



CodeMeter Entwickler-Handbuch

Version 5.10 - Oktober 2013

Gedruckt in Deutschland

Alle Rechte vorbehalten. Kein Teil dieser Dokumentation, der dazugehörigen Software und der anderen Bestandteile des beschriebenen Produkts darf ohne schriftliche Genehmigung der Firma Wibu-Systems nachgedruckt, vervielfältigt, in eine andere Sprache übersetzt oder auf elektromagnetischem, optischem, mechanischem oder anderem Wege abgespeichert werden.

Diese Dokumentation, die Hardware (CmDongle usw.) und die dazugehörige Software wurden mit großer Sorgfalt erstellt. Dennoch sind Irrtümer nicht auszuschließen. Wibu-Systems weist darauf hin, dass für Fehler innerhalb der Dokumentation, der Hardware oder der Programme seitens des Benutzers keinerlei Haftungsansprüche geltend gemacht werden können.

Wibu-Systems behält sich das Recht vor, Programme oder die Dokumentation von Zeit zu Zeit zu ändern, ohne den Benutzer darüber informieren zu müssen.

WIBU, CodeMeter, SmartShelter sind geschützte Warenzeichen von Wibu-Systems. Alle anderen in dieser Dokumentation genannten Marken- und Produktnamen sind Handelsnamen, Dienste, Warenzeichen und Firmennamen sind in der Regel durch ihren Inhaber geschützt.

Wibu-Systems ist Mitglied im:



PCMCIA seit 1993



USB Implementers Forum seit 1997



SD Card Association seit 2007



Bitkom Bundesverband Informationswirtschaft, Telekommunikation und Neue Medien seit 2003



VDMA Verband Deutscher Maschinen- und Anlagenbau e.V. seit 2008



OPC Foundation seit 2012

wie auch Mitglied in den Entwicklerprogrammen von Autodesk, Apple, HP, IBM, Intel und Microsoft.



OEM Hardware Solutions
Partner

Microsoft Gold Certified Partner



Microsoft Embedded Partner



Strategic Software Partner Industrial and Medical

Inhaltsverzeichnis

I	Version	13
II	Einleitung	15
1	Sicherheitshinweise	17
2	Installation	17
3	Mitgelieferter CmDongle	17
4	Weitere Hilfe-Dokumentation	18
5	Typographische Konventionen	19
6	Support durch Wibu-Systems	19
7	Über Wibu-Systems	20
III	Softwareschutz und Lizenzmanagement	22
1	CmDongle: Vielfalt der CodeMeter-Bauformen	26
2	CmActLicense: Bindung und Aktivierung	28
2.1	CmActLicense: Bindungsschemata	28
2.2	CmActLicense: Aktivierung	30
3	Vielfalt der Betriebssysteme für CodeMeter	31
4	Zusätzliche Eigenschaften	32
5	CodeMeter als Token	32
6	CodeMeter auf Embedded Systemen	34
IV	Das CodeMeter-Konzept	35
1	Product Item Options - Lizenzinträge nach Maß	38
1.1	Product Code	39
1.2	Text	40
1.3	License Quantity (Lizenzanzahl)	40
1.4	Activation Time (Aktivierungsdatum)	41
1.5	Expiration Time (Verfallsdatum)	42
1.6	Usage Period (Nutzungsdauer)	43
1.7	Customer Owned License Information (COLI)	43
1.8	Unit Counter (Begrenzungszähler)	44
1.9	Feature Map	45
1.10	Maintenance Period (Wartungszeitraum)	47
1.11	Linger Time	48
1.12	User Data	48

1.13 Protected Data	49
1.14 Extended Protected Data	49
1.15 Hidden Data	50
1.16 Secret Data	51
2 Sicherheit großgeschrieben	51
3 Lizenzmodelle - Abbildungsvielfalt mit CodeMeter	52
3.1 Die Umsetzung von Lizenzmodellen mit CodeMeter	54
3.1.1 Lokale Einzelplatzlizenz.....	54
3.1.2 Concurrent-/ Floating-Lizenz im Netzwerk.....	54
3.1.3 Demo-Versionen	55
3.1.4 Modularer Lizenz.....	55
3.1.5 Miete, Leasing	56
3.1.6 Tatsächlich ausgeführte Aktionen.....	56
3.1.7 Downgrade/Versionsmanagement.....	56
3.1.8 Overflow	57
3.1.9 Hot / Cold Standby.....	57
3.1.10 Named User Lizenzen.....	58
3.1.11 Rechnergebundene Lizenzen.....	58
3.1.12 Lizenzausleihe	58
4 Sicherheit durch Verschlüsselung	58
4.1 Schlüsselableitung: ein Lizenzeintrag - viele Schlüssel	59
5 Kryptographie	62
5.1 Direkte und indirekte Verschlüsselung	62
5.2 Symmetrische Verschlüsselung	62
5.2.1 Streamcipher (AES_STREAM).....	63
5.2.2 Electronic Codebook Mode (AES_ECB).....	63
5.2.3 AES - Cipher Block Chaining Mode (CBC) (empfohlen).....	63
5.2.4 AES - Cipher Feedback Mode (CFB).....	64
5.3 Asymmetrische Verschlüsselung	64
5.3.1 ECC - Elliptic Curve Cryptography.....	64
5.3.2 ECIES - Elliptic Curve Integrated Encryption Scheme.....	64
5.3.3 ECDSA - Elliptic Curve Digital Signature Algorithm.....	65
5.3.4 RSA	65
5.4 Zusätzliche Verschlüsselungen	65
V CodeMeter Start Center	66
1 Aufbau und Navigation	66
1.1 Menüleiste	66
VI CodeMeter Lizenzserver	68

VII Automatischer Softwareschutz mit AxProtector	71
1 Aufbau und Navigation	72
1.1 Menüleiste	73
1.2 Navigationsfenster	74
1.3 Eingabefenster	74
1.4 Hinweis- und Fehler-Fenster	74
1.5 Projekttyp-Bereich	75
2 Projekt Dialog	75
3 Projekttypen	75
4 AxProtector Karteireiter	76
4.1 Windows Anwendung oder DLL	76
4.1.1 Dateiauswahl	77
4.1.2 Lizenzierungssysteme	78
4.1.3 Lizenzbelegung	81
4.1.4 Laufzeiteinstellungen	83
4.1.4.1 Erweiterte Laufzeiteinstellungen	85
4.1.5 Sicherheitsoptionen	88
4.1.5.1 Erweiterte Sicherheitsoptionen	92
4.1.6 Fehlermeldungen	92
4.1.7 Erweiterte Optionen	95
4.1.7.1 Lizenzlisten	96
4.1.7.2 IxProtector	101
4.1.7.3 Dateiverschlüsselung	103
4.1.8 Zusammenfassung	107
4.2 NET Assembly	110
4.2.1 Dateiauswahl	112
4.2.2 Lizenzierungssysteme	113
4.2.3 Lizenzbelegung	116
4.2.4 Laufzeiteinstellungen	118
4.2.4.1 Erweiterte Laufzeiteinstellungen	120
4.2.5 Sicherheitsoptionen	124
4.2.6 Fehlermeldungen	127
4.2.7 .NET Einstellungen	129
4.2.8 Erweiterte Optionen	130
4.2.8.1 Lizenzlisten	131
4.2.8.2 IxProtector	135
4.2.9 Zusammenfassung	138
4.3 Mac OS X Anwendung oder Dylib	140
4.3.1 Dateiauswahl	141

4.3.2 Lizenzierungssysteme.....	142
4.3.3 Lizenzbelegung	145
4.3.4 Laufzeiteinstellungen.....	147
4.3.4.1 Erweiterte Laufzeiteinstellungen.....	149
4.3.5 Fehlermeldungen	152
4.3.6 Sicherheitsoptionen.....	153
4.3.6.1 Erweiterte Sicherheitsoptionen.....	156
4.3.7 Erweiterte Optionen.....	157
4.3.7.1 Lizenzlisten	158
4.3.7.2 IxProtector	163
4.3.8 Zusammenfassung.....	165
4.4 Java Anwendung (jar-Datei)	168
4.4.1 Dateiauswahl	170
4.4.2 Lizenzierungssysteme.....	171
4.4.3 Lizenzbelegung	174
4.4.4 Laufzeiteinstellungen.....	176
4.4.4.1 Erweiterte Laufzeiteinstellungen.....	178
4.4.5 Sicherheitsoptionen.....	182
4.4.6 Fehlermeldungen	183
4.4.7 Java Einstellungen.....	184
4.4.8 Erweiterte Optionen.....	186
4.4.9 Zusammenfassung.....	187
4.5 Linux Anwendung oder Shared Object	190
4.5.1 Dateiauswahl	191
4.5.2 Lizenzierungssysteme.....	192
4.5.3 Lizenzbelegung	195
4.5.4 Laufzeiteinstellungen.....	197
4.5.4.1 Erweiterte Laufzeiteinstellungen.....	199
4.5.5 Sicherheitsoptionen.....	202
4.5.5.1 Erweiterte Sicherheitsoptionen.....	205
4.5.6 Fehlermeldungen	206
4.5.7 Erweiterte Optionen.....	208
4.5.7.1 Lizenzlisten	209
4.5.7.2 IxProtector	213
4.5.8 Zusammenfassung.....	215
5 IxProtector Karteireiter	218
5.1 Windows Anwendung oder DLL	218
5.1.1 Dateiauswahl	219
5.1.2 Fehlermeldungen	220
5.1.3 Erweiterte Optionen.....	222

5.1.3.1 <i>Lizenzlisten</i>	223
5.1.3.2 <i>IxProtector</i>	227
5.1.4 Zusammenfassung.....	229
5.2 .NET Assembly	232
5.2.1 Dateiauswahl	233
5.2.2 Fehlermeldungen	234
5.2.3 .NET Einstellungen.....	236
5.2.4 Erweiterte Optionen.....	237
5.2.4.1 <i>Lizenzlisten</i>	238
5.2.4.2 <i>IxProtector</i>	242
5.2.5 Zusammenfassung.....	245
5.3 Max OS X Anwendung oder Dylib	247
5.3.1 Dateiauswahl	248
5.3.2 Fehlermeldungen	249
5.3.3 Erweiterte Optionen.....	251
5.3.3.1 <i>Lizenzlisten</i>	252
5.3.3.2 <i>IxProtector</i>	256
5.3.4 Zusammenfassung.....	258
5.4 Linux Anwendung oder Shared Object	261
5.4.1 Dateiauswahl	262
5.4.2 Fehlermeldungen	263
5.4.3 Erweiterte Optionen.....	265
5.4.3.1 <i>Lizenzlisten</i>	266
5.4.3.2 <i>IxProtector</i>	270
5.4.4 Zusammenfassung.....	272
6 Sonstige Karteireiter	275
6.1 Dateiverschlüsselung	275
6.1.1 Dateiauswahl	276
6.1.2 Lizenzierungssysteme.....	277
6.1.3 Lizenzbelegung	280
6.1.4 Erweiterte Optionen.....	282
6.1.4.1 <i>Lizenzlisten</i>	283
6.1.4.2 <i>Dateiverschlüsselung</i>	286
6.1.5 Zusammenfassung.....	289
7 Kommandozeilen-Optionen für AxProtector	292
7.1 Grundsätzliche Einstellungen	292
7.2 Einstellungen zum Lizenzierungssystem	293
7.3 Einstellungen zu Verschlüsselungsvorgängen	296
7.4 Einstellungen mit Bezug zur Laufzeit	310
7.5 Java-spezifische Einstellungen	313

7.6 Bedienungseinstellungen	319
VIII Individueller Softwareschutz mit IxProtector, WUPI und CodeMeter Kern-API	320
1 IxProtector und das Softwareschutz-API - WUPI	320
2 WUPI Funktionen	322
2.1 Individueller Softwareschutz mit WUPI: ein Beispiel Indexbasierte Platzhalter	326
2.1.1 Definition der Module.....	326
2.1.2 Platzhalter in IxProtector Lizenz- und Funktionslisten.....	327
2.1.3 Programmierung des CmContainer.....	330
2.1.4 Einfügen in den Quelltext.....	331
2.1.5 Verschlüsselung mit AxProtector.....	333
3 Das CodeMeter Kern-API	333
3.1 Funktionsbereiche	334
3.1.1 Zugriffs API	334
3.1.2 Authentifizierungs API.....	334
3.1.3 Enabling API	334
3.1.4 Verschlüsselungs API.....	335
3.1.5 Fehlermanagement API.....	335
3.1.6 Management API	336
3.1.7 Programming API	336
3.1.8 Remote Update API.....	336
3.1.9 Zeit Management API.....	337
3.2 CodeMeter API Guide	337
3.2.1 Aufbau und Navigation.....	338
3.2.2 Menüleiste	338
3.2.3 Karteireiter	340
3.2.4 Baumansichtsfenster.....	340
3.2.5 Handle-Anzeigefenster.....	340
3.2.6 Interaktiv-Bereich	341
3.2.7 Quellcode-Bereich.....	341
3.3 Beispianwendungen: CmDemo, CmCalculator, WupiCalculator	341
3.3.1 CmDemo	341
3.3.2 CmCalculator	343
3.3.3 WupiCalculator	343
IX Programmierung von CmContainern und Lizenzierungsverwaltung	344
1 CodeMeter License Editor	346
1.1 Oberfläche und Navigation	347
1.1.1 Menüleiste	347
1.1.2 Symbolleiste	348

1.1.3	Baumansichtsfenster.....	349
1.1.4	Darstellungsfenster.....	350
1.1.5	Ausgabefenster	350
1.2	Arbeiten mit CodeMeter License Editor	351
1.2.1	Starten CodeMeter License Editor.....	351
1.2.2	Anzeige von angeschlossenen CmDongles.....	351
1.2.2.1	<i>Aktualisieren der Anzeige.....</i>	<i>351</i>
1.2.2.2	<i>Remote Programmiermodus.....</i>	<i>351</i>
1.2.3	Anlegen und Ändern eines Firm Items.....	352
1.2.4	Löschen eines Firm Items.....	352
1.2.5	Anlegen und Ändern eines Product Items.....	353
1.2.6	Löschen eines Product Items.....	354
1.2.7	Ausführen der Programmierung.....	354
2	CmBoxPgm	355
2.1	Syntax der Kommandozeile	355
2.2	Verwendung	356
2.3	Grundlegende Befehle	356
2.4	CmContainer Optionen	357
2.5	Firm Item Optionen	359
2.6	Product Item Optionen	360
2.7	CmActLicense Optionen	368
2.8	Lizenzausleihe Optionen	376
2.9	FSB Einträge Optionen	378
2.10	Enabling Optionen	379
2.11	Spezielle Befehle	381
3	Die CodeMeter License Central	384
3.1	Das Prinzip	384
3.2	Architektur	386
3.3	Funktionen	387
3.3.1	Sales-Interface	387
3.3.1.1	<i>Connectoren</i>	<i>387</i>
3.3.1.2	<i>Gateway</i>	<i>388</i>
3.3.2	Depot-Interface	390
3.3.3	Admin-Interface	391
3.4	Einsatz-Szenarien für die CodeMeter License Central	391
4	Programmierung per Dateiaustausch	392
X	Auslieferungsoptionen (Deployment)	395
1	Installationspakete für Nicht-Windows Betriebssysteme	396

2 Auslieferung für Windows Betriebssysteme	396
2.1 Vorkonfigurierte Installationspakete	397
2.2 Anpassungsoptionen für Installationspakete	398
3 Installation mobil auf dem CmDongle (Windows)	402
4 Kopieren der CodeMeter Runtime ohne Installation unter Windows	404
XI Erweiterte CodeMeter Eigenschaften	406
1 Die Implicit Firm Item (IFI) Ebene	406
2 Enabling	406
2.1 Enabling Blocks als Ein- und Ausschalter	408
2.1.1 Zugriffscode - Enabling Access Code	408
2.1.2 Zugriffsart - Simple oder Time PIN.....	408
2.1.3 Aktivierungsmodus - Enabling Mode.....	409
2.1.4 Löschen und Bearbeiten von Enabling Blocks.....	409
2.2 Zuordnung (Lookup) von Enabling Blocks	410
2.2.1 Berechtigungsebenen - Enabling Level.....	410
2.2.2 Der Pflicht-Kennzeichner - Required Flag	411
2.3 Enabling-Beispiel	412
3 Verwendung eigener Schlüssel	415
4 Der Zeitserver: System-Zeiten und die Zertifizierte Uhrzeit	417
5 Sperren des CmContainers	421
6 Sicherung des CmDongle-Inhaltes	422
7 CodeMeter im Wide Area Network (WAN)	423
7.1 WAN-Infrastruktur	424
7.2 CodeMeter-seitige Implementierung	426
7.2.1 Programmierung der Lizzenzen.....	426
7.2.2 Verschlüsselung der geschützten Anwendung.....	429
7.2.3 Konfigurieren der CmWAN-Netzwerkkommunikation.....	429
7.2.3.1 CodeMeter WebAdmin-Konfiguration.....	430
7.2.3.2 Profiling in der Registry oder in der server.ini-Datei.....	431
XII Handbuch	433
1 Wichtige erste Informationen	433
2 Installation	435
2.1 Installation unter 32/64-Bit Windows	436
2.1.1 Installierte Dateien unter 32/64-Bit Windows.....	436
2.1.2 Deinstallation unter 32/64-Bit Windows.....	438

