logfile: <ul style="list-style-type: none"> on startup of <i>CodeMeter License Server</i> each time when an entry is changed, e.g. by plugin / logout or remote programming. <p> In the cases mentioned above, all License entries of the current server are written preceded by a List of Licenses⁴⁸⁰ entry.</p> |
| 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 logout and the subsequent rebooking. Moreover, the access id may change by automatic rebooking after logout.

Licenses with a License Quantity value of 0 (license for local 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".

12.7.4.3 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 mluUserDefinedId and mszUserDefinedText.

12.7.4.4 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> |

12.7.4.5 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> |

12.7.4.6 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> |

12.7.4.7 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.

12.7.4.8 Administrative Entry

| | |
|---------------------|--|
| Entry type | Administrative entry |
| Description | An Administrative entry describes some event on the <i>CodeMeter License Server</i> . |
| Writing time | The Administrative entry is written at the moment the described event occurred. |
| Syntax | <timestamp> Admin Server:<server> "CodeMeter_started <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*.

12.8 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, Mac OS, and Linux are supported.

Alternatively to the Mass Storage Device status, thus *CmDongles* can display as HID without a drive status.



HID is currently available for all *CmDongles*, 1001-02-xxx (without FlashDisk).

Requirements

- *CmContainer* with the ID "2-xxxxxxxx" (Samsung chips)
- 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.

12.8.1 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.

The screenshot shows the 'Content | CmContainer' page of the CodeMeter WebAdmin interface. The 'CmContainer' dropdown is set to '2-2251132'. The 'Name' field is empty. The 'CmContainer Type' is 'CmStick/C 2.01'. The 'First Device' field is circled in red and contains 'E: (No Flash)'. The 'Status' section has three options: 'Disabled' (red square), 'Enabled until Unplugged' (yellow square), and 'Enabled' (green square), with 'Enabled' selected. Below the container details, there are fields for 'System Time (PC)', 'System Time (CmContainer)', 'Certified Time (CmContainer)', and 'Free Memory'. There are also 'Update' and 'Defragment' buttons.

2. Call [cmu](#)⁴⁷⁴.

For Windows OS call *cmu* by the system menu item "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**". For the operating systems Mac OS, Linux and Sun 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:

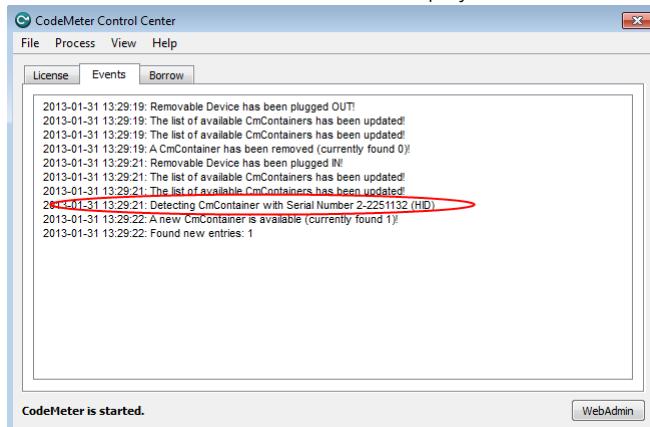
```
cmu32 - CodeMeter Universal Support Tool.
Version 5.00 of 2013-Jan-30 (Build 1039) for Win32
Copyright (C) 2007-2013 by WIBU-SYSTEMS AG. All rights reserved.
```

```
- 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.

The screenshot shows the 'Content | CmContainer' section of the CodeMeter WebAdmin interface. The 'First Device' field is highlighted with a red oval, displaying the value 'No drive assigned (HID)'. The 'Status' field has three radio button options: 'Disabled' (selected), 'Enabled until Unplugged', and 'Enabled'. Below the status section, there are system time details and memory usage information.

| | |
|-------------------------------|---|
| CmContainer: | 2-2251132 |
| Name: | <no name> |
| CmContainer Type: | CmStick/C 2.01 |
| First Device: | No drive assigned (HID) |
| Status: | <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled until Unplugged <input type="radio"/> Enabled |
| System Time (PC): | 2013-01-31 13:44:57 |
| System Time (CmContainer): | 2013-01-31 13:44:58 |
| Certified Time (CmContainer): | 2012-11-26 15:37:36 |
| Free Memory: | 94 % (367.696 Bytes) |

12.8.2 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.

The screenshot shows the 'Server' tab of the CodeMeter WebAdmin interface. A red oval highlights the 'First Device' field, which contains the text 'No drive assigned (HID)'. Other visible fields include 'CmContainer' (2-2251132), 'Name' (<no name>), 'CmContainer Type' (CmStick/C 2.01), 'Status' (Enabled), and various system and memory status indicators.

2. Call [cmu](#)

For Windows OS call `cmu` by the system menu item "**Start | All Programs | CodeMeter | Tools | CodeMeter Command Prompt**". For the operating systems Mac OS, Linux and Sun 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.
```

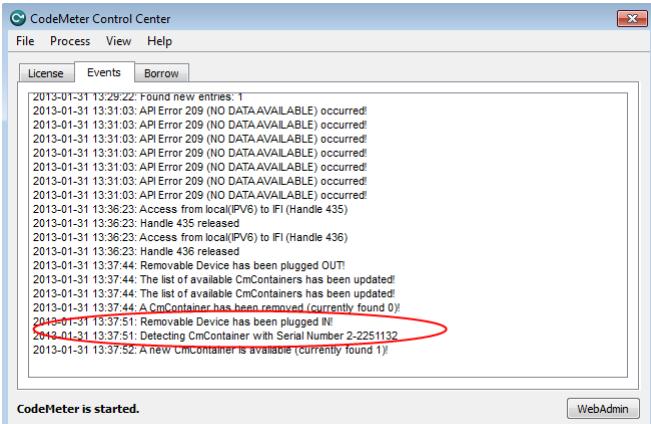
```
- CmStick/C with Serial Number 2-2251132 and version 2.01
Version: 2.01
Flash Size: no real flash available
Virtual Drive: No drive assigned (HID)
Communication: Human Interface Device (HID)
```

```
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.

The screenshot shows the 'CodeMeter WebAdmin' interface with the 'Content | CmContainer' tab selected. The page displays information for a CmContainer named '2-2251132'. The 'First Device' field is set to 'E: (No Flash)' and is circled in red. The 'Status' section shows three radio button options: 'Disabled' (red), 'Enabled until Unplugged' (yellow), and 'Enabled' (green), with 'Enabled' being the selected option.

| CmContainer: | 2-2251132 |
|-------------------------------|---|
| Name: | <no name> |
| CmContainer Type: | CmStick/C 2.01 |
| First Device: | E: (No Flash) |
| Status: | <input checked="" type="radio"/> Enabled <input type="radio"/> Enabled until Unplugged <input type="radio"/> Disabled |
| System Time (PC): | 2013-01-31 13:42:30 |
| System Time (CmContainer): | 2013-01-31 13:42:31 |
| Certified Time (CmContainer): | 2012-11-26 15:37:36 |
| Free Memory: | 94 % (367,696 Bytes) |

Buttons for 'Update' and 'Defragment' are located at the bottom right of the table.

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 <i>CmContainer</i> if crack attempts are detected. Available for different project types and as a commandline version. |
| CmActLicense | Completely software-based variant of the protection and licensing technology <i>CodeMeter</i> ®. 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 <i>CmContainer</i> . You can also apply scripts and batch files for mass production and test automation. Programming is simultaneously applied in one pass to several <i>CmContainer</i> . |
| CmContainer | Summarizing notion for describing the license carriers of both <i>CodeMeter</i> ® variants. <i>CmDongle</i> in the case of the hardware-based licensing system and <i>CmActLicense</i> in the case of the software-based licensing system. |
| CmDongle | Hardware-based variant of the protection and licensing technology <i>CodeMeter</i> ®. Available in many form factors for a variety of interfaces. |
| CmDust | The <i>CodeMeter</i> ® Enduser Support Tool logs important system and <i>CodeMeter</i> ® settings and helps Wibu-Systems Support to find remedies for eventually occurring errors. |
| CmFAS | see <i>CodeMeter Field Activation Service</i> |
| cmu | Commandline alternative to perform many <i>CodeMeter Control Center</i> functions (<i>CodeMeter Universal Support Tool</i>). |
| <i>CodeMeter API Guide</i> | 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. |
| <i>CodeMeter Field Activation Service</i> | see <i>File-based Remote Programming</i> |
| <i>CodeMeter Control Center</i> | <i>CodeMeter Control Center</i> provides the protected software to access the <i>CodeMeter</i> ® runtime environment. It displays information on connected <i>CmContainer</i> , and presents options to configure connected <i>CmContainer</i> . Moreover, an assistant creates license request files and imports license update files (<i>CmFAS Assistant</i>). |
| <i>CodeMeter License Central</i> | Ticket-based system for creating, managing, and delivering licenses for software and digital content. Available in a <i>Desktop</i> and an <i>Internet</i> Edition. |
| <i>CodeMeter License Editor</i> | Graphical tool allowing you to create, edit or delete licenses and their components (Firm Item, Product Item, and Product Item Options) in <i>CmDongles</i> . Next to programming of locally connected <i>CmDongles</i> also file-based remote programming (<i>CodeMeter Field Activation Service</i> , <i>CmFAS</i>) is supported. Suitable for testing license strategies. |
| <i>CodeMeter License Server</i> | Runtime environment (<i>CodeMeter.exe</i>) for the protection and licensing technology <i>CodeMeter</i> ®. |
| <i>CodeMeter Start Center</i> | Start screen tool to access and open most of the <i>CodeMeter</i> ® applications and tools. |
| <i>CodeMeter WebAdmin</i> | Graphical <i>CodeMeter</i> ® tool displaying information on connected <i>CmContainer</i> and related license entries in a browser. In addition, configuration and analyzing options for the <i>CodeMeter</i> ® runtime environment (<i>CodeMeter License Server</i>) are provided. |

| Term | Description |
|------------------------------------|---|
| <i>CodeMeter®</i> | Wibu-Systems' technology for protecting and licensing of software and digital content. |
| File-based Remote Programming | Remote updating a <i>CmContainer</i> requires some information on the <i>CmContainer</i> to be reprogrammed. This information is safely stored and transferred in a context file, i.e. *.WibuRaC file (license request file). Based on this license request file use the <i>CodeMeter®</i> programming tools to create an update file (*.WibuRaU) (license update). Subsequently, this file is safely transferred into the <i>CmContainer</i> . In addition, on creating the *.WibuRaU file automatically also a *.WibuRaM file is created which maps the <i>CmContainer</i> content at the time the licenses have been updated. An CmFAS Assistant in <i>CodeMeter Control Center</i> supports the licensee when updating licenses. |
| Enabling | Procedure to directly activate or deactivate the complete but also single Firm Item levels and license entries of the <i>CmContainer</i> by using an access code. |
| 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 <i>CmContainer</i> . 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 <i>CmDongle</i> which allows to program other <i>CmContainer</i> . The FSB is unique for each licensor. |
| FSB | see Firm Security Box |
| HIP | <i>High Level Programming API</i> see <i>Programming API</i> |
| IFI | see Implicit Firm Item |
| Implicit Firm Item | The Implicit Firm Item level in the <i>CmContainer</i> 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 <i>CmContainer</i> has licensor privileges for the Implicit Firm Item level including write access. |