2.2 Installation unter Mac OS Betriebssystemen	438
2.2.1 Installierte Dateien unter Mac OS.....	438
2.2.2 Deinstallation unter Mac OS.....	440
2.3 Installation unter Linux Betriebssystemen	440
2.3.1 Deinstallation unter Linux.....	441
2.4 Installation unter Sun Solaris Betriebssystemen	441
2.4.1 Deinstallation unter Sun Solaris.....	443
3 CodeMeter Kontrollzentrum	443
3.1 Struktur und Navigation	445
3.2 Menüleiste	446
3.3 Lizenz-Karteireiter	449
3.4 Ereignisse-Karteireiter	453
3.5 Ausleihe-Karteireiter	453
3.6 Status und Öffnen von CodeMeter WebAdmin	454
4 Einspielen und Aktualisierung von Lizenen	455
4.1 Der CmFAS Assistent im CodeMeter Kontrollzentrum	455
4.1.1 Erzeugen der Lizenzanforderungsdatei.....	457
4.1.1.1 Bestehende Lizenz erweitern.....	457
4.1.1.2 Lizenz eines neuen Herstellers hinzufügen.....	459
4.1.2 Lizenzaktualisierung einspielen.....	461
4.1.3 Quittung erzeugen.....	462
5 CodeMeter WebAdmin	464
5.1 Voraussetzungen	466
5.2 Starten von CodeMeter WebAdmin	467
5.3 Statusinfomation	467
5.3.1 Generelle Informationen.....	468
5.3.2 Informationen über den CmContainer.....	469
5.4 Konfiguration	471
5.4.1 Netzwerk	471
5.4.2 Server	473
5.4.3 Proxy Einstellungen.....	475
5.4.4 Zugriffsschutz	476
5.4.5 Zeitserver	479
5.4.6 WebAdmin	480
5.4.7 Datensicherung	481
5.4.8 Lizenzausleihe	482
5.5 Lizenzanzeige	482
5.5.1 Lokale Lizenen	483
5.5.1.1 Lizenzgeber-Informationen.....	484
5.5.1.2 Produkt-Informationen.....	484

5.5.2	Benutzerdaten	486
5.6	Lizenzanzeige im Netzwerk	487
5.6.1	Cluster - Lizenzen zusammengefasst.....	488
5.6.1.1	<i>Session Details</i>	489
5.6.2	Aktuell angemeldete Benutzer.....	490
5.6.3	Lizenzverfolgung	491
5.7	Diagnose	496
5.8	Datensicherung	497
5.9	Info	499
5.10	Hilfe	499
6	CmDust	500
7	CMU - CodeMeter Universal Support Tool	502
8	CodeMeter License Tracking	505
8.1	Voraussetzungen	505
8.2	Konfiguration	505
8.2.1	Profiling	505
8.3	Format der Protokollierungsdatei	506
8.3.1	Definitionen und Wertebereiche.....	507
8.4	Eintragstypen (Entry Types)	508
8.4.1	List of Licenses-Eintrag.....	508
8.4.2	License-Eintrag	508
8.4.3	Access-Eintrag	509
8.4.4	Release-Eintrag	509
8.4.5	Borrow Access-Eintrag.....	509
8.4.6	Borrow Return-Eintrag.....	509
8.4.7	Denial-Eintrag	510
8.4.8	Administrative-Eintrag.....	510
9	HID-Unterstützung	510
9.1	Umstellen: Massenspeicher zu HID	511
9.2	Umstellen: HID zu Massenspeicher	513
XIII	Glossar	516
Index		519

1 Version

CodeMeter® Entwickler-Handbuch Version 5.10, 16.10.2013

Copyright © 2007-2013

durch WIBU-SYSTEMS AG, Karlsruhe / Germany

Alle Rechte vorbehalten.

Wibu-Systems Kontaktinformationen:

Adresse: WIBU-SYSTEMS AG
Rüppurrer Straße 52-54
D-76137 Karlsruhe

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

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

E-mail: support@wibu.com

AxProtector Java verwendet die ASM Java-Programmbibliothek.

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 nutzt jQuery-Funktionalitäten.

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 Einleitung

Die Schutztechnologie *CodeMeter®* von Wibu-Systems bietet sichereren Schutz und effektive Lizenzkontrolle für Software und digitale Inhalte. Das vorliegende Entwickler-Handbuch ist in verschiedene Kapitel aufgeteilt.

Dieses Kapitel gibt Ihnen einen Überblick über den Aufbau des Handbuchs, enthält Hinweise für Nutzer des *CodeMeter®* Software Development Kits (SDK), informiert Sie über die verwendete Typographie und hilft Ihnen bei der Kontaktaufnahme mit dem Support Team von Wibu-Systems.

Das folgende [Kapitel II](#)²² zeigt überblicksartig die Merkmale, die *CodeMeter®* in den Bereichen Sicherheit, hard- und softwarebasierter Softwareschutz und flexibles Lizenzmanagement besonders auszeichnet. Das [Kapitel III](#)³⁵ schließt an und beschreibt wie das *CodeMeter®*-Konzept Schutz-, Lizenzierungs- und Sicherheitsanforderungen umsetzt und führt grundlegende Begrifflichkeiten ein.

Das [Kapitel IV](#)⁶⁶ folgt mit der Beschreibung von *CodeMeter Start Center*, der Kommunikationszentrale zum Aufruf der einzelnen *CodeMeter®*-Werkzeuge, während [Kapitel V](#)⁶⁸ *CodeMeter Lizenzserver* als zentrale Komponente von *CodeMeter®* beschreibt, der als Dienst auf jedem Rechner läuft, auf dem geschützte digitale Inhalte verwendet werden sollen.

[Kapitel VI](#)⁷¹ und [Kapitel VII](#)³²⁰ zeigen Ihnen, wie Sie automatischen und individuellen Schutz in Ihre Software integrieren. Zum einen über *AxProtector*, dem automatischen Softwareschutz für verschiedene Projekttypen mit den Varianten der Benutzeroberfläche und Kommandozeile. Zum anderen über *IxProtector* für den individuellen Softwareschutz mit der Schnittstelle *Softwareschutz-API* (WUPI) und das grundlegende *CodeMeter Kern-API*.

[Kapitel VIII](#)³⁴⁴ beschreibt die Anwendungen, die Ihnen zum Erstellen, Verwalten und Ausliefern von *CodeMeter®*-Lizenzen zur Verfügung stehen: *CodeMeter License Editor*, *CmBoxPgm* und *CodeMeter License Central*. [Kapitel IX](#)³⁹⁵ schließt sich mit einer Beschreibung der Auslieferungsoptionen an: Was geht an Software wie an Ihre Kunden?

[Kapitel X](#)⁴⁰⁶ informiert Sie über erweiterte *CodeMeter®*-Eigenschaften wie *Implicit Firm Item*, *Enabling*, Verwendung eigener Schlüssel, und die Sicherung des *CmContainer*-Inhaltes.

Das [Kapitel XI](#)⁴³³ schließlich ist als Administrator-Handbuch gedacht, das die *CodeMeter®*-Installationsinformationen für unterschiedliche Betriebssysteme enthält und die Werkzeuge *CodeMeter WebAdmin*, *CodeMeter Kontrollzentrum* sowie *CmDust* und *cmu* beschreibt, die den Administrator in der täglichen Arbeit mit *CodeMeter®* unterstützen.

Den Abschluss bilden ein Glossar und ein ausführlicher Index.

Insgesamt orientiert sich der Aufbau an folgendem Schaubild.

Softwareschutz (Verschlüsselung) und Integration in die Software	Navigation	Programmierung und Produktion von CmContainern, Verwaltung von Lizzenzen
<p>Automatische Integration</p> <ul style="list-style-type: none"> AxProtector Schutztechnologie und Werkzeug zum automatischen Schutz von Anwendungen ohne Eingriffe in den Quelltext während der Entwicklung. <p>Individuelle Integration</p> <ul style="list-style-type: none"> IxProtector In AxProtector integrierte Schutztechnologie zum individuellen Schutz von Anwendungen mit Eingriff in den Quelltext während der Entwicklung. Softwareschutz-API WUPI Schlanke Schnittstelle (WIBU Universal Protection Interface) und Werkzeug zum individuellen Schutz von Anwendungen mit Eingriff in den Quelltext während der Entwicklung und zur Laufzeit. Kern-API Schnittstelle und Werkzeug zum individuellen Schutz von Anwendungen mit Eingriff in den Quelltext. Mit CodeMeter API Guide zur interaktiven Bedienung während der Entwicklung und Laufzeit. 	<p>CodeMeter Start Center Einstiegsseite für den Zugriff auf die zentralen Werkzeuge.</p>  <p>Laufzeit-Komponenten für den Entwickler, Administrator und Endkunden</p> <ul style="list-style-type: none"> CodeMeter Kontrollzentrum Oberfläche zur lokalen Konfiguration von CodeMeter Lizenzserver inklusive Aktualisierung von Lizzenzen, CmFAS Assistent. CodeMeter WebAdmin Anwendung zur Konfiguration von CodeMeter Lizenzserver und Anzeige der vorhandenen Lizzenzen im CmContainer. 	<p>CodeMeter License Central (Desktop / Internet) Werkzeug zur Integration von Softwareschutz in Vertriebs-, Produktions- und Supportprozesse (Backoffice-Integration).</p> <ul style="list-style-type: none"> CodeMeter Producer Datenbankorientiertes Werkzeug zur Programmierung von CmContainern abgelöst durch CodeMeter License Central. Ab Version 4.1 keine Weiterentwicklung. CmBoxPgm Kommandozeilen-Werkzeug zur Batch-Programmierung von CmContainern mit Skripten in der Produktion. CodeMeter License Editor Graphisches Werkzeug zur Programmierung von CmContainern zum Testen von Lizenzierungsstrategien. Programmier-API Schnittstelle zur Programmierung von CmContainern und zur Lizenzverwaltung in eigenen Anwendungen, (HIP, High Level Programming Interface).

Abbildung 3: Aufbau der Dokumentation

2.1 Sicherheitshinweise

Die Hardware der WIBU-SYSTEMS AG dient dem Schutz und Lizenzierung digitaler Produkte und wurde entsprechend dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln entwickelt, gefertigt und geprüft.

Für weitergehende Informationen zu Zertifizierungen der Hardware siehe die entsprechenden Dokumente auf der Wibu-Systems [Webseite](http://www.wibu.com/de/zertifikate.html) (<http://www.wibu.com/de/zertifikate.html>).

Beachten Sie bitte die folgenden Sicherheitshinweise:

- Bitte verwenden Sie die Hardware nur für die in dieser Anleitung beschriebenen bestimmungsgemäßen Art und Weise. Schließen Sie die Hardware nur an die jeweils passende vorgesehene Schnittstelle an. Eine anderweitige Verwendung oder das Öffnen oder eigenständige Reparaturen der Hardware führen eventuell zu Beschädigungen am Produkt oder in dessen Umgebung. Das Verändern der Hardware beeinträchtigt die Produktsicherheit. Achtung Verletzungsgefahr!
- Im laufenden Betrieb kann sich die Hardware erwärmen - das Erwärmen stellt jedoch einen normalen Betriebsparameter dar.
- Halten Sie die Hardware von Nässe und hoher Luftfeuchtigkeit fern und vermeiden Sie starke Erschütterungen, Staub, Hitze und direkte Sonneneinstrahlung, um Betriebsstörungen zu vermeiden.
- Abhängig vom verwendeten Betriebssystem kann die Erkennung der Hardware einige Sekunden beanspruchen. Nach dem beendeten Zugriff auf die Hardware sollte abhängig vom verwendeten Betriebssystem mehrere Sekunden gewartet werden, bevor die Hardware entfernt wird. Andernfalls kann keine sichere Datenspeicherung /Datenübertragung gewährleistet werden.
- Dieses Produkt ist kein Spielzeug und gehört nicht in Kinderhände.

Das Nichtbeachten der Sicherheitshinweise schließt eine Gewährleistung aus.

2.2 Installation

Zur Installation von *CodeMeter®* auf Windows Betriebssystemen legen Sie die beiliegende DVD in Ihr DVD-ROM Laufwerk. Danach erscheint automatisch das *CodeMeter®* DVD-Menü.



Sollte das DVD-Menü nicht erscheinen, starten Sie die Datei `start.exe`, die sich im Wurzelverzeichnis der DVD befindet.

Nach Auswahl der gewünschten Sprache klicken Sie auf die "**CodeMeter SDK**" Schaltfläche. Folgen Sie danach den Anweisungen des Installationsassistenten, um das *CodeMeter®* SDK auf Ihren PC zu installieren.

Zur Installation auf andere Betriebssysteme suchen Sie bitte die entsprechenden Dateien aus der Ordnerstruktur aus.

2.3 Mitgelieferter CmDongle

Zusammen mit dem *CodeMeter®* Software Development Kit (SDK) erhalten Sie einen Dongle, die *CodeMeter®*-Hardware, den *CmDongle*.

Dieser ist gleichzeitig der führende *CmDongle*, die sogenannte Firm Security Box, mit der Sie andere *CmContainer* programmieren können.

 Für *CmDongle* ist bereits ein Evaluierungslizenzeintrag mit dem öffentlichen Firm Code 10 und Product Code 13 ist programmiert.

Für die softwarebasierte *CodeMeter®*-Variante *CmActLicense* besitzen Sie eine Evaluierungslizenz mit Firm Code 5010 und Product Code 13.

Entscheiden Sie sich für *CodeMeter®*, erhalten Sie einen individuellen Firm Code. Sie erhalten eine *CmFirm.wbc*-Datei (bei *CmActLicense* noch zusätzlich eine *CmActFI.wbc*). Beide Dateien können Sie per Drag & Drop über das *CodeMeter Kontrollzentrum* einspielen. Sie finden die Dateien im Verzeichnis "C:\ProgramData\CodeMeter\DevKit".

Danach können Sie mit dieser Firm Security Box als Lizenzgeber Lizenzinformationen in andere *CmContainer* übertragen.

2.4 Weitere Hilfe-Dokumentation

Zusätzlich zu diesem Handbuch (erreichbar über "**Start | Alle Programme | CodeMeter | Documentation**") stehen Ihnen die folgenden Onlinehilfe-Dateien zur Verfügung. Diese finden Sie über die entsprechenden Aufrufe in den Anwendungen sowie in den angelegten Menü-Einträgen nach der Installation des SDK (Software Development Kit).

Hilfe-Datei	Erreichbar über
<i>CodeMeter®</i> Benutzerhilfe als HTML-Dateien inklusive den Teilen <i>CodeMeter® Runtime Kit</i> , <i>CodeMeter Lizenzserver</i> , <i>CodeMeter Kontrollzentrum</i> , <i>cmu Kommandozeilenprogramm</i> , <i>CodeMeter WebAdmin</i> , Lizenzierung - <i>Field-Activation-Service</i> , <i>CodeMeter® FAQ</i> (deutsch und englisch)	entsprechende Menüeinträge oder Schaltflächen oder " Start Alle Programme CodeMeter Documentation " [%Program Files%\CodeMeter\Runtime\help\CmUserHelp]
AxProtector-Hilfe als kompilierte Hilfedatei in Deutsch und Englisch	" Start Alle Programme AxProtector Help "
<i>Software Protection API</i> als kompilierte Hilfedatei in Englisch	" Start Alle Programme CodeMeter Documentation "
<i>Kern-API</i> als kompilierte Hilfedatei in Englisch	" Start Alle Programme CodeMeter Documentation "
<i>CodeMeter Java-API</i> als HTML-Dateien in Englisch	" Start Alle Programme CodeMeter Documentation "
<i>Programming API</i> als HTML-Dateien in Englisch	" Start Alle Programme CodeMeter Documentation Programming API " für die Programmiersprachen C++, Delphi, Java.
Beispiele zur Programmierung von <i>CmContainern</i> und zugehörige Sample Help Documentation	" Start Alle Programme CodeMeter Samples " [%\Users%\Public\Documents\WIBU-SYSTEMS] Anleitung " Start Alle Programme CodeMeter Documentation "

2.5 Typographische Konventionen

Das Handbuch verwendet die folgenden semantischen Auszeichnungen, Texthervorhebungen und Symbole.