| <i>IxProtector</i> | Individual advanced protection technology applied for software and digital content. 'Real' source code fragments are encrypted and decrypted by interfaces (<i>Software Protection API</i> , <i>WUPI</i>) and security mechanisms. Suited to implement modular software protection. |
| <i>Core API</i> | Powerful interface to communicate with <i>CmContainer</i> at runtime of <i>CodeMeter License Server</i> . All other APIs and protection mechanisms (<i>AxProtector</i> , <i>IxProtector</i> , <i>Software Protection API</i> <i>WUPI</i>) base on <i>Core API</i> 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 Update File (*.WibuCmRaU) | The update file for a <i>CmContainer</i> valid only for a single unique <i>CmContainer</i> can be imported only once. |
| License Request File (*.WibuCmRaC) | The context file of a <i>CmContainer</i> mirroring the as-is status of license entries serves as basis for license updating in the process of file-based (remote) programming. |
| <i>SmartBind</i> | Binding scheme used in <i>CmActLicense</i> licensing system optimizes assuring the validity of <i>CmActLicense</i> licenses, if hardware properties of the PC change to which the licenses are |

| Term | Description |
|--|---|
| 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) | <p>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.</p> <p>These optional characteristics are combinable in a variety of ways, and constitute the basis for the mapping of any imaginable license strategy.</p> |
| Product Item | Logical hierarchical entry level in a <i>CmContainer</i> 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. |
| <i>Programming API</i> | This class-oriented interface allows you to access any object or process required to program or organize license entries in a <i>CmContainer</i> , and features extended customizing for the integration of <i>CodeMeter</i> into own applications. The <i>Programming API</i> is available for many programming languages |
| <i>Software Protection API</i> | Interface which decrypts segments protected by <i>IxProtector</i> 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. |
| <i>Wibu Universal Protection Interface</i> | see <i>Software Protection API</i> |
| WUPI | see <i>Wibu Universal Protection Interface</i> |

Index

- * -

*.lif (license information file) 29

- A -

Activation Time

Product Item Option 40

ASM program library

AxProtector Java 13

Copyright 13

AxProtector .NET

.NET Options 121

Activate Hardware Locking 117

Activate IxProtector 122

Activate Plug-out Check 111

Activate Runtime Check 111

Activation Time 114

Advanced Options 122

Also at Runtime Check 112

Automatic Method Encryption 118

Automatic Trap Generation 117

Basic Debugger Check 117

Check Certified Time 115

CmContainer / PC System Time check 115

Create Logfile 122

Create mobile application 116

Customized Error Messages 120

Decrement by 112

Default Error Messages 119

Destination File 106

Encryption Algorithm 108

Encryption Time check 115

Error Messages 119, 120

Exclusive Mode 110

Expiration Time 114

Extended Commandline Options 122

Feature Code 108

File To Protect 106

Firm Code 107

Inline Messages (.NET only) 120

IxProtector Bytes 128

IxProtector Methods 128

IxProtector Name 127

IxProtector Views 127

License access lock (Configuration) 117

License Handling 110

License List Feature Code 124

License List Firm Code 124

License List Product Code 124

License List Subsystem 125

License List Description 124

License List Id 124

License List Ignore Linger Time 125

License List Licensing Systems 124

License List Minimum Driver Version 125

License List Minimum Firmware 125

License List Release Date 125

License List WupiReadData 125

License List WupiWriteData 125

License Option 110

Licensing Systems 107, 108

Linger Time 110

Logging 122

Maintenance Period 114

Max. Allowed Ignores 111

Maximum Certified Time Age 115

Minimum Driver Version 108

Minimum Firmware 108

Minutes to be allowed older 115

Minutes to be allowed younger 115

No Strong Name 121

No User Limit 110

Normal User Limit 110

Obfuscation 116, 282, 283

Optimization 122

Period 111

Period without time checking 115

Probing 121

Product Code 108

Reflector defence 117

Release Date 108

Runtime Settings 111, 112

Runtime Settings, advanced 113, 114, 115, 116

Security Options 117

Set Certified Time 115

Source File 106

Station Share 110

Strong Name from Container 121

Strong Name from File 121

Subsystem Local 110

Subsystem Network 110

Summary Back 129

Summary Finish 129

Summary Protect now 130

Terminate host application 116

Thresholds Expiration Time 112

Thresholds Unit Counter 112

Unit Counter 112, 113

User Message DLL 119

WibuKey Compatibility Mode 110

- AxProtector Commandline
-I (Create *.wbc file) 295
-# (Logging) 295
-? -h (Options and help) 295
-@cmds.wbc (Parameter in executable file) 295
-A (Insert security code) 270
-A[AES] (Encryption algorithm) 271
-ANF (Message if assembly not found, .NET) 289
-CAA (Security options) 274
-CACT (CmContainer SystemTime) 274
-CAD (File Access Mode) 274, 275
-CAE (Plug-Out) 275
-CAG (Anti-Debugging .NET) 276, 277
-CAG (Anti-Debugging Java) 277
-CAG (Anti-Debugging) 275, 276
-CAL (areas of encryption) 277
-CAM (System Menu) 277
-CAR (Runtime Check) 277
-CAS (Percentage Encrypted) 278
-CAT (Certified and System Time) 278
-CAV (Virus check) 278
-CAZ (Encryption Time) 278
-CCA (Target platform and option -CCX) 278
-CCB (.reloc Section translation) 279
-CCC (ActiveX / OCX Images) 279
-CCD (Shared Objects options) 280
-CCE (PE is not enlarged) 279
-CCH (Preventing global hooking) 279
-CCI (Load sequence) 279
-CCK (Shared Objects unload) 280
-CCM (Load sequence wibucrt.dll) 279
-CCQ (Clear licenses on Exit Process) 279
-CCR (Deactivation renaming of Sections) 279
-CCS (Licenses from first connected CmContainer) 279, 280
-CCX (Mixed Mode Assemblies) 280
-CDC (File name extension for file encrypting) 280
-CDH (Access to license file encryption) 280
-CDK (Licensing system file encryption) 281
-CFx (Feature Code) 272
-CI (IxProtector) 281, 282
-CID (IxProtector) 281
-CIH (WUPI and Hooking) 281
-CIN (No error messages IxProtector) 281
-CK (RID key cache, .NET) 282