Formatierungsdefinition	Art der Information
Kursiv	Produktnamen
Arial Narrow Kursiv	Wichtige Begriffe
<i>Arial Narrow Kursiv</i>	Eigenschaftselemente
"Fett Anführungszeichen"	Objekte, die Sie auswählen müssen, wie zum Beispiel Menüs, Schaltflächen, oder Begriffe in einer Liste.
"Fett Arial Narrow"	Befehlsnamen
GROSSBUCHSTABEN COURIER NEW	TASTENBEZEICHNUNGEN AUF DER TASTATUR. ZUM BEISPIEL SHIFT, STRG ODER ALT.
Courier New	Pfadangaben oder Dateinamen
Piktogramme	Beschreibung
	Dieses Symbol weist Sie darauf hin, dass wichtige Anweisungen folgen, die Sie unbedingt beachten sollten.
	Dieses Symbol weist Sie auf zusätzliche Informationen hin, die von generellem Interesse sind, und Sie bei der Benutzung der jeweiligen Anwendung unterstützen.
	Dieses Symbol weist Sie auf ein Beispiel hin, das Ihnen bei der Umsetzung hilft.

2.6 Support durch Wibu-Systems

Unseren Kunden steht ein professionelles Team bestens geschulter Mitarbeiter vom ersten Schritt an zur Seite. Der direkte Kundenkontakt ermöglicht es uns, Wünsche und Anregungen schnell zu erfüllen. Umfangreiche FAQ's für den Endanwender zum Thema *CodeMeter®* finden Sie auf unserer [CodeMeter® Support-Seite](#) sowie Informationen zum Thema *CodeMeter®*.

Anwender-Support

Wibu-Systems stellt eine kostenlose Anwender-Hotline für Ihre Endkunden zur Verfügung.

Entwickler (Kundensupport)

Sie erreichen uns (Baden-Württemberg-) werktags (Montag bis Freitag) durchgehend von 8.00 bis 17.00 unter der Telefonnummer +49-721-93172-14 oder per E-Mail unter support@wibu.com.

Support-Verträge mit erweiterten Leistungen auf Anfrage.

Bitte nennen Sie uns Ihre Kundennummer, damit wir Ihr Anliegen schnellstmöglich bearbeiten können.

Support Informationen

Zur Bearbeitung benötigen wir folgende Informationen:

- Art der Implementierung (automatisch/kunden-individuell)
- Betriebssystem
- Version der installierten *CodeMeter®*-Software
- Verwendete *CodeMeter®*-Hardware

- Detaillierte Fehlerbeschreibung

2.7 Über Wibu-Systems

WIBU-SYSTEMS AG wurde 1989 von Oliver Winzenried und Marcellus Buchheit gegründet und bietet als Technologieführer Lösungen zum Schutz und zur Lizenzierung digitaler Produkte an.

Die Unabhängigkeit vom eingesetzten Betriebssystem und die Vielfalt von Bauformen der Kopierschutz-Hardware *CmDongle* (USB, PC Card, Express Card|34, Compact Flash Card, SD- und MicroSD-Card, ASIC) ist gleichbedeutend mit Software-, Dokumenten-, Zugriffs- und Medienschutz für Anwendungen vom Smart Phone über Embedded Systeme, Desktop PCs und Server bis zum Cloud Computing. Wibu-Systems beschäftigt aktuell 100 Mitarbeiter, davon 80 am Hauptsitz in Karlsruhe. Als inhabergeführtes Unternehmen ist es finanziell unabhängig und verfolgt die Strategie der nachhaltigen Entwicklung.

Höchste Qualität bei Sicherheit, Zuverlässigkeit, Langlebigkeit sowie optimale Beratung und Service ist das Ziel. Die internationalen Aktivitäten des Unternehmens werden von Niederlassungen in Seattle (USA) sowie Schanghai und Peking (China), Verkaufsbüros in Belgien, Großbritannien, den Niederlanden, Portugal und Spanien sowie von Distributoren in über 25 Ländern unterstützt.

Mehr als 6.000 Hersteller weltweit setzen *WibuKey* und *CodeMeter®* ein. Wibu-Systems hilft Ihnen mehr zu verkaufen, indem der Verlust durch Raubkopien reduziert und durch flexible Lizenzmodelle der Kundenkreis vergrößert wird. *CodeMeter®* in der hardwarebasierten *CmDongle*-Variante bietet einzigartige Funktionen wie das Speichern tausender Lizenzen auch unterschiedlicher Anbieter, die einfache Lizenzübertragung online sowie die Kombination mit Flash-Massenspeicher. *CmActLicense* die software- und aktivierungsbasierte Lösung für Software im Niedrigpreissegment oder für Großunternehmen bindet die Lizenzinformation an Eigenschaften des Kundenrechners. Die *CodeMeter License Central* hilft dem Hersteller beim Erstellen, Verwalten und Ausliefern von Lizenzen und der Integration in Vertriebsprozesse und ERP-Systeme. *SmartShelter* erlaubt den sicheren Schutz von PDF-Dokumenten, *SmartShelter SDL* (Secure Data Layer) ermöglicht den Schutz beliebiger Dateien für beliebige Anwendungen. Authentifizierungslösungen umfassen den einfachen und sicheren Zugriff auf Webseiten und den Login für gehostete Anwendungen (*CSSi*). Im Media-Bereich werden digitale Videos, Flash-Animationen und Audio-Dateien sicher geschützt.

Wibu-Systems ist nach DIN EN ISO 9001:2008 zertifiziert und aktives Mitglied in internationalen und lokalen Verbänden wie BITKOM, VDMA oder SIIA und Standardisierungsgremien wie PCMCIA, USB Implementers Forum und SD Card Association. Wibu-Systems ist Microsoft Gold Certified Partner, Windows Embedded Partner sowie Partner in Entwicklerprogrammen von Apple, Adobe, Autodesk, Wind River und weiteren. Die Produkte wurden bereits mehrfach ausgezeichnet, beispielsweise bei SIIA Codie Awards als "Best Digital Rights Management" Lösung und beim internationalen IF Product Design Award. Das Unternehmen ist federführend in Forschungsprojekten mit Hochschulen und anderen Unternehmen, teilweise gefördert vom BMF und BMWi. Beispiele sind MimoSecco, in welchem eine flexible sichere Middleware-Lösung für Drittanwendungen im Bereich des Cloud Computings entwickelt wird und OpenID/Card, in dem eine Ausgabe von virtuellen Identitäten auf Basis des neuen Personalausweises für beliebige Identitäts-Provider ermöglicht werden soll.

So erreichen Sie uns weltweit

Deutschland	+49 (0) 721-93172-0	sales@wibu.de
USA	+1.425.775.6900	info@wibu.us

So erreichen Sie uns weltweit

China	Shanghai +86-(0) 21-55661790	info@wibu.com.cn
	Peking +86 (0) 10-82961560/61	info@wibu.com.cn
Großbritannien	+44 (0) 20 314 747 27	sales@wibu.co.uk
Irland	+44 (0) 20 314 747 27	sales@wibu.uk.co
Niederlande	+31 (0) 74 75 01 495	sales@wibu-systems.nl
Frankreich	+33 (0) 173030491	info@wibu.fr
Belgien	+32 (0) 3 400 03 14	sales@wibu.be
Spanien	+34 (0) 91 414 8768	sales@wibu.es
Andere Länder	+49 (0) 721-93172-0	sales@wibu.com

3 Softwareschutz und Lizenzmanagement

Mit *CodeMeter®* bietet Wibu-Systems eine sichere, hardware- und softwarebasierte Technologie zum Schutz und zur Lizenzierung von digitalen Inhalten für Smart Phone, Embedded Systeme, Desktop PCs, Server und Cloud Computing.

Im Folgenden wird in der Dokumentation der Begriff *CmDongle* stellvertretend für alle *CodeMeter®*-Hardware-Bauformen verwendet. *CmActLicense* steht für die rein software- und aktivierungsbasierte Variante des Schutz- und Lizenzierungssystems *CodeMeter®*. Werden beide Varianten angesprochen, so wird der Begriff *CmContainer* benutzt.

Die Schutzwirkung wird dadurch erzielt, dass eine durch *CodeMeter®* geschützte Software entweder nur mit der dazugehörigen Kopierschutz-Hardware (*CmDongle*) oder der software- und aktivierungsbasierter Variante (*CmActLicense*) funktioniert. Der *CmDongle* ist als USB Version (*CmStick/M/E/I/T/O*), als PC Card (*CmCard/M*, Cardbus, 32 Bit), als Express Card|34 (*CmCard/E*), als Compact Flash Card (*CmCard/F*) und als SD- und MicroSD-Card sowie als ASIC verfügbar.

WibuKey

Wibu-Systems bietet daneben *WibuKey* an. Auch *WibuKey* verschlüsselt Software und speichert Lizenzen für digitale Produkte. Die Hardware (*WibuBox*) ist vielfältig nutzbar und in verschiedenen Bauformen erhältlich, von der PC Card über den USB-Anschluss bis zu älteren Schnittstellen wie COM und LPT sowie als integrierter Schaltkreis (ASIC). Die meisten der Anwendungen, Schnittstellen und Werkzeuge, die für *CmDongle* zur Verfügung stehen, sind auch mit *WibuKey* und *CmActLicense* nutzbar. Für mehr Informationen besuchen Sie die Wibu-Systems Internetpräsenz www.wibu.com.

Schutz von Urheber- und Lizenzrechten

CodeMeter® stellt auf anwenderfreundliche Art die Einhaltung des Urheberrechts technisch sicher. Dabei ist *CodeMeter®* eine Technologie, die nicht nur Softwareschutz durch starke Verschlüsselung bietet, sondern die gleichzeitig auch die sichere Abbildung von Lizenzierungsstrategien erlaubt. Der Schutz basiert auf Verschlüsselungs- und Entschlüsselungsvorgängen, die sicher im *CmContainer* erfolgen.

Sie integrieren diesen Schutz über vielfältige Werkzeuge und Schnittstellen einmal in die Software und liefern dasselbe Programm zugeschnitten auf Ihre Kunden und für verschiedene Lizenzmodelle aus. Und die Software selbst läuft dann ausschließlich mit dem dazu passend programmierten *CmContainer*. Was bei Mitbewerbern heute als "Protect Once, Deliver Many™" bezeichnet wird, gibt es bei Wibu-Systems als Selbstverständlichkeit seit Firmengründung.

Wie die untenstehende Abbildung zeigt, erfüllt *CodeMeter®* dabei alle Anforderungen, die an eine sichere und effektive Technologie im Umfeld von Softwareschutz und Lizenzmanagement gestellt werden.

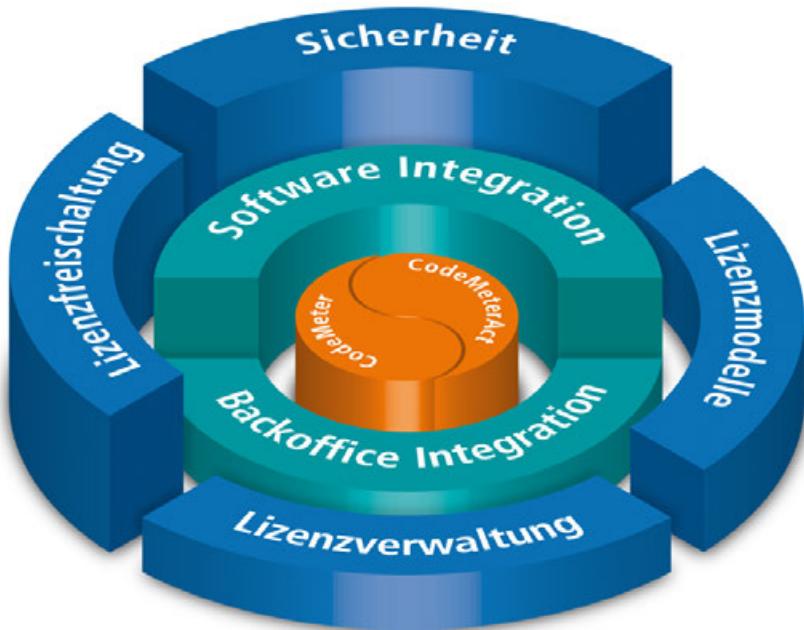


Abbildung 4: Softwareschutz und Lizenzmanagement mit CodeMeter® im Überblick

Sicherheit

- ⌚ Zum Schutz setzt CodeMeter® modernste Verschlüsselungsalgorithmen ein. Dazu gehören der AES (Advanced Encryption Standard) mit 128 Bit Schlüssellänge und der ECC (Elliptic Curve Cryptography) mit 224 Bit Schlüssellänge für asymmetrische Verschlüsselungen und Signaturen. Zudem wird für asymmetrische Verschlüsselungen auch RSA mit 2048 Bit Schlüssellänge eingesetzt.
- ⌚ Alle verwendeten Schlüssel sind sicher im *CmContainer* gespeichert. Der Empfänger kann die Schlüssel nicht aus dem *CmContainer* auslesen. Zusätzlich ist auch die Verwendung von wechselnden Schlüsseln möglich, indem zur Laufzeit der Anwendung weitere Informationen in die Ver- und Entschlüsselungsvorgänge einfließen. Die Schlüssel können auch mittels eines Zufallsgenerators innerhalb des *CmContainer* erzeugt werden.
- ⌚ Sicherer Master-Dongle, die Firm Security Box (FSB) zum Programmieren von Lizenzen in den *CmContainer*. Die FSB ist für jeden Lizenzgeber eindeutig.
- ⌚ Der Softwareschutz von Wibu-Systems hat sich vier Mal erfolgreich der internationalen Hacker-Szene gestellt.
- ⌚ Der *CmDongle* ist zusätzlich gegen alle bekannten Analysemethoden geschützt (z.B. Elektronenmikroskop, DPA) und die Kommunikation zwischen *CmDongle* und dem PC erfolgt ebenfalls verschlüsselt.
- ⌚ Die Entschlüsselung der geschützten Anwendung (Quelltext und Ressourcen) erfolgt im Arbeitsspeicher nie vollständig, sondern nur bei Bedarf. Diese "On-Demand-Decryption" schützt effektiv vor unbefugtem Zugriff.

fektiv vor Memory Dumping und verhindert die Extraktion ungeschützter Versionen der Anwendung.

- ⇒ *CodeMeter®* bietet mehrstufigen, kombinierbaren und ineinandergrifenden Schutz über verschiedene Ebenen hinweg:
 - Automatischer Schutz von Anwendungen mit *AxProtector* als sicherer Basisschutz ohne Eingriffe in den Quelltext. Dieser umfasst u.a. Überprüfungen der Lizenzenschaften zur Laufzeit der Anwendung, effektive Anti-Debugging-Maßnahmen, Modifikation von Ressourcen sowie Sperren des *CmContainer* bei Erkennung von Hack-Versuchen.
 - Individueller, erweiterter Schutz während der Anwendungsentwicklung mit *IxProtector*. Es werden "echte" Quelltext-Fragmenten über Schnittstellen (*Softwareschutz-API*, *WUPI*) und Sicherheitsmechanismen ver- und entschlüsselt.
- ⇒ Zusätzliche technisch ausgefeilte Sicherheitsmechanismen, die in den *CodeMeter®*-Technologien, Werkzeugen und Schnittstellen integriert sind und ständig sicherheitstechnisch überarbeitet werden.
- ⇒ Manipulationssicherer Schutz von Nutzungszeiträumen, Aktivierungs- und Ablaufterminen für Anwendungen über eine interne Uhr im *CmContainer* und einen Mechanismus, der sichere Zeitzertifikate zur Verfügung stellt.

Lizenzabbildung

- ⇒ Programmierung von Lizenzinträgen in den *CmContainer* mit Optionen:
 - Versehen der Lizenz mit beschreibenden Informationen.
 - Festlegen der Anzahl gleichzeitiger Nutzer in einem Netzwerk über eingebaute Netzwerkunterstützung (LAN und WAN).
 - Setzen von Aktivierungs- und Ablaufdaten mit Start bzw. Ende zu einem fixen Zeitpunkt sowie Festlegen einer Nutzungsdauer mit Start zu einem variablen Zeitpunkt.
 - Anlegen und Anzeigen von benutzerspezifischen Informationen.
 - Setzen von unabhängigen Zählern, die bei bestimmten Aktionen definiert heruntergezählt werden können.
 - Verwenden einer Feature Map zur Freischaltung einzelner Module einer Anwendung bei Belegung nur eines einzigen Lizenzintrages, oder zur Versionsverwaltung.
 - Verwenden von Wartungszeiträumen zur Gewährung von Support- und Wartungsleistungen für die Software-Nutzung.
 - Verwenden zusätzlicher binärer Informationen über diverse Datenfelder, auch als Speicherort für alternative Schlüsselquellen.
- ⇒ Beliebig miteinander kombinierbare Eigenschaften schaffen die Grundlage für die Abbildung jeder nur denkbaren Lizenzstrategie.

Lizenzstrategie	Lizenzmodelle
Standard Lizenzmodelle	Einzelplatz-Lizenz Floating- / Concurrent-Lizenzen Demo-Versionen Modulare Lizenzen
Nutzungsabhängige Lizenzmodelle	Miete, Leasing Software-Wartung / Software Assurance

Lizenzstrategie	Lizenzmodelle
	Tatsächlich ausgeführte Funktionen (pay-per ...)
Erweitertes Lizenzmanagement	Downgrade- / Versionsmanagement Overflow Cold- / Hot-Standby Named User Lizenzen Rechnergebundene Lizenzen Lizenzausleihe (Lizenzen zum Mitnehmen) Volumenlizenzen

- ⌚ Der *CodeMeter®* SmartCard Chip mit 60/384 kByte-Speicher erlaubt die Programmierung von bis zu 6.000 Lizenzinträgen in einen *CmDongle*.
- ⌚ Herstellerunabhängige Nutzung und Verwaltung von Lizenzinträgen über die eindeutige und sichere Trennung separater Lizenzcontainer im *CmDongle*. Mehrere Software-Hersteller können sich dadurch einen *CmDongle* teilen.

Lizenzverwaltung

- ⌚ Effizientes Ticketsystem *CodeMeter License Central* in einer *Desktop / Internet Edition*. Die Eingabe von Auftrags-, Kunden- und Artikelnummer erzeugen passende Tickets, die für weitergehende Aufgaben in Vertrieb und Produktion verwendet werden.
- ⌚ Integration der Lizenzverwaltung in Vertriebs- und Support-Prozesse durch *CodeMeter License Central Internet* mit den Schnittstellen: Internet-Gateway zum Kunden, Connectoren zu ERP/CRM-Systemen und Connectoren zu Online-Shops.
- ⌚ Datenaustausch über SOAP (xml-basiert) mit minimalen Anpassungen im Online Shop oder dem ERP-System. Vorhandene Lizenzgeneratoren und kundenspezifische Auftragsfelder können meist sofort übernommen werden.

Lizenzfreischaltung

- ⌚ Neben der lokalen Programmierung auch sicheres Programmieren, Modifizieren oder Löschen der vollständigen Inhalte und Optionen von Lizenzen in *CmContainern* über den Austausch von Dateien.
- ⌚ Dateibasierte Fernprogrammierung mit *CmFAS* (*CodeMeter Field Activation Service*) oder SOAP-basiert über *CodeMeter License Central*.

Software Integration

- ⌚ Automatische Integration des Schutzes in die Software als Basisschutz über automatisches Verschlüsseln von ausführbarem Quelltext ohne Eingriffe in den Quelltext mit *AxProtector*.
 - Einfach zu bedienende Anwendungsoberfläche mit den wichtigsten Einstellungsoptionen für die Verschlüsselung verschiedener Projekttypen (Windows 32-Bit/64-Bit, Mac OS X, Java, .NET, Linux etc.).
 - Frei konfigurierbare Meldungsdialoge.
 - Erzeugen und Weiterverwendung einer Kommandozeile für *AxProtector*-Kommandozeile.
- ⌚ Individuelle Integration des Schutzes in die Software als zusätzlicher Schutz ergibt höchste Flexibilität und zusätzlichen Schutz zur Laufzeit einer Anwendung.
 - Definition und Schutz einzelner Bereiche und Funktionen im Quelltext und deren anschließende Verknüpfung mit unterschiedlichen Lizenzinträgen zur Laufzeit der Anwendung mit der Schutztechnologie *IxProtector*, die in *AxProtector* integriert ist.

Wibu-Systems empfiehlt zur Erhöhung des Schutzes die Kombination von automatischer und individueller Integration.



Außerdem werden Sicherheitsmechanismen von *AxProtector* wie *IxProtector* ständig weiterentwickelt und verbessert. Die Anwendung braucht nach Aktualisierungen nicht neu kompiliert, sondern lediglich mit *AxProtector* bzw. *IxProtector* neu verschlüsselt werden.

- Ent- und Verschlüsseln von *IxProtector* geschützten Bereichen während der Laufzeit mit WUPI (*WIBU Universal Protection Interface*). Dieses schlanke, nur wenige, aber elementare Funktionen umfassende Softwareschutz-API ist universell für viele Programmiersprachen einsetzbar.
- ☞ Zusätzliche Anforderungen (Ver- und Entschlüsselung von Daten, Personalisierung, Auslesen weiterer Daten) erfüllt das *CodeMeter Kern-API* mit umfangreichen Funktionen. Über den interaktiven *CodeMeter API Guide* erhalten Sie schnell den passenden Quelltext.

Back Office Integration

- ☞ Einfache und schnelle Erstellung und Programmierung von Lizenzen während der Entwicklung, oder beim Testen von Lizenzierungsstrategien mit der graphisch-intuitiven *CodeMeter License Editor*-Oberfläche, wenn nur ein kleine Anzahl von *CmDongles* im Einsatz ist.
- ☞ Kommandozeilen-Programmierung mit Skripten und Batch-Dateien für die Massenproduktion und Automatisierung von Tests mit *CmBoxPgm*. Die Programmierung von Abläufen kann in einem Durchgang auf mehrere *CmContainer* angewendet werden.
- ☞ Erzeugen, Verwalten und Ausliefern von Lizenzen mit dem effizienten Ticketsystem *CodeMeter License Central* in einer *Desktop* und *Internet* Edition.
- ☞ Weitere Anforderungen, die nicht mit den zur Verfügung stehenden Erstellungs-, Programmierungs- und Verwaltungswerkzeugen umgesetzt werden können, sind über das zugrundeliegende *Programmier-API (HIP, High Level Programming API)* in eigene Anwendungen integrierbar.

3.1 CmDongle: Vielfalt der CodeMeter-Bauformen

CmDongle ist die hardwarebasierte Variante der Schutz- und Lizenzierungstechnologie *CodeMeter®*. Hier sind die Lizenzen und die für die Ver- und Entschlüsselung notwendigen Schlüssel hochsicher im SmartCard Chip des Dongle abgespeichert.

CmDongle ist in einer großen Vielfalt für unterschiedliche Schnittstellen verfügbar:

Bauform	Beschreibung
	CmStick Standard-Edition für die USB Schnittstelle in Kunststoff ohne zusätzlichen Flash-Speicher ¹⁾
	CmStick ME Metall-Edition, in edler Metallausführung ohne zusätzlichen Flash-Speicher ¹⁾
	CmStick/M Version beider Editionen mit zusätzlichem Flash-Speicher, um die Software direkt vom <i>CmDongle</i> mobil starten zu können
	CmStick/T Version beider Editionen mit batteriebetriebener interner Uhr ohne zusätzlichen Flash-Speicher ¹⁾

Bauform	Beschreibung
	CmStick/C Kompakt-robuste Edition ohne zusätzlichen Flash-Speicher. ¹⁾
	CmStick/I USB Flash Disk Modul mit einer 2X5 Pfostenbuchse im 2,54 mm Raster-Standardformat
	CmStick/CI USB Flash Disk Modul mit einer 2X4 Pfostenbuchse im 2,00 mm Raster
	CmCard PC Card, 32 Bit, mit Flash Memory
	CmCard/E CmCard als Express Card mit 34 Standard Interface
	CmCard/CF CF Card (Compact Flash) mit 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 und 16 GB)
	CmASIC ASIC für die Integration in eigene Hardware

Abbildung 5: CmDongle-Bauformen

¹⁾ Für diese Bauform ist alternativ eine Anmeldung als Human Interface Device (HID) am System möglich. Für Voraussetzungen und Details siehe [hier](#)⁵¹⁰.

3.2 CmActLicense: Bindung und Aktivierung

CmActLicense ist die softwarebasierte Variante der Schutz- und Lizenzierungstechnologie *CodeMeter®*. Hier sind die Lizenzen und die Schlüssel, die für die Ver- und Entschlüsselung notwendig sind, in einer mit kryptographischen Verfahren abgesicherten und signierten *CmActLicense*-Lizenzdatei abgespeichert. Dieser virtuelle *CmContainer* ist eindeutig und ausschließlich an einen bestimmten Rechner oder ein bestimmtes Gerät gebunden.

Die eindeutige Bindung wird über einen digitalen "Fingerabdruck" gewährleistet, der aus bestimmten Hardware-Merkmalen eines Rechners oder eines Gerätes gebildet wird. Dies stellt sicher, dass *CmActLicense*-Lizenzen ausschließlich für den so identifizierten Rechner oder das Gerät Gültigkeit besitzen und nicht übertragbar sind.

3.2.1 CmActLicense: Bindungsschemata

Bindungsschemata

Die Strukturierung, welche Hardware-Merkmale wie zur Bindung herangezogen werden, erfolgt über die Verwendung von Bindungsschemata. Diese Schemata lassen sich in drei Kategorien ordnen: dynamisch gewichtet über *CodeMeter® SmartBind*, explizit über *Binding Extension* und ohne Bindung über das Schema *None*.

CodeMeter® SmartBind

Die dynamisch gewichtete Bindung über die Verwendung des Schemas [SmartBind](#)³⁶⁹ optimiert die fortbestehende Gültigkeit von Lizenzen, wenn sich für einen Rechner oder ein Gerät die Hardware-Merkmale ändern.

SmartBind bezieht dabei viele Hardware-Merkmale ein und gewichtet sie auf der Grundlage interner Algorithmen, sodass eine Änderung eines bestimmten Hardware-Merkmales nicht gleich zwangsläufig zu einer Reaktivierung einer Lizenz führen muss. Der Rechner oder das Gerät werden weiterhin eindeutig erkannt.

SmartBind bietet eine einfache und zugleich sichere Art, eine Lizenz an einen Rechner oder ein Gerät zu binden. Durch die Vielzahl der dynamisch gewählten Merkmale bietet das Schema sowohl Zuverlässigkeit (Reliability), als auch Schutz gegen Manipulation (Security). Für mehr Information über das Verfahren siehe das separate Dokument "[SmartBind Whitepaper](#)", das Sie von der Wibu-Systems Webseite herunterladen können.

Optional kann in begründeten Einzelfällen für *SmartBind* zusätzlich eine Toleranzgrenze gesetzt werden. Sie bestimmt die zulässige Abweichung zwischen der Ausgangskonfiguration eines Rechners oder Gerätes zum Zeitpunkt der Lizenzaktivierung und der aktuellen Konfiguration.

 Wibu-Systems empfiehlt *SmartBind* und die eingestellte Toleranzgrenze als Standard-Bindungsschema zu verwenden.

Für die Programmierung von *CmActLicense*-Lizenzen mit *SmartBind* über *CmBoxPgm* siehe [hier](#)
³⁷²

Für begründete Einzelfälle unterstützt *CmActLicense* auch Bindungsschemata, die sich entweder auf bestimmte feste³⁷⁰ oder konfigurierbare³⁷⁰ Hardware-Merkmale eines Rechners oder eines Gerätes beziehen können. Wibu-Systems empfiehlt jedoch, vor Nutzung dieser Optionen Wibu-Systems Support zu kontaktieren.

CodeMeter® Binding Extension

Für Fälle, in denen die Bindung von Lizzenzen an herstellerspezifische Merkmale eines Gerätes oder eigene, sichere Merkmale eines eigenen Zielsystems - etwa im Embedded-Bereich - gewünscht wird, steht das Bindungsschema [Binding Extension](#)³⁷⁰ zur Verfügung.

Bei Verwendung dieser Merkmale liefert der Software-Hersteller mit dem Installationsprogramm seines Produktes zusätzlich ein signiertes Plugin aus. Dieses Plugin wird von *CodeMeter Lizenzserver* bei Bedarf geladen und stellt die Funktionalität zur Ermittlung der Merkmale bereit. Auf diese Weise können alle denkbaren Merkmale, beispielsweise eines Endkundenrechners oder eines Embedded-Zielsystems, als Bindungsmerkmal für *CmActLicense*-Lizenzen verwendet werden.

Für mehr Information siehe das separate Dokument "CmActLicense Binding Extension", das Sie auf Anfrage von Wibu-Systems erhalten.

Wenn Sie das Schema *Binding Extension* für das individuelle Binden einer *CmActLicense* an eine eigene Hardware verwenden, können Sie ab der *CodeMeter®* Version 4.40 bei einem bekanntem Bindungswert [vorberechnete Lizenzdateien](#)³⁶⁸ erstellen und ausliefern. Der Schritt der Erzeugung einer Lizenzanforderungsdatei auf dem Zielsystem bei der späteren Aktivierung ist damit nur noch optional.

None-Bindungsschema

Über die Verwendung des Bindungsschemas [None](#)³⁷⁰ ist es möglich, geschützte Software ohne Bindung an einen bestimmten Rechner oder ein bestimmtes Gerät auszuliefern.

Dies ist zum Beispiel der Fall, wenn die Bindung einer Lizenz zwar zeitlich befristet, aber gleichzeitig für beliebige Rechner oder Geräte gültig sein soll, etwa für Test- oder Demo-Zwecke. Dazu bietet Wibu-Systems das [Trial License](#)³⁷⁴-Lizenzmodell an, mit der Sie Demo-Lizenzen mit einer Laufzeit von maximal 90 Tagen erzeugen. Diese Lizenzen verlieren nach diesem Zeitpunkt ihre Gültigkeit und sind nicht wieder einspielbar.

Ein weiterer Anwendungsfall ist die Erstellung von zeitlich unbefristeten und wiedereinspielbaren Lizenzen für beliebige Rechner oder Geräte, wenn insbesondere der Schutz gegen ein Reverse Engineering im Vordergrund steht. Dazu bietet Wibu-Systems das [Protection Only](#)³⁷⁵-Lizenzmodell an.

Für beide 'None-Bind'-basierte Lizenzmodelle ist ein separater Lizenzeintrag in der [Firm Security Box](#)³⁷ (FSB) erforderlich, der für den Evaluierungs-Firm Code 5010 im SDK enthalten ist.

Zusätzliche Optionen für die Gültigkeit von CmActLicense-Lizenzen

Zusätzlich zu den Bindungsschemata ist es möglich, weitere Optionen festzulegen, die bei der Aktivierung einer *CmActLicense*-Lizenz greifen. Die folgende Tabelle listet diese Optionen auf:

Option	Beschreibung
Betriebssysteme	Mit dieser Option geben Sie an, auf welchem Betriebssystem(en) <i>CmActLicense</i> -Lizenzen verwendet werden dürfen.
Virtuelle Maschinen	Mit dieser Option erlauben Sie, dass <i>CmActLicense</i> -Lizenzen innerhalb virtueller Maschinen genutzt werden dürfen.
Mehrfache Lizenzeingabe	Mit dieser Option erlauben Sie, dass eine <i>CmActLicense</i> -Freischaltdatei beliebig oft auf einem Rechner oder Gerät eingespielt werden darf.
<i>CodeMeter® Runtime</i>	Mit dieser Option können Sie eine minimal erforderliche <i>CodeMeter® Runtime</i> -Version setzen.

3.2.2 CmActLicense: Aktivierung

Die Aktivierung von *CmActLicense*-Lizenzen lehnt sich in weiten Zügen an das standardisierte *CodeMeter®*-Verfahren zur dateibasierten Fernprogrammierung von [CmDongles](#)²⁸ an. Das Verfahren basiert auf dem Transfer von Lizenzanforderungs- und Lizenzaktualisierungsdateien.

Lizenzanforderungsdateien (Kontext-Dateien) enthalten dabei den jeweils aktuellen Stand von Lizenzinformationen auf Seiten des Kunden und Lizenzaktualisierungsdateien (Update-Dateien) werden vom Hersteller genutzt, um Lizenzen zu aktualisieren und sie für eine Aktivierung durch den Kunden zur Verfügung zu stellen.