-CML (Min. size methods, .NET) 283
-CO (Obfuscation) 282, 283
-CP (Cleanup mechanism Windows) 283
-CPA (Encrypting property accessors, .NET) 283
-D (Driver version) 272
-EA (Activation Time check) 283
-EC (MSIL Code class construction, .NET) 283
-EE (Expiration Time check) 284
-EF (Decrement Firm Access Counter) 284
-EM (Maintenance Period required) 284
-ET (Enforce Certified Time update) 284
-EU (Unit Counter check) 284, 285
-EXTRACT (Assembly printout, .NET) 295
-FW (Firmware Version) 272
-FW (minim. Firmware on encryption) 285
-Fx (Firm Code) 271
-G (Exclude areas from encryption) 285
-I (Exception handling Plugin DLLs) 286
-ja (Main Class arguments runtime Java) 290
-jb (Exception handling Java) 290
-jcl (Class Loader) 290
-jd (Min.- Max. Version Java) 290
-jff:[c|w] (external Class files Java) 294
-jh (Hide and rename of classes Java) 290
-jl (White/ blacklist classes Java) 291
-jm (Starting Main Class Java) 290
-jn (Class loading Java) 291
-jo (Output options *.jar Java) 291
-jpc (define additional JAR files Java) 295
-jvm (Option Virtual Machine Java) 292
-jx (Exit application Java) 292
-K (Licensing System) 271
-L (Language message texts) 286
-M (Output texts for message texts) 287
-N (Network access mode) 273
-O (Name and path of encrypted target file) 289
-prio (set process priority, Windows only) 289
-PROBING (Path information when assembly found, .NET) 289
-Px Product Code) 271
-RD (Release Date) 272
-RID (Number RID variants) 285
-RID (Number of RID and Trap variants IxProtector) 285
-S (Search order for licenses) 272
-Silverlight (Encrypting) 286
-SNK (Strong Name Key, .NET) 289
-SW (Search order for licenses, WAN) 272
Syntax 270
-TRAP (Hacker traps, .NET) 289
-U (Calling message DLL) 288
-UI (Inline message assembly, .NET) 289
-UM (Call message assembly, .NET) 288
-V (Verbose mode) 295
-WC (Threshold Certified Time) 285
-WE (Threshold Expiration Time) 285
-WP (Threshold Usage Period) 286

- AxProtector Commandline
 -WU (Threshold Unit Counter) 286
 -X (Static linking) 270
 -XAP (XAP file for Silverlight) 286
- AxProtector File Encryption
 Advanced Options 260
 Create Logfile 260
 Destination File 255
 Encryption algorithm 257
 Exclusive Mode 259
 Extended Commandline Options 260
 Feature Code 257
 File Encryption Extension 265
 File Encryption File Access Mode 266
 File Encryption Name 265
 File Encryption Player Check 266
 File Encryption Writing existing file 267
 File Encryption Writing New File 267
 File To Protect 255
 Firm Code 256
 License Handling 258, 259
 License List Licensing Systems 263
 License List Description 262
 License List Feature Code 262
 License List Firm Code 262
 License List Id 261
 License List Ignore Linger Time 263
 License List Licensing Systems 262
 License List Minimum Firmware 263
 License List Product Code 262
 License List Release Date 263
 License List Subsystem 262
 License List WupiReadData 263
 License List WupiWriteData 263
 License Option 259
 Licensing Systems 256, 257
 Linger Time 259
 Logging 260
 Minimum Driver Version 257
 Minimum Firmware 257
 No User Limit 259
 Normal User Limit 259
 Product Code 256
 Release Date 257
 Source file 255
 Station Share 259
 Subsystem Local 258
 Subsystem Network 258
 Summary Back 268
 Summary Finish 268
- Summary Protect Now 269
 WibuKey Compatibility Mode 259
- AxProtector Java
 Activate Runtime Check 162
 Activation Time 165
 Advanced Options 171
 Blacklist 170
 Callback Manipulation Check 167
 Check Certified Time 166
 Class Name 168
 Classes to encrypt 170
 CmContainer / PC System Time check 166
 Create Logfile 171
 Customized Error Messages 168
 Default Error Messages 168
 Destination File 157
 Encryption Algorithm 159
 Encryption Time check 166
 Error Messages 168
 Exclusive Mode 161
 Expiration Time 165
 Extended Commandline Options 171
 Feature Code 158
 File To Protect 157
 Firm Code 158
 IxProtector 171, 292
 JVMPi Detection 167
 License Handling 160, 161
 License Option 161
 Licensing Systems 158, 159
 Linger Time 161
 Logging 171
 Maintenance Period 165
 Max. Allowed Ignores 162
 Maximum Certified Time Age 166
 Method encryption 292
 Minimum Driver Versions 159
 Minimum Firmware 159
 Minutes to be allowed older 166
 Minutes to be allowed younger 166
 No User Limit 161
 Normal User Limit 161
 Options Call System.exit() 169
 Options Java Runtime 169
 Options main class 169
 Options Minimum Java Version 169
 Options Split Output 170
 Parameter main class 169
 Period 162
 Period without time checking 166

| | |
|------------------------------------|---------------|
| AxProtector Java | |
| Product Code | 158 |
| Release Date | 159 |
| Rename encrypted classes | 170 |
| Runtime Settings | 162, 163 |
| Runtime Settings, advanced | 164, 165, 166 |
| Security Options | 167 |
| Set Certified Time | 166 |
| Source File | 157 |
| Station Share | 161 |
| Subsystem Local | 160 |
| Subsystem Network | 160 |
| Summary Back | 172 |
| Summary Finish | 172 |
| Summary Protect Now | 173 |
| Threshold Unit Counter | 163 |
| Thresholds Expiration Time | 163 |
| Unit Counter | 164 |
| Unit Counter Also at Runtime Check | 163 |
| Unit Counter Decrement by | 163 |
| User Message Class | 168 |
| User Messages | 168 |
| VM Verification | 167 |
| Whitelist | 170 |
| WibuKey Compatibility Mode | 161 |
| AxProtector Linux | |
| Activate Ixprotector / WUPI | 189 |
| Activate Runtime Check | 180 |
| Add virus check | 187 |
| Advanced Options | 189 |
| CmContainer / PC System Time check | 184 |
| Create Logfile | 189 |
| Customized Error Messages | 188 |
| Default Error Messages | 188 |
| Destination File | 175 |
| Encryption Time check | 184 |
| Error Messages | 188 |
| Exclusive Mode | 179 |
| Extended Commandline Options | 189 |
| Feature Code | 176 |
| File to protect | 174, 175 |
| Firm Code | 176 |
| IxProtector Function Description | 195 |
| IxProtector Function Id | 195 |
| IxProtector Function Length | 195 |
| IxProtector Function License List | 195 |
| IxProtector Function Name | 195 |
| IxProtector Function Trap | 195 |
| License Handling | 178, 179 |
| License List Minimum Firmware | 192 |
| License List Release Date | 192 |
| License List Id | 190 |
| License List Ignore Linger Time | 192 |