Bevor im Fall von *CmActLicense*-Lizenzen eine Aktivierung durchgeführt werden kann, müssen zunächst die konkreten Hardware-Merkmale eines Rechners oder eines Gerätes ermittelt werden. Der Hersteller erstellt dazu die Lizenzinformationsdatei (*.lif). Diese Datei entspricht einem leeren Lizenzcontainer beinhaltet aber Angaben über [Bindungsschema](#)²⁸ und [zusätzliche Aktivierungsoptionen](#)²⁹, die verwendet werden sollen, um eine Lizenz eindeutig an den Rechner oder das Gerät zu binden.

Indem der Kunde diesen leeren Lizenzcontainer einspielt, werden zum einen die notwendigen Informationen über den Rechner oder das Gerät ermittelt und zum anderen die Grundlage für die Bindung einer Lizenz über einen eindeutigen digitalen "Fingerabdruck" gelegt. Die initiale Lizenzanforderungsdatei, die der Kunde erstellt, enthält dann alle notwendigen Lizenzinformationen, die der Hersteller benötigt, um wiederum eine *CmActLicense*-Lizenz zu programmieren, die dann eindeutig an den bestimmten Rechner oder das bestimmte Gerät gebunden ist und nur für diesen Rechner oder dieses Gerät aktiviert werden kann. Der Transfer dieser Bindungs- und Aktivierungsinformationen findet über die Lizenzaktualisierungsdatei statt, die der Kunde einspielt.

Die folgende Abbildung illustriert diesen Vorgang:

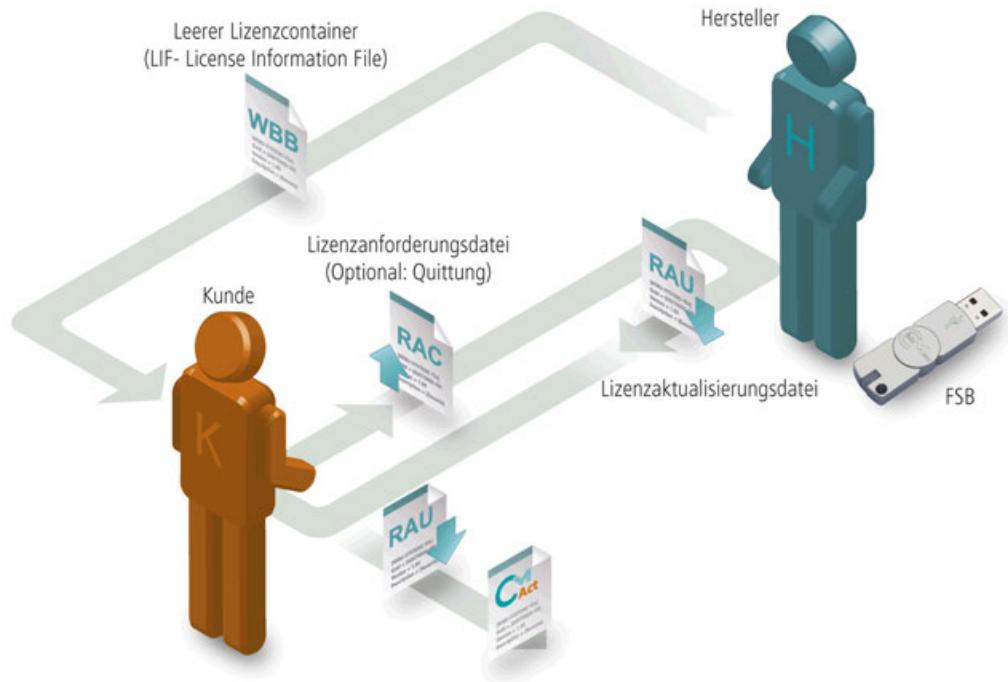


Abbildung 6: Aktivierung von *CmActLicense*-Lizenzen

3.3 Vielfalt der Betriebssysteme für CodeMeter

CodeMeter® ist verfügbar für viele Betriebssysteme und Laufzeitumgebungen wie Windows Betriebssysteme 32-Bit/64-Bit, MacOS X, Linux 32-Bit/64-Bit, Java, Sun Solaris 10, .NET.

Betriebssystem	CodeMeter
Windows XP	✓
Windows Vista	✓
Windows 7	✓
Windows 8	✓
Windows 2000 Server	✓
Windows 2003 Server	✓
Windows 2008 Server	✓

Betriebssystem	CodeMeter
MacOS X	✓
Linux	✓
Sun Solaris 10	✓
Windows XP Embedded	✓
Windows CE 5.0	✓
Windows CE 6.0	✓
VxWorks	✓

3.4 Zusätzliche Eigenschaften

Zusätzlicher Flash-Speicher und Mobile Anwendungen

- In der Version mit zusätzlichem Flash Memory ist ein *CmContainer CmDongle* und Speichermedium in einem. Das direkte Ausliefern der Software auf der *CodeMeter®*-Hardware ist dann möglich. Die Software startet ohne Installation direkt vom *CmDongle*.
- *CmDongle* verwendet den industriauglichen SLC-Speicher (Single Level Cell). Dieser ist schneller, langlebiger und robuster gegen Datenverluste als der im Consumer-Bereich verwendete MLC-Speicher (Multi Level Cell).

Alle Treiber sind schon da

- Treiberlose Benutzung für viele Plattformen über *CodeMeter Lizenzserver*. Der im Hintergrund arbeitende Dienst kommuniziert nach unten über den betriebssystemeigenen USB bzw. Mass Storage Device Treiber mit *CmDongle/CmActLicense* und stellt nach oben die Schnittstelle für das *CodeMeter Kern-API* zur Verfügung.

Lizenzserver Einstellungen

- Lokale Konfigurationseinstellungen für *CodeMeter Lizenzserver* werden mit *CodeMeter Kontrollzentrum* vorgenommen. *CmContainer* können hierbei sowohl lokal, als auch im Netzwerk angeschlossen sein. *CodeMeter Lizenzserver* ist standardmäßig als Dienst bzw. Daemon (Linux, Mac) installiert und wird daher bei jedem Systemstart automatisch gestartet. Ist der Dienst gestartet, so können andere Programme auf Lizenzen zugreifen, die in *CmContainer* gespeichert sind, und geschützte Datenbereiche in *CmContainer* verwenden.

Anzeige der Lizenzeinträge

- Information über verbundene *CmContainer* und die darin programmierten Lizenzeinträge zeigt *CodeMeter WebAdmin* an und bietet vielfältige Konfigurations- und Analysemöglichkeiten.

3.5 CodeMeter als Token

Der *CmDongle* wird meist für die Entschlüsselung geschützter Software und zum Lizenzmanagement verwendet. *CodeMeter®* ist aber auch als Zertifikatsspeicher für gängige Formate wie X.509 einsetzbar. Um ein kryptografisches Gerät als Token zu verwenden, muss dieses Gerät in der Lage sein, geheime

Schlüssel sicher aufzubewahren und zu verwenden. *CodeMeter®* kann mit der Speichermöglichkeit im Secret Data-Feld seit jeher beliebige Daten und damit auch Schlüssel sicher aufbewahren. Weiterhin ist in den aktuellen Firmware-Versionen die Verwendung des anerkannten und standardmäßig verwendeten, asymmetrischen Kryptografie-Algorithmus RSA mit einer Schlüssellänge bis 2048 Bit vorhanden. Damit sind alle Voraussetzungen gegeben, *CodeMeter®* als Token einzubinden. Einzig fehlte bislang die Möglichkeit, diese Eigenschaften im Rahmen der üblicherweise auf den Systemen verwendeten Schnittstellen zu benutzen. Durch die Zusammenarbeit mit charismathics konnte diese Lücke nun auch geschlossen werden, so dass einem Einsatz von *CodeMeter®* als Token in vielen Applikationen und Anwendungsfällen nichts mehr im Wege steht.

Asymmetrische Verschlüsselung

Bei der asymmetrischen Verschlüsselung ist der private Schlüssel dem Besitzer bekannt, der öffentliche Schlüssel kann völlig frei verteilt werden. Die wesentliche Eigenschaft der Asymmetrie ist, dass zwar aus dem privaten Schlüssel der öffentliche Schlüssel berechnet werden kann, aber der umgekehrte Weg, die Berechnung des privaten Schlüssels aus dem öffentlichen Schlüssel, selbst mit großem Aufwand nicht möglich ist.

Public Key Infrastruktur

Bei der Verwendung eines Tokens geht es darum, sich selbst gegenüber anderen auszuweisen, etwas nachprüfbar zu unterschreiben oder auch zu verschlüsseln. Der kritische Punkt ist hier die Frage: Wem kann ich wie weit vertrauen? Daher ist eine Grundvoraussetzung eine vertrauenswürdige Public Key-Infrastruktur (PKI), die es für alle Beteiligten möglich macht, die Authentizität des Partners zu verifizieren. Hierzu müssen die verwendeten Schlüssel von einer Ausgabestelle beglaubigt werden, diese Beglaubigung wiederum kann von den anderen Partnern überprüft werden und stellt so sicher, dass man wirklich mit dem Partner kommuniziert, mit dem man zu kommunizieren glaubt. Diverse Dienstleister haben eine solche Infrastruktur aufgebaut, und auf Grund des verwendeten Standards der X.509-Zertifikate können von diesen Dienstleistern ausgegebene Zertifikate in und mit *CodeMeter®* gespeichert und verwendet werden.

Einsatzgebiete

Beim Einsatz von *CodeMeter®* als Token im Rahmen einer PKI wird der private Schlüssel zusammen mit anderen Angaben im Rahmen eines X.509-Zertifikats im *CmDongle* abgespeichert. Mit einem solchen Zertifikat und den damit möglichen kryptografischen Verfahren können Sie verschiedene Dinge tun wie zum Beispiel den VPN-Zugang absichern, E-Mails signieren und/oder verschlüsseln, oder eine Zugangskontrolle mit starker Zwei-Faktor-Authentifizierung realisieren. Ebenso können Sie ein zertifikatsbasiertes Windows-Login verwenden, sich bei webbasierten Anwendungen authentifizieren (Software as a Service) oder eine unternehmensweite Single-Sign-On-Lösung für Windows aufsetzen.

Vermitteltätigkeit

Die Middleware CSSI (Charismathics Smart Security Interface) von charismathics stellt nach oben alle Dienste für Zugriff, Identifikation und Authentifizierung zur Verfügung und vermittelt diese nach unten an *CodeMeter®* weiter, das hier die notwendigen Funktionen bereitstellt und somit die Speicherung und Verwendung der Zertifikate ermöglicht. Hierbei unterstützt die CSSI Middleware die Windows-proprietäre CSP-Schnittstelle (Crypto Service Provider) sowie die generische PKCS#11-Schnittstelle vollständig. So mit stehen diese Dienste plattformübergreifend für Windows, Mac OS X und Linux zur Verfügung.

Token und Dongle ohne Middleware

Für proprietäre Anwendungen können Sie *CodeMeter®* als Dongle und Token zugleich auch ohne die

CSSI Middleware benutzen. Wenn Sie sich selbst um das Schlüsselmanagement kümmern, können Sie mit eigenen oder vorhandenen Schlüsseln über den ECIES-Algorithmus im *CodeMeter Kern-API* signieren und verschlüsseln.

3.6 CodeMeter auf Embedded Systemen

Wibu-Systems bietet *CodeMeter® Compact Driver* für Embedded-Systems an, der *CodeMeter®* Lizenzserver ersetzt und den direkten Zugriff auf *CmDongle*- oder *CmActLicense*-Lizenzen aus der eigenen Software heraus erlaubt.

CodeMeter® Compact Driver ist als ANSI C Quellcode verfügbar oder als statische Bibliothek und kann auf dem Zielsystem kompiliert werden. *CodeMeter® Compact Driver* zeichnet sich dadurch aus, dass er modular für Ihr Projekt von Wibu-Systems zusammengestellt wird und somit genau für den jeweiligen Einsatz optimiert ist. Speziell auf einem eigenen oder einem Embedded-Betriebssystem ist er die passende Alternative.

Eine Integration von *CodeMeter®* in das Echtzeit-Betriebssystem VxWorks von Wind River und in die Automatisierungssoftware CODESYS SPS von 3S-Smart Software Solutions GmbH ist verfügbar.

4 Das CodeMeter-Konzept

Eine Lizenz wird in *CodeMeter®* zunächst durch zwei eindeutige Zahlen identifiziert: durch einen Firm Code und einen Product Code.

Den Firm Code erhalten Sie von Wibu-Systems. Diese Zahl identifiziert jeden Lizenzgeber individuell und wird nur einmal vergeben.

Der Product Code ist eine Zahl, die Sie frei wählen können. Damit identifizieren Sie die Produkte, die Sie schützen und lizenziieren möchten.



Wollen Sie mehr als nur ein Produkt schützen und lizenziieren, können Sie einen Product Code für jedes einzelne Produkt verwenden. Umfangreiche Produkte können aber auch mehrere Product Codes gleichzeitig haben, z.B. Programme mit vielen Modulen.

Die Einträge im *CmContainer* sind in einer baumartigen Struktur hierarchisch in logische Bereiche gegliedert.

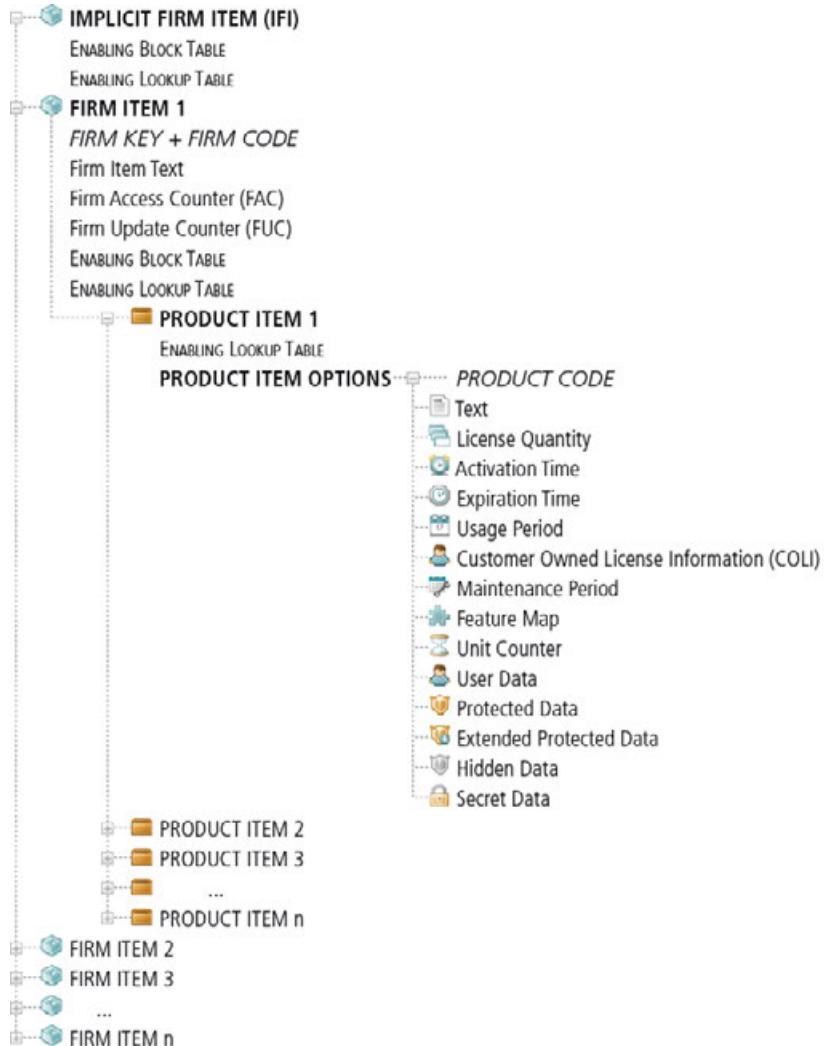


Abbildung 7: CmContainer Lizenz-Eintragsstruktur

Auf der obersten Ebene befinden sich die Firm Item-Einträge, die für jeden einzelnen Lizenzgeber separat den jeweiligen Firm Code enthalten.

Firm Item Options (FIO)

Weitere Eigenschaften, die Firm Item Options (FIO) , benennen jede Eintragsebene und zählen z.B. mit, wie oft sie durch eine Aktualisierung oder einen Zugriff adressiert wurde (Firm Item Text, Firm Update Counter, Firm Access Counter).

Jeder Lizenzgeber besitzt seinen eigene Firm Item-Eintragsebene im Lizenzcontainer und nur er ist in der Lage auf seiner Eintragsebene, für sein Firm Item, Lizenzeinträge für Produkte zu erzeugen, zu ändern oder zu löschen.