| License List WupiReadData | 192 |
| License List WupiWriteData | 192 |
| License Option | 179 |
| Licensing Systems | 176, 177 |
| Linger Time | 179 |
| Link API statically to Application | 187 |
| Logging | 189 |
| Maintenance Period | 183 |
| Minimum Driver Version | 176 |
| Minimum Firmware | 177 |
| Minutes to be allowed older | 184 |
| Minutes to be allowed younger | 184 |
| No User Limit | 179 |
| Normal User Limit | 179 |
| Period without time checking | 184 |
| Product Code | 176 |
| Release Date | 177 |
| Runtime Settings | 180, 181 |
| Runtime Settings, advanced | 183, 184 |
| Security Option, advanced | 187, 188 |
| Size of encrypted Code (in %) | 188 |
| Source file | 174 |
| Station Share | 179 |
| Subsystem Local | 178 |
| Subsystem Network | 178 |
| Summary Back | 197 |
| Summary Finish | 197 |
| Summary Protect Now | 198 |
| Threshold Expiration Time | 181 |
| Threshold Unit Counter | 181 |
| WibuKey Compatibility Mode | 179 |
| AxProtector MAC OS | |
| Activate Ixprotector / WUPI | 145 |
| Activate license access lock | 142, 186 |
| Activate Plug-out Check | 136, 181 |
| Activate Runtime Check | 136 |
| Activation Time | 139, 183 |
| Add virus check | 144 |
| Advanced Debugger Check | 142, 185 |
| Advanced Options | 145 |
| Basic Debugger Check | 142, 185 |
| Check Certified Time | 140, 184 |
| CmContainer / PC System Time check | 140 |
| Create Logfile | 145 |
| Customized Error Messages | 141 |
| Default Error Messages | 141 |
| Destination File | 132 |

- AxProtector MAC OS
 Encryption Algorithm 133, 176
 Encryption Time check 140
 Error Messages 141
 Exclusive Mode 135
 Expiration Time 139, 183
 Extended Commandline Options 145
 Feature Code 133
 File to protect 131, 132
 Firm Code 133
 IxProtector Function Description 151
 IxProtector Function Id 151
 IxProtector Function Length 151
 IxProtector Function License List 151
 IxProtector Function Name 151
 IxProtector Function Trap 151
 License access lock (Configuration) 143, 186
 License Handling 135
 License List Description 147, 191
 License List Feature Code 147, 191
 License List Firm Code 147, 191
 License List Licensing Systems 147, 191
 License List Minimum Driver Version 148, 192
 License List Minimum Firmware 148
 License List Product Code 147, 191
 License List Release Date 148
 License List Id 146
 License List Ignore Linger Time 148
 License List Subsystem 148, 192
 License List WupiReadData 148
 License List WupiWriteData 148
 License Option 135
 Licensing Systems 133, 134, 176
 Linger Time 135
 Link API statically to Application 144
 Logging 145
 Maintenance Period 139
 Max. Allowed Ignores 136, 180
 Maximum Certified Time Age 140, 184
 Minimum Driver Version 133
 Minimum Firmware 134
 Minutes to be allowed older 140
 Minutes to be allowed younger 140
 No User Limit 135
 Normal User Limit 135
 Period 136, 180
 Period without time checking 140
 Product Code 133
 Release Date 134
 Runtime Settings 136, 137, 140, 180, 181, 184
 Runtime Settings, advanced 138, 139, 140, 182, 183, 184
 Security Option, advanced 144
 Security Options 142, 143, 185, 186
 Set Certified Time 140, 184
 Size of encrypted Code (in %) 144
 Source file 131
 Station Share 135
 Subsystem Local 135
 Subsystem Network 135
 Summary Back 153
 Summary Finish 153
 Summary Protect now 154
 Threshold Expiration Time 137
 Threshold Unit Counter 137
 Unit Counter 138, 182
 Unit Counter Also at Runtime Check 137, 181
 Unit Counter Decrement by 137, 181
 Virtual Machine Detection 142, 186
 WibuKey Compatibility Mode 135
- AxProtector Windows
 Activate Automatic File Encryption 91
 Activate IxProtector 91
 Activate runtime check 79
 Activates license access lock 86
 Activating logout check (CmDongle) 79
 Activation Time 82
 Add control and about menu 84
 Add Virus Check 88
 Advanced Options 91, 92
 Also at runtime check 80
 Basic Debugger Check 86
 Check Certified Time 83
 CmContainer / PC System Time check 83
 Create Logfile 92
 Create mobile application 84
 Customized Error Messages 90
 Decrement by 80
 Default Error Messages 89
 Destination file 74
 Dynamic Code Modification 85
 Dynamic loading of Wibu-Systems libraries 91
 Encryption Algorithm 76
 Encryption Time check 83
 Error Message 89
 Error Messages 89, 90
 Exclusive Mode 78
 Expiration Time 80, 82
 Extended Commandline Options 91
 Extended Static Modification 85
 Feature Code 76

AxProtector Windows
File Encryption Extension 100
File Encryption Player Check 100
File Encryption Writing Existing File 101
File Encryption Writing New File 102
File Encryption File Access Mode 101
File Encryption Name 100
File to protect 74
Firm Code 75
Generic Debugger Detection 86
IDE Debugger Check 86
Ignore Linger Time 78
Inline Messages (not available) 90
IxProtector Function Description 98
IxProtector Function Id 98
IxProtector Function Length 98
IxProtector Function License List 98
IxProtector Function Name 98
IxProtector Function Trap 98
Kernel Debugger Check 86
License access lock (Configuration) 86
License Handling 77, 78
License List Id 93
License List Description 93
License List Feature Code 94
License List Firm Code 94
License List Ignore Linger Time 95
License List Licensing Systems 94
License List Minimum Driver Version 95
License List Minimum Firmware 95
License List Product Code 94
License List Release Date 95
License List Subsystem 95
License List WupiReadData 95
License List WupiWriteData 95
License Option 78
Licensing Systems 75, 76
Link API statically to Application 88
Logging 92
Maintenance Period 82
Max. Allowed Ignores 79
Maximum Certified Time Age (hours) 83
Minimum Driver Version 76
Minimum Firmware 76
Minutes to be allowed older 83
Minutes to be allowed younger 83
No User Limit 78
Normal User Limit 78
Period 79
Period without time checking (hours) 83

Product Code 75
Release Date 76
Resource Encryption 85
Runtime Settings 79, 80
Runtime Settings, advanced 81, 82, 83, 84
Security Options 85, 86
Security Options, advanced 88
Set Certified Time 83
Size of encrypted Code (in %) 88
Source file 74
Static Code Modification 85
Station Share 78
Subsystem Local 77
Subsystem Network 77
Summary Back 103
Summary Finish 103
Summary Protect Now 104
Suppress IxProtector Error Messages 89
Terminate host application 84
Threshold 80
Unit Counter 80, 81
User Message DLL 89
Virtual Machine Detection 86
WibuKey Compatibility Mode 78
AxProtector-Kommandozeile
-CMD(n) (Wiederverschlüsseln von Methoden, .NET) 283
- B -
Backup of CmDongle 398
Binding Extension 28
Binding scheme 28
- C -
Certified Time 394
update 442
CmActLicense
Activating licenses 29, 408
additional activation options 29
Binding Extension 28
Binding scheme 28
None-Binding 28
Protection-Only license model 28
Smart Bind 345
SmartBind 28
SmartBind behaviour in VM 345
Trial License license model 28
CmActLicense license
Programming (CmBoxPgm) 348
CmBoxPgm 331
Activation by Phone CmActLicense 346
Activation Time -pat 337

CmBoxPgm 331
Activation Time, absolute -pata 337
Activation Time, relative -patr 337
Add / Update -cau 333
Add -ca 332
Attach Enabling Block -eatt 355
Backup File -bkp 357
Box Index -qb 334
Box Index Range -qnx 334
Box Passwort -pwd 334
Cleanup Registry -rcl 358
CmActLicense Activation Code -lac 344
CmActLicense Activation File -laf 344
CmActLicense Binding Value -lbind 344
CmActLicense in VM -lopt:vm 346
CmActLicense License ID -lpid 347
CmActLicense License Information File (phone) -lip 346
CmActLicense License Information File -lif 346
CmActLicense License Options -lopt 346
CmActLicense Name -lpn 347
CmActLicense reimport -lopt:reimport 347
CmActLicense Target Operating System -los 347
Create RaC File -crac 357
Customer Owned License Information -pcoli 338
Delete -cd 333
Delete if possible -cdx 333
Detach Enabling Block -edet 356
Disable Time -edt 356
Disable Time, absolute -edta 356
Disable Time, relative -edtr 357
Display CmActLicense Binding Scheme -lfs 345
Display CmActLicense Installation ID -ldi 345
Display CmActLicense License File -ldf 345
Enabling Access Code -eac 355
Enabling -e 355
Enabling Mode -em 357
Enabling Text -et 357
Expiration Time -pet 338
Expiration Time, absolute -peta 338
Expiration Time, relative -petr 339
Extended Protected Data -ped 338
Feature Map -pfm 339
File-based Activation CmActLicense 346
Firm Access Counter -fac 335
Firm Code -f 335
Firm Item Text -ft 336
Firm Key -fk 354
Firm Precise Time, absolute -fpta 335
Firm Precise Time, relative -fptr 335
Firm Update Counter -fuc 336
FSB Entry -fsb 354
Help -? 358
Hidden Data -phd 339
License Borrowing Client -blc 353
License Borrowing Server -bls 352
License Quantity -plq 340
Linger Time -plt 340
List -l 333
Logging -log 358
Maintenance Period -pmp 340
Maintenance Period, Date -pmpd 341
Maintenance Period, Integer-pmpi 341
Minimum required runtime (CmActLicense) 346
Network License Counter -pnwc 341
Product Code -p 337
Protected Data -ppd 341
Recursive Removal -r 334
Remote Activation -ra 358
Remote Activation Update -rau 334
Secret Data -psd 341
Sequence Dump -sqd 358
Serial number -qs 334
Smart Bind 345
Text -pt 342
Unit Counter -puc 342
Unit Counter, absolute -puca 342
Unit Counter, relative -pucr 343
Update -cu 333
Usage Period -pup 343
Usage Period, absolute -pupa 343
Usage Period, relative -pupr 344
User Data -pud 343
Validation Mode -val 359
Verbose Mode -v 359
WUPI Data -pwupidata 344
CmDongle
First connection 408
CmDust 472
CmFAS Assistant 425
CmFirm.wbc 18
CmLicense Editor 322
Display Window 326
Menu Bar 324
Output Window 326
Remote Programming 327
Structure and Navigation 323
Symbol Bar 324
Tree View 325
WibuCmRaC 327
WibuCmRaM 327

| | | | |
|------------------------------------|-----|-------------------------------|----------|
| CmLicense Editor | 322 | Functional Areas | 309 |
| WibuCmRaU | 327 | Management API | 311 |
| Working with | 327 | Programming API | 311 |
| CmStick/T | 396 | Remote Update API | 312 |
| cmu | | Time Management API | 312 |
| CodeMeter Universal Support Tool | 474 | CodeMeter FAQ | 410 |
| CmWAN | | CodeMeter License Central | 360 |
| AxProtector (encrypting) | 404 | Admin-Interface | 367 |
| CmBoxPgm (programming) | 401 | Application Scenarios | 367 |
| CodeMeter API Guide (accessing) | 401 | Architecture | 362 |
| CodeMeter WebAdmin (configuring) | 404 | Connectors | 364 |
| profiling | 404 | Depot-Interface | 366 |
| registry, server.ini (configuring) | 404 | Gateway | 365 |
| CodeMeter | | Principle | 360 |
| Concept | 34 | CodeMeter License Editor | |
| Form factors | 26 | Firm Code | 328 |
| Operating Systems | 30 | PIO Activation Time | 329 |
| Token | 32 | PIO Expiration Time | 329 |
| CodeMeter API Guide | 313 | PIO Feature Map | 329 |
| Blocks Tab | 316 | PIO License Quantity | 329 |
| Function Tab | 316 | PIO Maintenance Period | 330 |
| Handle Display Window | 317 | PIO Text | 329 |
| Interactive Area | 317 | PIO Unit Counter | 329 |
| Menu Bar | 315 | PIO Usage Period | 329 |
| Source Code Area | 317 | Product Code | 329 |
| Structure and Navigation | 314 | CodeMeter License Server | 65 |
| Tree View | 316 | Run CmWAN Server (WebAdmin) | 445 |
| WUPI Tab | 316 | Run Network Server (WebAdmin) | 445 |
| CodeMeter Compact Driver | 33 | CodeMeter License Tracking | 477 |
| CodeMeter Control Center | 411 | Access Entry | 481 |
| Activation status | 414 | Administrative Entry | 482 |
| Borrowing Tab | 421 | Borrow Access Entry | 481 |
| Certified Time Update | 416 | Borrow Return Entry | 482 |
| CmDongle register | 417 | Configuration | 478 |
| Event Tabs | 421 | Denial Entry | 482 |
| Firmware Update | 418 | License Entry | 480 |
| License import | 414 | List of Licenses Entry | 480 |
| License Tab | 417 | LogFile Format | 479 |
| Logging, activate | 415 | Release Entry | 481 |
| Menu Bar | 414 | Requirements | 478 |
| Start CodeMeter Service | 416 | CodeMeter on Embedded Systems | |
| Status and Open | 424 | Compact Driver | 33 |
| Stucture and Navigation | 413 | CodeMeter Sample Applications | |
| CodeMeter Core API | 308 | CmCalculator | 319 |
| Access API | 309 | CmDemo | 317, 318 |
| Authentication API | 310 | WupiCalculator | 319 |
| Enabling API | 310 | CodeMeter service | |
| Encryption API | 310 | Behavior at system startup | 408 |
| Error Management API | 311 | start (Linux) | 412 |

| | |
|---|-----|
| CodeMeter service | |
| start (Mac OS) | 412 |
| start (Sun Solaris) | 413 |