Das ist der Grund dafür, dass Lizenzen in einem einzelnen *CmDongle* auch herstellerunabhängig verwaltet werden können. Verschiedene Hersteller können sich einen *CmDongle* teilen und damit Kosten sowie Bearbeitungsaufwand sparen. Der Lizenznehmer hat den Vorteil, dass er alle Lizenzen in einem *CmDongle* verfügbar hat und so nur einen Anschluss belegt. In einem *CmDongle* können insgesamt bis zu 6.000 Lizenzeinträge gespeichert werden.

Implicit Firm Item (IFI)

Das Implicit Firm Item auf der Ebene der Firm Items ist ein besonderer Eintrag. Dieser logische Bereich der Eintragsstruktur ist für jeden Besitzer eines *CmContainers* frei zugänglich. Einzige Voraussetzung ist hier das gültige Kennwort für den *CmContainer*.

Product Items, die Lizenzeinträge

Die Lizenzeinträge für die Produkte selbst liegen auf der Ebene der Product Items. Die Firm Item-Ebene kann ein oder mehrere Lizenzeinträge, d.h. Product Items enthalten.

Auf der Ebene der einzelnen Lizenzeinträge, der Product Items, liegen ebenfalls die Product Item Options. Sie enthalten den Product Code, der einen Lizenzeintrag eindeutig definiert und weitere Optionen, die die eigentlichen Eigenschaften einer Lizenz festlegen, z.B. wieviele Lizenzen gleichzeitig in einem Netzwerk genutzt werden dürfen, wie lange eine Lizenz gültig ist, welche Funktionen benutzt und abgerechnet werden, usw. Außerdem stehen unterschiedliche Dateneinträge zur Verfügung, die zusätzliche Informationen enthalten und sich jeweils in ihren Zugriffsberechtigungen unterscheiden. Für einen Überblick und Beschreibung der [Product Item Options](#)³⁸.

Diese optionalen Eigenschaften lassen sich für jede Lizenz beliebig miteinander kombinieren und schaffen damit die Grundlage für die Abbildung jeder nur denkbaren Lizenzstrategie (zu [Lizenzmodellen](#)⁵² und Umsetzungsbeispielen).

Für die Eintragsstruktur eines *CmContainers* gilt, dass die Firm Item-Ebenen ausschließlich mit einem eindeutig identifizierten Firm Code erzeugt werden, und dass keine Lizenzeinträge außerhalb dieser Ebene erzeugt, bearbeitet oder gelöscht werden können. Um dies sicherzustellen, erhalten Sie von Wibu-Systems neben dem Firm Code zusätzlich eine Firm Security Box (FSB), die an Ihren Firm Code gebunden ist.

Firm Security Box

Die Firm Security Box (FSB) ist ein Art Master-*CmContainer*, der erforderlich ist, um Lizenzen mit Ihrem Firm Code in *CmContainer* programmieren zu können.

Dadurch stellt Wibu-Systems sicher, dass nur Sie als Besitzer der Firm Security Box in der Lage sind andere *CmContainer* mit Ihrem Firm Code zu programmieren. Das Programmieren selbst ist durch Kryptographie abgesichert, und die dazu benötigten Schlüssel sind sicher in Ihrer FSB gespeichert.

Firm Key

Und schließlich erhalten Sie von Wibu-Systems einen Firm Key. Der Firm Key ist ein geheimer Schlüssel und beeinflusst fast alle Ver- und Entschlüsselungsvorgänge von Lizenzen, deren Authentifizierung sowie das

Anlegen, Aktualisieren und Löschen von Lizenzinträgen auf der Ebene der Product Items.

Der Firm Key wird initial in Ihrer Firm Security Box ausgeliefert. Wenn Sie aus einem höheren Sicherheitsbedürfnis heraus, den Firm Key selber festlegen möchten, so können Sie das tun.

 Bei einer Änderung des initialen Wertes des Firm Keys müssen Sie aber unbedingt sicherstellen, dass Sie den Firm Key äußerst sicher verwahren. Bei Verlust dieses Schlüssels kann auch Wibu-Systems den Firm Key nicht wiederherstellen.

Der Firm Key wird in der Firm Security Box (FSB) in der Product Item Option (PIO) Secret Data gespeichert.

 Aus Sicherheitsgründen ist ein Abrufen des Firm Keys aus der Firm Security Box (FSB) nicht möglich.

4.1 Product Item Options - Lizenzinträge nach Maß

Jeder Lizenzintrag auf der Ebene der Product Items kann verschiedene miteinander kombinierbare Product Item Options (PIO) besitzen. Diese PIO erlauben Ihnen, für jeden Kunden individuelle Lizenzmodelle festzulegen.

Alle Kunden erhalten von Ihnen die gleiche Software, und über die PIO legen Sie im *CmContainer* fest, welche Funktionen Ihrer Software der Anwender nutzen kann. Die Eigenschaften der einzelnen Optionen finden Sie in der folgenden Tabelle:

Product Item Option	Bemerkung	Lese-Zugriff	Schreib-Zugriff	Eingang in die Verschlüsselung
Text	256 Zeichen Doppel-Byte, Verwendung als Anzeige in <i>CodeMeter WebAdmin</i>	✓	✓	✗
License Quantity	Lizenzanzahl; Anzahl der gleichzeitig benutzbaren Lizenzen, Verwendung für Floating Lizenzen im Netzwerk	✓	mit FSB	✗
Activation Time	Aktivierungsdatum, Verwendung für zeitlich befristete Versionen	✓	mit FSB	✓
Expiration Time	Ablaufdatum, Verwendung für zeitlich befristete Versionen	✓	mit FSB	✓
Usage Period	Nutzungsdauer, Verwendung für zeitlich befristete Lizenzen	✓	einmalig beim ersten Start	✓
Customer Owned License Information	128 Zeichen, Verwendung für kunden-spezifische Daten (z.B. Name des Lizenznehmers)	✓	mit FSB	✗
Feature Map	32-Bit Maske, Verwendung für Freischaltung von Features oder für Versionsverwaltung	✓	mit FSB	✓
Maintenance Period	Wartungszeitraum, Verwendung für zeitlich befristete Software-Wartungsverträge	✓	mit FSB	✓
Linger Time	Nachlaufzeit, Verwendung für die zeitliche Steuerung bei erneutem Starten	✓	mit FSB	✗
Unit Counter	Zähler, Verwendung in pay-per-use-, Reduzieren / er-	✓	Reduzieren / er-	✓

Product Item Option	Bemerkung	Lese-Zugriff	Schreib-Zugriff	Eingang in die Ver-schlüsselung
	pay-per-click-, pay-per-print- oder pay-per-start-Versionen		höhen mit FSB	
User Data	256 Byte Daten, Verwendung zum Speichern von Konfigurationsdaten	✓	✓	X
Protected Data	256 Byte zum Speichern von zusätzlichen Daten	✓	mit FSB	X
Extended Protected Data	(128 +128) x 256 Bytes ¹⁾	✓	mit FSB	X
Hidden Data	(128 +128) x 256 Byte Daten, Verwendung als Schlüsselquelle ^{1) 2)}	mit Passwort	mit FSB	als eigener Schlüssel
Secret Data	(128 +128) x 256 Byte Daten, Verwendung als Schlüsselquelle ¹⁾	X	mit FSB	als eigener Schlüssel

1) (128 +128) x 256 Bytes: 128 (0-127) frei verwendbar, 128 (128-255) reserviert für Wibu-Systems

2) Das [Lesen](#)³²⁴ und [Schreiben](#)³²⁵ von Daten aus bzw. in einen CmContainer ist auch ohne FSB-Zugriff während der Laufzeit über spezielle [WUPI-Funktionen](#)³²² möglich, wenn die CmContainer speziell dafür vorbereitet wurden.

Tabelle 2: Übersicht Product Item Options (PIO)

Zum Anlegen, Ändern und Löschen der Lizenzoptionen benötigen Sie in den meisten Fällen Ihre Firm Security Box. Damit stellt Wibu-Systems sicher, dass Ihre Kunden die von Ihnen verkauften Lizizenzen nicht selbstständig ändern können. Lediglich die Optionen Text und User Data können ohne Firm Security Box geschrieben werden. Gleichzeitig können Sie durch die Verwendung der CodeMeter®-Werkzeuge und Schnittstellen über Abfragen und Prüfungen gewährleisten, dass Ihre Software beim Kunden mit den korrekten Lizenzinformationen verwendet wird.

4.1.1 Product Code

Mit der PIO Product Code identifizieren Sie einen Lizenzintrag eindeutig.

- Der Product Code ist ein 32 Bit-Wert kann vom Lizenzgeber frei gewählt werden.
- Die Festlegung und Programmierung des Product Codes (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Ändern/Löschen	
CodeMeter License Editor	Setzen des Product Codes ³⁵³
CmBoxPgm	Setzen der PIO /p ³⁶¹
CodeMeter License Central	Setzen ³⁹¹ der PIO
Anlegen/Ändern/Löschen	
AxProtector	Product Code muss gesetzt werden.
Softwareschutz-API (WUPI)	WupiCheckLicense ³²³ oder WupiAllocate License ³²² WupiQueryInfoId ³²³

Anlegen/Ändern/Löschen	Abrufen von Informationen aus dem gerade belegten Lizenzeneintrag.
Kern-API	CmAccess ³³⁴ und Bearbeitung des Handles mit CmCrypt ³³⁵

4.1.2 Text

Mit der PIO Text beschreiben Sie einen Lizenzeneintrag.

- Die Text-Option kann bis zu 256 Doppel-Byte umfassen und Informationen, wie z.B. den Namen des Produkts oder des Benutzers beinhalten, wie er in *CodeMeter WebAdmin* angezeigt wird.
- Der Schreib- und Lesevorgang ist nicht begrenzt, d.h. eine Firm Security Box (FSB) wird nicht benötigt.

Die nachfolgende Übersicht zeigt Ihnen, welche *CodeMeter®*-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CodeMeter License Editor	Setzen ³⁵³ der PIO Text
CmBoxPgm	Setzen der PIO /pt ³⁶⁶
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetProductItemText ³⁴⁵
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.3 License Quantity (Lizenzanzahl)

Mit der PIO License Quantity legen Sie die Anzahl der gleichzeitig benutzbaren Lizenzen in einem Netzwerk fest. Diese PIO benötigen Sie zur Umsetzung von Lizenzmodellen wie Concurrent, Floating-Lizenzen oder Terminalserver-Sitzungen.

Gleichzeitig müssen Sie die Lizenzbelegung über verschiedene Zugriffsmodi definieren, d.h. Sie legen fest, wie sich die gestarteten Instanzen der geschützten Software und die belegten Lizenzen im Netzwerk zueinander verhalten.

Diese Festlegung erfolgt nicht im *CmContainer* selbst, sondern bei der Verschlüsselung der Software.

- Die License Quantity-Option kann bis zu 4 Bytes umfassen und enthält die Information über die Anzahl der Lizenzen in einem Netzwerk.

Das Setzen der Eigenschaft auf 0 bestimmt eine ausschließlich lokale Lizenz..

- Die Festlegung und Programmierung der License Quantity (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Tabelle zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CodeMeter License Editor	Setzen ³⁵³ der PIO  License Quantity
CmBoxPgm	Setzen der PIO /plq ³⁶⁴
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetAbsoluteLicenseQuantity ³⁴⁵ oder SetRelativeLicenseQuantity ³⁴⁵
Abfragen/Überprüfen	
AxProtector	Lizenzbelegung und Lizenzoptionen
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.4 Activation Time (Aktivierungsdatum)

Mit der PIO Activation Time setzen Sie Lizenzmodelle um, die mit einem Aktivierungsdatum im Sinne eines "gültig ab..." eine Startzeit für die geschützte Anwendung festlegen.

Legen Sie zusätzlich eine [Ablaufzeit](#)⁴² (Expiration Time) fest, so realisieren Sie Lizenzmodelle, die zeitlich befristet sind, z.B. Leasing-, Vermietungs-, Abonnement-Modelle.

- Eine Activation Time definiert einen sekundengenaugen Wert in Intervallen zwischen dem 1. Januar 2000, 0:00:00 und dem 31. Dezember 2099, 23:59:59. Dieser Wert wird immer im Zeitzonenformat UTC (*Universal Time Coordinated*) gespeichert und ist unabhängig von Zeitzonen oder auch Zeitumstellung.
- Der Zugriff auf die Lizenz erfolgt nur, wenn die Box Time und die Certified Time im *CmContainer* zeitlich nach der festgelegten Activation Time liegen. Dies wird durch einen [ausfall- und manipulationssicheren Prüfungsmechanismus](#)⁴¹⁷ gewährleistet.
- Die Activation Time ist Bestandteil der [Schlüsselableitung](#)⁵⁹. Dieser Schlüssel wird bei jedem Vorgang ermittelt, der eine Verschlüsselung, Entschlüsselung oder Authentifizierung umfasst. Eine nicht erlaubte Manipulation der Activation Time, etwa durch Setzen eines früheren Datums, führt daher zu unterschiedlichen Ableitungsergebnissen und der lizenzierte Zugriff wird unterbunden.
- Ein Anwender kann die Activation Time nicht direkt verändern, d.h. nicht auf ein früheres Datum setzen.
- Die Festlegung und Programmierung der Activation Time (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.
- Die Activation Time kann auf einen festen Wert gesetzt (absolut), oder zu einem eventuell schon vorhandenen Wert dazu addiert werden (relativ).

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CodeMeter License Editor	Setzen ³⁵³ der PIO  Activation Time

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO  pat 
CodeMeter License Central	Setzen  der Anzahl von Tagen, die die Anwendung ab dem ersten Aufruf laufen soll
Programmier-API	Aufruf der Klasse ProductItemParamSet  und nachfolgend SetAbsoluteActivationTime  oder SetRelativeActivationTime 
Abfragen/Überprüfen	
AxProtector	Optionsauswahl in erweiterten Laufzeiteinstellungen
Softwareschutz-API (WUPI)	WupiQueryInfo  Abrufen von Informationen aus dem gerade belegten Lizenzeneintrag
Kern-API	CmAccess  und im Managing API GetBoxContents 

4.1.5 Expiration Time (Verfallsdatum)

Mit der PIO Expiration Time setzen Sie Lizenzmodelle um, die mit einem Ablaufdatum im Sinne eines "gültig bis..." ein Nutzungsende festlegen.

Legen Sie zusätzlich eine [Startzeit](#)  (Activation Time) fest, so realisieren Sie Lizenzmodelle, die zeitlich befristet sind, z.B. Leasing-, Vermietungs-, Abonnement-Modelle.

- Eine Expiration Time definiert einen sekundengenauen Wert in Intervallen zwischen dem 1. Januar 2000, 0:00:00 und dem 31. Dezember 2099, 23:59:59. Dieser Wert wird immer im Zeitzonenumformat UTC (*Universal Time Coordinated*) gespeichert und ist unabhängig von Zeitzonen oder Zeitumstellung.
- Der Zugriff auf die Lizenz erfolgt nur, wenn die Box Time und die Certified Time im *CmContainer* zeitlich vor der festgelegten Expiration Time liegen. Dies wird durch einen [ausfall- und manipulationssicheren Prüfungsmechanismus](#)  gewährleistet.
- Die Expiration Time ist Bestandteil der [Schlüsselableitung](#)  . Dieser Schlüssel wird bei jedem Vorgang ermittelt, der eine Verschlüsselung, Entschlüsselung oder Authentifizierung umfasst. Eine nicht erlaubte Manipulation der Expiration Time, etwa durch eine Verlängerung, führt daher zu unterschiedlichen Ableitungsergebnissen und der lizenzierte Zugriff wird unterbunden.
- Ein Anwender kann die Expiration Time nicht direkt verändern, besonders nicht auf ein späteres Datum verlängern.
- Die Festlegung und Programmierung der Expiration Time (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.
- Die Expiration Time kann auf einen festen Wert gesetzt (absolut), oder zu einem eventuell schon vorhandenen Wert dazu addiert werden (relativ).

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CodeMeter License Editor	Setzen  der PIO  Expiration Time
CmBoxPgm	Setzen der PIO  
CodeMeter License Central	Setzen  der PIO

Anlegen/Verändern/Löschen

Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetAbsoluteExpirationTime ³⁴⁵ oder SetRelativeExpirationTime ³⁴⁵
Abfragen/Überprüfen	
AxProtector	Optionsauswahl in erweiterten Laufzeiteinstellungen.
Softwareschutz-API (WUPI)	WupiQueryInfo ³²³ Abrufen von Informationen aus dem gerade belegten Lizenzeintrag
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.6 Usage Period (Nutzungsdauer)

Mit der PIO Usage Period definieren Sie einen festen Nutzungszeitraum für die Nutzung der geschützten Software an. Damit setzen Sie Lizenzmodelle um, deren Laufzeit nicht an ein festes Startdatum gebunden ist. Der Nutzungszeitraum beginnt dann mit dem ersten Start der geschützten Anwendung. Hiermit realisieren Sie z.B. 'echte' Demoversionen.