| start (Windows) | 411 |
| stop (Linux) | 412 |
| stop (Mac OS) | 412 |
| stop (Sun Solaris) | 413 |
| stop (Windows) | 411 |
| CodeMeter Start Center | 63 |
| CodeMeter Time Server | |
| Box Time (System Time CmContainer) | 394 |
| Certified Time | 394 |
| System Time | 394 |
| CodeMeter Universal Support Tool | |
| cmu | 474 |
| CodeMeter WebAdmin | 435 |
| Authentication | 451 |
| Backup execute | 470 |
| Certified Time Update | 442 |
| Configuration Access Control | 447 |
| Configuration Backup | 452 |
| Configuration Borrowing | 453 |
| Configuration Certified Time | 450 |
| Configuration Network | 443 |
| Configuration Proxy | 446 |
| Configuration WebAdmin | 451 |
| Configuration Server | 445 |
| Content Backup/Restore | 470 |
| Content CmContainer | 441 |
| Content Licenses | 454 |
| Content User Data | 458 |
| Diagnosis Logfile | 469 |
| Firewall | 436 |
| Free licenses | 462 |
| Help | 472 |
| Home | 439 |
| Info | 472 |
| LAN Server | 445 |
| License display network | 459 |
| License Display Network (summarized) | 460 |
| License Display Network (User) | 463 |
| License display of CmContainer | 454 |
| License Display User (IFI) | 458 |
| Network Port | 437 |
| Profiling | 448 |
| Remote Access | 451 |
| Run CmWAN Server | 445 |
| Run Network Server | 445 |
| Server Cluster | 460 |
| Server License Tracking | 464 |
| Server User | 463 |
| Server Access | 451 |
| Server License Tracking Lizenzverfolgung | 464 |
| Server Search List | 443 |
| Start | 438 |
| WAN Server | 445 |
| White and Blacklist | 448 |
| CodeMeter.ini file | 379 |
| CodeMeterCSSI | 32 |
| Communication mode | |
| IPv4, IPv5 | 437 |
| Platform-specific defaults | 437 |
| Profiling | 437 |
| Shared Memory | 437 |
| Connecting the CmDongle | 408 |
| Customer Owned License Information (COLI) | |
| Product Item Option | 42 |
| - D - | |
| Deployment | 372 |
| "silent" installing of the runtime | 376 |
| Copy installation on Windows | 381 |
| Customizing Installation Packages (Windows) | 375 |
| directed installing of features (Windows) commandline | 377 |
| Merge Module configuration parameter (Windows) | 377 |
| Merge Modules (Windows) | 374 |
| Mobile installation on CmDongle | 379 |
| Non-Windows operating systems | 373 |
| Preconfigured Installation Packages (Windows) | 374 |
| Windows operating systems | 373 |
| Disable Time | 385 |
| suspending | 385 |
| Driver version (minim.) | |
| AxProtector Commandline | 272 |
| - E - | |
| Enabling | 383 |
| Disable Time | 385 |
| Enabling Access Code | 385 |
| Enabling Block | 385 |
| Enabling Level | 387 |
| Enabling Mode | 386 |
| Enabling Status | 386 |
| Example | 388 |
| Lookup | 387 |
| Required Flag | 388 |
| Simple PIN | 385 |
| Time PIN | 385 |
| Encryption | 56 |
| Asymmetric | 61 |

| | | | |
|--------------------------------|--------------|-------------------------------------|----------|
| Encryption | 56 | Modular Software Protection | 296 |
| Direct | 59 | IxProtector .NET | |
| Indirect | 59 | .NET Options | 216 |
| Key derivation | 56 | Activate IxProtector | 217 |
| Symmetric | 59 | Advanced Options | 217 |
| Encryption Code Options | 56 | Create Logfile | 217 |
| Expiration Time | | Customized Error Messages | 215 |
| Product Item Option | 40 | Default Error Messages | 214 |
| Extended Protected Data | | Destination File | 214 |
| Product Item Option | 47 | Error Messages | 214, 215 |
| - F - | | Extended Commandline Options | 217 |
| Feature Code | 43 | File To Protect | 213, 214 |
| AxProtector Commandline | 272 | Inline Messages | 215 |
| Feature Map | | IxProtector Bytes | 223 |
| Product Item Option | 43 | IxProtector Methods | 223 |
| Version Management | 43 | IxProtector Name | 222 |
| FIO (Firm Item Options) | 34 | IxProtector Views | 222 |
| Firm Code | 34 | License List Description | 219 |
| AxProtector Commandline | 271 | License List Feature Code | 219 |
| Firm Item | 34 | License List Firm Code | 219 |
| Firm Item Options (FIO) | 34 | License List Id | 219 |
| Firm Item Text | | License List Ignore Linger Time | 220 |
| define | 333 | License List Licensing Systems | 219 |
| Firm Key | 34 | License List Minimum Driver Version | 220 |
| Firm Security Box (FSB) | 36 | License List Minimum Firmware | 220 |
| Firmware Version | | License List Product Code | 219 |
| AxProtector Commandline | 272 | License List Release Date | 220 |
| Form Factors | | License List Subsystem | 220 |
| CodeMeter | 26 | License List WupiReadData | 220 |
| Human Interface Device (HID) | 26 | License List WupiWriteData | 221 |
| FSB (Firm Security Box) | 36 | Logging | 217 |
| - H - | | No Strong Name | 216 |
| HID | | Optimizing | 217 |
| cmu programming | 477 | Probing | 216 |
| Set to HID | 483 | Source File | 213 |
| Set to Mass Storage Device | 485 | Strong Name from Container | 216 |
| HID (Human Interface Device) | 26, 408, 482 | Strong Name from File | 216 |
| Hidden Data | | Summary Back | 224 |
| Product Item Option | 48 | Summary Finish | 224 |
| Human Interface Device (HID) | 26, 482 | Summary Protect Now | 225 |
| - I - | | User Message DLL | 214 |
| IFI (Implicit Firm Item) | 383 | IxProtector Linux | |
| Implicit Firm Item | 34 | Advanced Options | 243 |
| Individual Software Protection | 296 | Create Logfile | 243 |
| IPv4, IPv6 | 437 | Customized Error Messages | 242 |
| IxProtector | | Default Error Messages | 241 |

- IxProtector Linux
IxProtector Function Length 249
IxProtector Function Description 249
IxProtector Function Id 249
IxProtector Function License List 249
IxProtector Function Name 249
IxProtector Function Trap 249
License List Feature Code 245
License List Firm Code 245
License List Minimum Driver Version 246
License List Minimum Firmware 246
License List Product Code 245
License List Release Date 246
License List Description 245
License List Id 244
License List Ignore Linger Time 246
License List Licensing Systems 245
License List Subsystem 246
License List WupiReadData 246
License List WupiWriteData 247
Logging 243
Source File 240
Summary Back 251
Summary Finish 251
Summary Protect Now 252
Suppress IxProtector Error Messages 241
User Message DLL 241
- IxProtector Mac
Advanced Options 229
Create Logfile 229