- Die Festlegung und Programmierung der Usage Period (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen

CodeMeter License Editor	Setzen ³⁶³ der PIO  Usage Period
CmBoxPgm	Setzen der PIO /pup ³⁶⁷
CodeMeter License Central	Setzen ³⁹¹ der Anzahl von Tagen, die die Anwendung ab dem ersten Aufruf laufen soll
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetUsagePeriod ³⁴⁵

Abfragen/Überprüfen

AxProtector	---
	(aber Übernahme der Einstellungen aus dem Verfallsdatum)
Softwareschutz-API (WUPI)	WupiQueryInfo ³²³ Abrufen von Informationen aus dem gerade belegten Lizenzintrag

4.1.7 Customer Owned License Information (COLI)

Mit der PIO Customer Owned License Information (COLI) zeigen Sie zusätzliche personalisierte Lizenzinformationen an, z.B. Name des Lizenznehmers oder Seriennummer in *CodeMeter WebAdmin*.

- Die Customer Owned License Information-Option kann bis zu 256 Bytes umfassen.
- Die Festlegung und Programmierung der PIO Customer Owned License Information (COLI) (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

führen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /pcoli ³⁶²
Programmier-API	---
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.8 Unit Counter (Begrenzungszähler)

Mit der PIO Unit Counter als manipulationssicheren Zähler setzen Sie Lizenzmodelle um, die eine geschützte Software nach ihrer tatsächlichen Nutzung abrechnet, z. B pay-per-use, pay-per-click, pay-per-print, etc.

Sie definieren einen Ausgangswert und welche Software-Aktion diesen Zähler um wieviele Einheiten herunterzählt. Aktionen sind beispielsweise Anzahl der Aufrufe bestimmter Software-Funktionen, Anzahl der Ausdrucke, etc..

Gleichzeitig kann ein Unit Counter auch zur zeitlichen Begrenzung einer Lizenz verwendet werden, in dem man die Software in festen Intervallen zur Laufzeit auf eine gültige Lizenz prüft, und bei jeder Prüfung den Zähler um einen definierten Wert herunterzählt.

- Ein Unit Counter kann ganzzahlige Werte zwischen 0 und 4294967294 (Hex: FFFFFFFE) (32 Bits) annehmen. Bis Firmware Version 1.18 sind lediglich Werte zwischen 0 and 16777215 (24 Bit) möglich.
- Die Anzahl der Einheiten, um die der Unit Counter heruntergezählt wird (Dekrement, Delta-Wert), können Sie als Wert zwischen 1 und 99999 setzen. Das Herunterzählen erfolgt manipulationssicher im *CmContainer*.



Aus Sicherheitsgründen verringert ein Anwender den Wert eines Unit Counter durch Aufrufe der Anwendung, eine Erhöhung des Wertes ist nicht möglich.

- Der Unit Counter ist Bestandteil der [Schlüsselableitung](#)⁵⁹. Dieser Schlüssel wird bei jedem Vorgang ermittelt, der eine Verschlüsselung, Entschlüsselung oder Authentifizierung umfasst. Eine nicht erlaubte Manipulation des Unit Counter, etwa durch das Heraufsetzen des Unit Counter oder die Verringerung des Delta-Wertes, führt daher zu unterschiedlichen Ableitungsergebnissen und der lizenzierte Zugriff wird unterbunden.
- Wenn der Unit Counter einen Wert von 0 erreicht, wird der Zugriff auf die Lizenz nicht mehr ausgeführt. Nur spezielle Operationen, die den Unit Counter ignorieren, können noch ausgeführt werden.
- Der Unit Counter kann auf einen festen Wert gesetzt (absolut), oder zu einem eventuell schon vorhandenen Wert dazu addiert werden (relativ).

Die nachfolgende Übersicht zeigt Ihnen, welche *CodeMeter®*-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CodeMeter License Editor	Setzen ³⁵³ der PIO  Unit Counter
CmBoxPgm	Setzen der PIO  puc ³⁶⁶
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse <u>ProductItemParamSet</u> ³⁴⁵ und nachfolgend <u>SetAbsoluteUnitCounter</u> ³⁴⁵ oder <u>SetRelativeUnitCounter</u> ³⁴⁵
Abfragen/Überprüfen	
AxProtector	Wenn ein Unit Counter im <i>CmContainer</i> vorhanden ist, dann wird dieser beim Start der Software durch <i>AxProtector</i> automatisch um 1 heruntergezählt. Sie können das Dekrement aber auch abändern. Zusätzlich können Sie die Lizenz auch zur Laufzeit prüfen und dafür einen vorhandenen Unit Counter verwenden.
Softwareschutz-API (WUPI)	<u>WupiDecreaseUnitCounter</u> ³²³ Herunterzählen eines Unit Counter einer Lizenz, die im <i>AxProtector</i> mit der <code>Id = LicenseId</code> definiert ist. Dies können Sie verwenden, wenn Sie eine Pay-per-click Funktionalität in Ihre Software integrieren möchten. Besitzt die Lizenz keinen Unit Counter, dann liefert die Funktion keinen Fehler zurück, d.h. Sie können durch die Programmierung der Lizenz festlegen, ob der Lizenznehmer eine Pay-per-click, oder eine unlimitierte Lizenz erhält. <u>WupiQueryInfoId</u> ³²³ Abrufen von Informationen aus dem gerade belegten Lizenzintrag
Kern-API	<u>CmCrypt</u> ³³⁵

4.1.9 Feature Map

Mit der PIO Feature Map setzen Sie Lizenzmodelle um, die bestimmte Funktionen (Module, Features) oder Versionen einer Anwendung zur lizenzierten Nutzung freischaltet.

Sollen für unterschiedliche Module eines Programmes keine individuellen Product Items verwendet werden, so kann jedem Modul innerhalb eines Product Items ein Feature Code zugewiesen werden. Die Feature Map hat 32 Bit, so dass bis zu 32 Feature Codes (Bit-Werte) individuell zugewiesen und aktiviert werden können.

Vorteilhaft ist die Verwendung der Feature Map zur Versionskontrolle. Dazu wird jeder Programmversion ein eigener Feature Code zugewiesen.

Versionsverwaltung über Feature Code

Jede neue Hauptversion wird dabei als ein Bit kodiert. Soll Ihr Kunde mehrere Versionen nutzen können, dann aktivieren Sie die entsprechenden Bits über das Setzen des Bits auf einen Wert von 1.

In der Kombination mit der Lizenzanzahl können Sie so ein Downgrade-Recht im Netzwerk realisieren. Ihr Kunde kann dann bis zur vorgegebenen Anzahl an Lizensen entweder die aktuelle, oder die freigeschalteten Vorgängerversionen einsetzen. Aber in der Summe nie mehr als die von Ihnen programmierte Lizenzanzahl.

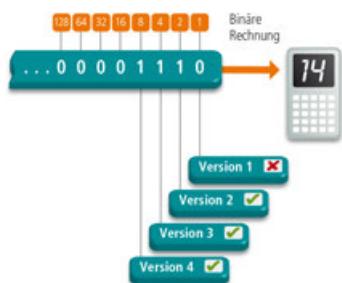


Abbildung 8: Versionsmanagement über Feature Code

Wie aus der Abbildung ersichtlich, bedeutet dann der Binärwert von "1110" bzw. der Dezimalwert von 14, dass Sie die Versionen 2 bis 4 freischalten, aber nicht die Version 1. In *AxProtector* und *IxProtector* ist an den betreffenden Stellen die Eingabe des Feature Codes möglich.



Selbst, wenn Sie die Feature Map-Option nicht zur Umsetzung von Lizenzmodellen benutzen, fließt der Wert der Feature Map, der Feature Code durch die [Schlüsselableitung](#)⁵⁹ in Ver- und Entschlüsselungsvorgänge ein. Die Feature Map hat dann einen Feature Code von 0 und die Product Item Option ist nicht aktiv..

Eine Feature Map hat folgende Eigenschaften:

- Bis zu 32 Features können unabhängig verwaltet werden. Jedes Feature wird durch ein einzelnes Bit in dieser Map dargestellt.
- Der Feature Code ist Bestandteil der [Schlüsselableitung](#)⁵⁹. Dieser Schlüssel wird bei jedem Vorgang ermittelt, der eine Verschlüsselung, Entschlüsselung oder Authentifizierung umfasst. Eine nicht erlaubte Manipulation des Feature Codes, etwa durch das Setzen eines entsprechenden Bits in der Feature Map, führt daher zu unterschiedlichen Ableitungsergebnissen und der lizenzierte Zugriff wird unterbunden.
- Ein Anwender kann den Feature Code nicht direkt verändern. Insbesondere kann er auch keine neuen Features hinzufügen, um diese dann zu aktivieren.
- Die Festlegung und Programmierung der Feature Map (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche *CodeMeter®*-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen

<i>CodeMeter License Editor</i>	Setzen ³⁵³ der PIO Feature Map
<i>CmBoxPgm</i>	Setzen der PIO /pfm ³⁶³
<i>CodeMeter License Central</i>	Setzen ³⁹¹ der PIO
<i>Programmier-API</i>	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetFeatureMap ³⁴⁵
Abfragen/Überprüfen	
<i>AxProtector</i>	Muss ausgefüllt sein

Abfragen/Überprüfen

Softwareschutz-API (WUPI)	WupiQueryInfo ³²³
Kern-API	Abrufen von Informationen aus dem gerade belegten Lizenzintrag CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.10 Maintenance Period (Wartungszeitraum)

Mit der PIO Wartungszeitraum (Maintenance Period) legen Sie eine absolute Zeitspanne im *CmContainer* ab, z.B. 1.4.2011 bis 31.03.2012. Diese Lizenz ist damit eingeschränkt auf Software-Versionen, die innerhalb des Wartungszeitraums (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Datum sich innerhalb des festgelegten Zeitraumes befindet. Es kann dabei gewählt werden, ob der Wartungszeitraum (Maintenance Period) vorhanden sein muss, oder ob gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist (Standard-Einstellung). Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.



Damit setzen Sie Lizenzmodelle um, die die Gewährung von Support- und Wartungsleistungen bei der Nutzung der Software abbilden.

- Die Wartungszeitraum (Maintenance Period)-Option hält zwei Werte zu je 32-Bit: Beginn und Ende des Wartungszeitraums (Maintenance Period). In beiden Werten können entweder Datumswerte eingegeben werden oder ganzzahlige Zeitstempel in der für CodeMeter® üblichen Formatierung (Sekunden seit 1.1.2000). Dies deckt die derzeit bei CodeMeter® üblichen Zeithorizonte bis maximal Februar 2136 ab.
- Die Festlegung und Programmierung der PIO Wartungszeitraum (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen

CodeMeter License Editor	Setzen ³⁵⁴ der PIO Wartungszeitraum
CmBoxPgm	Setzen der PIO /pmd ³⁶⁴
CodeMeter License Central	Noch nicht implementiert.
Programmier-API	Aufruf der Klasse MaintenancePeriodParamSet ³⁴⁵ und nachfolgend MaintenancePeriodPIO ³⁴⁴

Abfragen/Überprüfen

AxProtector	Kann aktiviert und überprüft werden
Softwareschutz-API (WUPI)	WupiQueryInfo ³²³ Abrufen von Informationen aus dem gerade belegten Lizenzintrag
Kern-API	CmCrypt2 ³³⁵ , CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.11 Linger Time

Mit der PIO Linger Time definieren Sie eine Zeitspanne in Sekunden wie lange eine Lizenz noch belegt bleibt solange nachdem eine geschützte Anwendung freigegeben oder beendet wurde (Nachlaufzeit).

Damit setzen Sie Lizenzmodelle um, bei denen das Verhalten beim Wiederstarten von geschützten Anwendungen zeitlich gesteuert werden soll.

- Die Linger Time-Option wird in Sekunden angegeben.
- Die Festlegung und Programmierung der PIO Linger Time (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Wie sich die Nachlaufzeit verhält, ist abhängig vom ausgewählten Zugriffsmodus, der in den Laufzeiteinstellungen gesetzt ist.

Zugriffsmodus	Linger Time-Verhalten
Normal user limit	Jede Lizenz läuft nach, da in diesem Modus jede gestartete Instanz eine Lizenz belegt. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Da in diesem Modus mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz belegen, läuft pro PC die Lizenz nach, die als letzte von einer der Instanzen der geschützten Anwendung freigegeben wurde.
Exclusive Mode	Eine Lizenz läuft nach, da in diesem Modus die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden kann. In einer Server-Client-Umgebung kann der Client dann in der definierten Zeitspanne keine Lizenz an diesem Server verwenden.
No user limit	Eine Lizenz läuft <u>nicht</u> nach, da in diesem Modus beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden können ohne zusätzliche Lizensen zu belegen.

Die nachfolgende Übersicht zeigt Ihnen, welche *CodeMeter®*-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
<i>CmBoxPgm</i>	Setzen der PIO /plt ³⁶⁴
<i>CodeMeter License Central</i>	Noch nicht implementiert
<i>Programmier-API</i>	Aufruf der Klasse LingerTimeParamSet ³⁴⁵ und nachfolgend LingerTimePIO ³⁴⁴
Abfragen/Überprüfen	
<i>AxProtector</i>	Kann deaktiviert werden
<i>Kern-API</i>	CmAccess2 ³³⁴

4.1.12 User Data

Mit der PIO User Data speichern Sie sichtbare Daten. Hier können Sie beispielsweise Konfigurationsdaten hinterlegen.

- Die User Data Option kann bis zu 256 Bytes umfassen.
- Der Schreib- und Lesevorgang ist nicht begrenzt, d.h. eine Firm Security Box (FSB) wird nicht benötigt. Zu Laufzeit der Anwendung ist diese PIO von jedem veränderbar.

Die nachfolgende Übersicht zeigt Ihnen, welche *CodeMeter®*-Werkzeuge und Schnittstellen Sie benutzen

können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /ppd ³⁶⁷ 
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetUserData ³⁴⁵
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.13 Protected Data

Mit der PIO Protected Data speichern Sie zusätzliche sichtbare Daten im Binärformat. Zum Beispiel spezifische Informationen zum Kunden.

- Die PIO Protected Data kann bis zu 256 Doppel-Byte Daten speichern.
- Die Festlegung und Programmierung der PIO Protected Data (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /ppd ³⁶⁵ 
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetProtectedData ³⁴⁵
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.1.14 Extended Protected Data

Mit der PIO Extended Protected Data speichern Sie zusätzliche sichere aber sichtbare Daten im Binärformat.

- Die PIO Extended Protected Data umfasst (128 + 128) Typen, die je eine Länge von bis zu 256 Bytes umfassen können.



Von den Typen sind 128 (0-127) für Kunden und 128 (128-256) für Wibu-Systems reserviert.

- Die Festlegung und Programmierung der PIO Extended Protected Data (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist hingegen nicht begrenzt.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen

können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /ped  
CodeMeter License Central	Setzen  der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet  und nachfolgend SetExtendedProtectedData 
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess  und im Managing API GetBoxContents 

4.1.15 Hidden Data

Mit der PIO Hidden Data speichern Sie zusätzliche sichere aber nur mit einem Passwort lesbare Daten im Binärformat. Die PIO kann zum Beispiel eigene Schlüsselkonstanten für die Entschlüsselung enthalten.

- Die PIO Hidden Data umfasst (128 + 128) Typen, die je eine Länge von bis zu 256 Bytes umfassen können.



Von den Typen sind 128 (0-127) für Kunden und 128 (128-256) für Wibu-Systems reserviert.

- Die Festlegung und Programmierung der PIO Hidden Data (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Der Lesezugriff ist ausschließlich mit einem Passwort möglich.
- Das [Lesen](#)  und [Schreiben](#)  von Daten aus bzw. in einen *CmContainer* ist aber auch ohne FSB-Zugriff während der Laufzeit über spezielle [WUPI-Funktionen](#)  möglich, wenn die *CmContainer* dafür vorbereitet wurden.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /phd  
CodeMeter License Central	Setzen  der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet  und nachfolgend SetHiddenData 
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	WupiReadData  oder WupiReadDataInteger  , WupiWriteData  oder WupiWriteDataInteger 
Kern-API	CmAccess  und im Managing API GetBoxContents 

4.1.16 Secret Data

Mit der PIO Secret Data speichern Sie zusätzliche sichere aber unsichtbare Daten im Binärformat. Die PIO kann zum Beispiel eigene Schlüssel für die Entschlüsselung enthalten.