Customized Error Messages 228
Default Error Messages 228
Destination File 227
Dynamic loading of Wibu-Systems libraries 229
Error Messsages 228
Extended Commandline Options 229
File To Protect 227
IxProtector Function Length 235
IxProtector Function Description 235
IxProtector Function Id 235
IxProtector Function License List 235
IxProtector Function Name 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 233
Logging 229
Source File 227
Summary Back 237
Summary Finish 237
Summary Protect Now 238
User Message DLL 228
- IxProtector Mac OS
IxProtector Function Trap 235
- IxProtector Windows
Advanced Options 203
Create Logfile 203
Customized Error Messages 202
Default Error Messages 201
Destination File 200
Dynamic loading of Wibu-Systems libraries 203
Error Messsages 201, 202
Extended Commandline Options 203
File To Protect 200
IxProtector Function Length 209
IxProtector Function Description 209
IxProtector Function Id 208
IxProtector Function License List 209
IxProtector Function Name 209
IxProtector Function Trap 209
License List Feature Code 205
License List Firm Code 205
License List Minimum Driver Version 206
License List Minimum Firmware 206
License List Product Code 205
License List Release Date 206
License List Description 205
License List Id 204
License List Ignore Linger Time 206
License List Licensing Systems 205
License List Subsystem 206
License List WupiReadData 206
License List WupiWriteData 206
Logging 203
Source File 200
Summary Finish 211
Summary Back 211
Summary Protect Now 212
Suppress IxProtector Error Messages 201
User Message DLL 201

| | |
|---------------------------------------|---|
| - J - | License update file 424 update 424 |
| jQuery | Linger Time Product Item Option 46 |
| CodeMeter WebAdmin 13 | Locking a CmContainer 396 |
| Copyright 13 | LogFile CodeMeter WebAdmin 469 |
| - K - | |
| Key Derivation | |
| Black Key 56 | |
| Feature Map 56 | |
| Firm Code 56 | |
| Product Code 56 | |
| Release Date 56 | |
| - L - | |
| License Borrowing | Maintenance Period Product Item Option 45 |
| Example 353 | Modular Software Protection 296 |
| License Borrowing (CmBoxPgm) 352, 353 | |
| License information file (*.lif) 29 | |
| License Model | - N - |
| Concurrent License 52 | Network access mode AxProtector Commandline 273 |
| Demo Version 52 | |
| Downgrade Management 54 | |
| Floating License 52 | |
| Hot / Cold Standby 54 | |
| License Borrowing 55 | |
| Local Single User Licenses 51 | |
| Machine-bound Licenses 55 | |
| Modular Licenses 53 | |
| Named User Licenses 55 | |
| Network License 52 | |
| Overflow Licenses 54 | |
| Pay-per ... 50, 53 | |
| Renting, Leasing 53 | |
| Standard 50 | |
| Version Management 54 | |
| License Quantity | - O - |
| Product Item Option 39 | Obfuscation AxProtector .NET 282, 283 |
| License request file | On-demand Decryption 56 |
| Add a license of a new ISV 430 | Operating Systems CodeMeter 30 |
| create 427 | |
| Extend existing licenses 428 | Own Key Hidden Data 391 Secret Data 391 |
| License Tracking 464 | - P - |
| License update file | PIO (Product Item Options) 37 |
| import 431 | Preconfigured Installation Packages (Windows) |
| Licenses | Full Installation Package 374 |
| *.WibuCmRac 424 | Installation Package for applications using FSB functions 374 |
| *.WibuCmRaU 424 | Merge modules 374 |
| CmFAS 424 | Reduced Installation Package 374 |
| import 424 | |
| license request file 424 | Product Code 34 AxProtector Commandline 271 Product Item Option 38 |
| | Product Item 34 |
| | Product Item Option Activation Time 40 Customer Owned License Information (COLI) 42 |
| | Expiration Time 40 Extended Protected Data 47 |
| | Feature Map 43 Hidden Data 48 License Quantity 39 |
| | Linger Time 46 Maintenance Period 45 |
| | Product Code 38 Protected Data 47 |

- Product Item Option
 - Secret Data 48
 - Text 38
 - Unit Counter 42
 - Usage Period 41
 - User Data 46
- Product Item Options (PIO) 34, 37
- Profiling 437
 - Location different operating systems 478
- Programming of CmContainer
 - *.wbb 368
 - *.WibuCmRaC 368
 - *.WibuCmRaM 368
 - *.WibuCmRaU 368
 - CmBoxPgm 331
 - CmLicense Editor 322
 - CodeMeter License Central 360
 - LIF, License Information File 368
 - Programming via file transfer 368
- Programming Samples
 - Samples 18
- Protected Data
 - Product Item Option 47
- Protection Only license
 - CmActLicense binding 29
 - Programming example 351
- R -**
 - Receipt 433
 - Release Date 45
 - AxProtector Commandline 272
- S -**
 - Samples
 - Programming 18
 - Search order for licenses
 - AxProtector Commandline 272
 - Search order for licenses (WAN)
 - AxProtector Commandline 272
 - Secret Data
 - Product Item Option 48
 - Server Search List 443
 - *.ini configuration file 443
 - Server Search List (CodeMeter WebAdmin) 443
 - Shared Memory 437
 - Smart Bind
 - CmActLicense 345
 - CmBoxPgm 345
 - SmartBind 28
- Support
 - Wibu-Systems 19
- System startup
 - CodeMeter service 408
- T -**
 - Temporary Enabling 386
 - Text
 - Product Item Option 38
 - Token 32
 - CodeMeter 32
 - Trial license
 - CmActLicense binding 29
 - Programming example 349
- U -**
 - UIK (User Individual Key) 383
 - Unit Counter
 - Product Item Option 42
 - Usage Period 41
 - Product Item Option 41
 - User Data
 - Product Item Option 46
 - User Individual Key (UIK) 383
- W -**
 - WAN
 - infrastructure 399
 - WAN, Wide Area Network 399
 - wbb file (CmActLicense) 408
 - wbc file 17
 - Wide Area Network, WAN 399
 - WUPI
 - Example: WupiCalculator 301
 - WUPI Function
 - WupiAllocateLicense 298
 - WupiCheckDebugger 298
 - WupiCheckLicense 298
 - WupiDecreaseUnitCounter 298
 - WupiDecryptCode 298
 - WupiEncryptCode 298
 - WupiFreeLicense 298
 - WupiGetHandle 298
 - WupiGetLastError 301
 - WupiQueryInfo 299
 - WupiReadData 300
 - WupiReadDataInteger 300
 - WupiWriteData 300, 301
 - WupiWriteDataInteger 301
 - WupiAllocateLicense 298

WupiCheckDebugger 298

WupiCheckLicense 298

WupiDecryptCode 298

WupiEncryptCode 298

WupiFreeLicense 298

WupiGetHandle 298

WupiGetLastError 301

WupiQueryInfo 299

WupiReadData 300

WupiReadDataInteger 300

WupiWriteData 300, 301

WupiWriteDataInteger 301

- X -

X.509 Certificates

CodeMeter 32