- Die PIO Secret Data umfasst (128 + 128) Typen, die je eine Länge von bis zu 256 Bytes umfassen können.



Von den Typen sind 128 (0-127) für Kunden und 128 (128-256) für Wibu-Systems reserviert.

- Die Festlegung und Programmierung der PIO Secret Data (Schreibvorgang) ist nur mit einer Firm Security Box (FSB) möglich. Ein Lesezugriff ist nicht möglich.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie benutzen können, um diese PIO anzulegen, zu ändern oder zu löschen, oder Abfragen/Überprüfungen durchzuführen.

Anlegen/Verändern/Löschen	
CmBoxPgm	Setzen der PIO /psd ³⁶⁶
CodeMeter License Central	Setzen ³⁹¹ der PIO
Programmier-API	Aufruf der Klasse ProductItemParamSet ³⁴⁵ und nachfolgend SetSecretData ³⁴⁵
Abfragen/Überprüfen	
AxProtector	---
Softwareschutz-API (WUPI)	---
Kern-API	CmAccess ³³⁴ und im Managing API GetBoxContents ³³⁶

4.2 Sicherheit großgeschrieben

Zusätzliche Vorteile des hardware-basierten Schutzes mit CodeMeter® entnehmen Sie der folgenden Tabelle.

Vorteil	Beschreibung
Firmware läuft geschützt in der Hardware	Die Firmware, d.h. die Speicherung und Berechnung der Schlüssel und die entsprechende Entschlüsselung bzw. Verschlüsselung läuft sicher geschützt im Smartcard-Chip im CmDongle ab. Dieser Teil kann vom Hacker nicht analysiert werden und stellt somit eine Black Box dar.
Hardware kann gesperrt werden	Wenn Sie innerhalb Ihrer Software einen Angriff erkennen (dies machen unsere Tools automatisch für Sie), dann haben Sie die Möglichkeit, ein Sperrkommando aus Ihrer Software heraus an den CmDongle zu schicken. Dieses Kommando sperrt alle Ihre Lizenzen auf Ihrer Firm Item-Ebene. Sie können diese Lizenzen per Fernprogrammierung wieder freischalten, aber bis zur Freischaltung verhält sich der CmDongle so, als wären diese Lizenzen (und damit die Schlüssel) nicht vorhanden. Der Hacker hat keinen zweiten Versuch.
Zähler können nicht durch ein Backup zurückgesetzt werden	Zähler werden sicher im Smartcard-Chip im CmDongle gespeichert. Damit können diese nicht von außen manipuliert oder durch das Ein-

Vorteil	Beschreibung
Gelöschte Lizenzen können nicht durch ein Backup zurückgesetzt werden	Lizenzen, die in einem <i>CmDongle</i> gelöscht wurden, sind nicht mehr vorhanden. Durch die Übermittlung einer fälschungssicheren Quittung ist der Lizenzgeber sicher, dass die Lizenz im aktuellen <i>CmDongle</i> nicht mehr vorhanden ist und auch nicht wieder hergestellt werden kann.
Expiration Time und Usage Period werden gegen die interne Uhr überprüft	Alle verwendeten Zeiten wie Expiration Time und Usage Period werden gegen die intern im Smartcard-Chip laufende Uhr geprüft. Die einge tragenen Zeiten können nicht manipuliert werden; die interne Uhr kann nicht zurückgesetzt werden. Damit ist eine abgelaufene Lizenz nicht wieder herstellbar.
Certified Time über Zeitserver und Sperre	Zusätzlich stellt Wibu-Systems weltweit verteilte Zeitserver bereit, die eine Certified Time zur Verfügung stellen und zeitliche Manipulationsversuche erschweren. Verloren gegangene <i>CmDongles</i> können von Wibu-Systems gesperrt werden. Die Sperre ist dann auf den <i>CodeMeter®</i> -Zeitservern hinterlegt und sobald der betreffende <i>CmDongle</i> seine Zeit aktualisieren will, wird dieser gesperrt.
Einfacher Wechsel auf einen anderen PC	Ein Umzug der Software auf einen anderen PC ist problemlos möglich. Software installieren und den <i>CmDongle</i> anstecken. Dies kann der Lizenznehmer selbst beliebig häufig durchführen, ohne dass er dazu den Lizenzgeber benötigt. Der Lizenzgeber kann sich auf der anderen Seite sicher sein, dass nach einem Wechsel auf einen anderen PC die Software nicht gleichzeitig noch auf dem alten PC ausgeführt werden kann.
Sicherheit vor Verlust der Lizenz durch Viren und andere Schad-Software	Die Programmierung (Anlegen, Ändern, Löschen) einer Lizenz in einem <i>CmDongle</i> ist durch Kryptographie abgesichert. Nur Sie mit Ihrer FSB können Einträge löschen. Damit ist kein Virus in der Lage die Lizenzen beim Lizenznehmer zu zerstören.

Tabelle 3: Vorteile der Hardware

4.3 Lizenzmodelle - Abbildungsvielfalt mit CodeMeter

Wie oben dargestellt, kann jeder Lizenzeintrag über beliebig miteinander kombinierbare Product Item Options verfügen. Damit erhalten Sie als Lizenzgeber die Möglichkeit, Ihre Lizenzstrategien über Lizenzmodelle abzubilden. Die folgende Tabelle zeigt welche grundlegenden Lizenzmodelle Sie innerhalb Ihrer Lizenzstrategie abbilden können.

Lizenzmodell	Beschreibung
Einzelplatz	Die Lizenz befindet sich auf einem lokalen Rechner (<i>CmActLicense</i>) oder in einem lokal angeschlossenen <i>CmDongle</i> . Die Software läuft auf dem gleichen Rechner, der gleichen Maschine oder in der Cloud.
Netzwerk	Die Lizenz befindet sich auf einem zentralen Server im Netzwerk. Die PC-Software nutzt diese als Floating Lizenz. Im Embedded-Bereich wird dies vor allem als Notfalllizenz verwendet. In der Cloud spielen Netzwerklicenzen eine untergeordnete Rolle.
Feature-on-Demand	Durch individuelle Lizenzen werden einzelne Produkte und Module freigeschaltet.

Lizenzmodell	Beschreibung
	Damit können Sie zusätzlichen Umsatz durch den Verkauf von Add-ons generieren. Vor allem im Embedded-Bereich werden damit weitere Funktionen für Service-Techniker nach Zustecken eines entsprechenden <i>CmDongles</i> nutzbar.
Unbefristete Lizenz (Perpetual License)	Die Lizenz wird dauerhaft erteilt und läuft nicht ab.
Demoversion	Der Anwender kann die Software in einem von Ihnen definierten Funktionsumfang und in dem von Ihnen definierten Zeitraum nutzen.
Miete, Leasing, Abo-Modelle (Subscription)	Die Lizenz ist für eine von Ihnen festgelegte Zeit gültig. Über die <i>CodeMeter License Central</i> können Sie die Lizenz automatisch verlängern.
Pay-per-Use	Die Lizenzen erlauben die Abrechnung in Abhängigkeit von benutzten Einheiten. Sie legen dabei selbst fest, ob diese Einheiten zeit- oder funktionsbasiert sind. In der Cloud wird diese Lizenz vor allem für eine datenabhängige Abrechnung verwendet.
Wartungsvertrag (Software Assurance)	Die Lizenz ist eine dauerhafte Lizenz, die zusätzlich die Information über einen Wartungsvertrag enthält. Updates, die während dieses Intervalls veröffentlicht werden, können vom Anwender ohne weitere Freischaltung sofort genutzt werden.
Downgrade-Recht	Die Lizenz umfasst das Recht, wahlweise eine ältere Version einzusetzen. Damit kann ein Großkunde weiterhin eine einheitliche Version einsetzen und zu einem von ihm gewählten Zeitpunkt auf die neue Version migrieren.
Grace Period-Lizenz	Die Lizenz umfasst das Recht, wahlweise die nächste Version einzusetzen. Damit können Sie die aktuelle Version auch noch nach der Ankündigung einer neuen Version abverkaufen.
Volumenlizenzen (mit Kontrolle)	Der Großkunde erhält die Möglichkeit, eine von Ihnen bestimmte Anzahl an Lizenzen zu aktivieren.
Volumenlizenzen (ohne Kontrolle)	Der Großkunde erhält einen Aktivierungscode, den er beliebig oft einsetzen kann. Die Anzahl der Lizenzen wird rein vertraglich geregelt (nur bei <i>CmActLicense</i>).
Lizenzierung von Versionen	Die Lizenz umfasst wahlweise eine oder mehrere Versionen der gleichen Software.
Cold Standby	Der Anwender besitzt eine Ersatzlizenz, die er im Falle eines Ausfalls verwenden kann. Der Anwender muss dazu die Lizenz aktiv hochfahren.
Hot Standby	Der Anwender besitzt eine Ersatzlizenz, die produktiv verfügbar ist und erst im Falle eines Ausfalls automatisch verwendet wird.
Hochverfügbarkeitslizenz	Der Anwender besitzt einen redundanten Lizenzserver ("2 aus 3"-Verfahren).
Overflow-Lizenzen	Der Anwender hat die Möglichkeit, in einem von Ihnen definierten Rahmen mehr Lizenzen zu verwenden als er besitzt. Diese Benutzung wird protokolliert und kann von Ihnen nachträglich berechnet werden.
Ausleihbare Lizenzen	Der Anwender hat die Möglichkeiten, Lizenzen zeitlich befristet von einem Lizenzserver auf einen lokalen Rechner (<i>CmActLicense</i>) oder in einen <i>CmDongle</i> auszuleihen. Nach Ende der Ausleihdauer ist die Lizenz automatisch wieder auf dem Lizenzserver verfügbar und lokal gesperrt. Lizenzen können manuell vor Ablauf zurückgegeben werden.
Nutzergebundene Lizenzen	Die Lizenz ist an einen Benutzernamen gebunden.
Rechnergebundene Lizenzen	Die Lizenz ist an einen Rechnernamen gebunden.
Zeitzonenbasierte Lizenzen	Die Lizenz kann nur in einer von Ihnen definierten geographischen Region (Zeitzone) verwendet werden.

Tabelle 4: Abbildbare Lizenzmodelle mit *CodeMeter*®

4.3.1 Die Umsetzung von Lizenzmodellen mit CodeMeter

Dieser Abschnitt zeigt Ihnen eine Reihe von Beispielen, wie Sie *CodeMeter®* nutzen, um verschiedene Lizenzmodelle umzusetzen. Die notwendigen [Programmierung](#)³⁴⁴ nehmen Sie mit den *CodeMeter®*-Anwendungen vor.

Alles was Sie zum Nachbauen der Beispiele benötigen, ist Ihr gültiger Firm Code. Mit diesem erstellen Sie Ihren "Lizenzverzeichnis"-Eintrag, das Firm Item. Danach legen Sie ihre eigentlichen Lizenzeinträge in diesen "Lizenzverzeichnis"-Eintrag ab, die Product Items. Product Items sind durch frei wählbare Product Codes eindeutig identifiziert. Jede Lizenz konfigurieren Sie dann mit Hilfe verschiedener Product Item Optionen nach Ihren Wünschen und Anforderungen.

 Natürlich können Sie diese Beispiele auch so abändern und miteinander kombinieren, dass sie genau Ihren Lizenzstrategien entsprechen.

4.3.1.1 Lokale Einzelplatzlizenz

Die Lizenz ist ausschließlich lokal auf dem Computer verfügbar, an dem ein *CmContainer* verfügbar ist.

 Definieren Sie auf der Product Item-Ebene einen von Ihnen frei gewählten Product Code. Setzen Sie die Product Item Option License Quantity auf einen Wert von 0.

 Jede Lizenz ist automatisch auch eine Floating Lizenz im Netzwerk. Durch das Setzen der License Quantity auf einen Wert von 0 erhalten Sie eine ausschließlich lokale Lizenz.

4.3.1.2 Concurrent-/ Floating-Lizenz im Netzwerk

Die Lizenz wird hier über einen zentralen Server zur Verfügung gestellt und erlaubt der vorgegebenen Anzahl von Clients die gemeinsame Nutzung der Lizenz.

 Definieren Sie auf der Product Item-Ebene die Anzahl der gleichzeitig im Netzwerk nutzbaren Lizzen über die Product Item Option License Quantity.

Aktivieren Sie dann auf dem gewünschten Computer, mit dem der *CmContainer* verbunden ist, *CodeMeter Lizenzserver*. *CodeMeter Lizenzserver* ist in der *CodeMeter®* Laufzeitumgebung bereits enthalten. Sie aktivieren den Server in [CodeMeter WebAdmin](#)³⁴⁴. In *CodeMeter WebAdmin* können Sie auch überwachen, wie viele Lizizen von welchem Computer aus belegt sind. Beim Zugriff auf die Lizenz können Sie zwischen verschiedenen Modi wählen.

- **UserLimit:** Jede gestartete Instanz Ihrer Software belegt genau eine Lizenz.
- **StationShare:** Pro Computer kann die Anwendung beliebig häufig gestartet werden, es zählt nur eine Lizenz pro PC.
- **NoUserLimit:** Die Software kann gestartet werden ohne eine Lizenz zu belegen, auch dann, wenn bereits alle Lizizen belegt sind.

 Bei Betrieb des *CmContainers* innerhalb einer virtuellen Maschine (VM) muss die Lizenz direkt in der Sitzung (session) verfügbar sein. Ein Sharing zwischen verschiedenen Sitzungen ist nicht möglich.

Im Betrieb auf einem Terminalserver oder auch im Multiuser-Modus unter Windows XP oder Windows Vista verhindern Sie Lizenzverletzungen durch das Setzen des *CodeMeter Lizenzservers* auf die Minimalversion 3.20.

Damit erledigt CodeMeter® das Handling automatisch für Sie und jede Session wird wie ein separater PC ausgewertet, inklusive allen Modi.

4.3.1.3 Demo-Versionen

Zeitliche Begrenzung

Zum einen können Sie die Lizenz für eine Demoversion zeitlich begrenzen. Sie können ein festes Ablaufdatum (Expiration Time) oder einen Nutzungszeitraum (Usage Period) vorgeben.

Bei der Verwendung eines festen Nutzungszeitraums entscheidet der Zeitpunkt der ersten Benutzung der geschützten Anwendung, wann der Testzeitraum endet. Hiermit realisieren Sie echte Demoversionen, deren Laufzeit nicht an einen vorher festgelegten Termin gebunden ist.

 Definieren Sie auf der Product Item Ebene ein Ablaufdatum oder einen Nutzungszeitraum über die Product Item Option Expiration Time bzw. Usage Period.

 Expiration Time bzw. Usage Period werden bei CodeMeter® gegen die interne Uhr im CmContainer verglichen und sind damit gegen Manipulationen sicher (zur Synchronisierung der Zeiten siehe [CodeMeter Zeitserver](#)⁴¹⁷).

Laufzeit-Begrenzung x-mal Starten

Das Begrenzen einer Lizenz im Rahmen einer Demo-Version können Sie aber auch über die Angabe umsetzen, wie oft eine Anwendung gestartet werden darf.

 Definieren Sie auf der Product Item Ebene einen Unit Counter, der bei jedem Start der Software zur Laufzeit der Anwendung heruntergezählt wird (Wert = Laufzeit / Zeiteinheit). In der Software zählen Sie dann den Unit Counter pro Zeiteinheit um den Wert von 1 herunter.

Funktionale Einschränkung

Demo-Versionen können sich auch durch den Funktionsumfang im Vergleich zur Vollversion unterscheiden. Für diesen Fall können Sie eine funktional eingeschränkten Demo-Version lizenziieren (siehe [Modulare Lizenzen](#)⁴⁵⁵).

4.3.1.4 Modulare Lizenzen

Modulare Lizenzen erlauben Ihnen die unterschiedliche Lizenzierung von gesonderten Teilen einer geschützten Anwendung (Module/Funktionalitäten). Sie haben zwei Möglichkeiten modulare Lizenzierung umzusetzen: über die Verwendung unterschiedlicher Product Codes, oder den Gebrauch der Product Item Option Feature Map.

Unterschiedliche Product Codes

 Definieren Sie einen Product Code für jedes Modul (Funktionalität) der Anwendung. Auf diese Weise können Sie 6.000 verschiedene Module aktivieren und bei jedem weitere Lizenzoptionen, wie Expiration Time oder License Quantity (Netzwerklicenzen) separat vorgeben.

Feature Map

 Definieren Sie einen Feature Code. Jedes Bit in der Feature Map steht dann genau für ein Modul (Funktionali-

tät). Durch das Programmieren der entsprechenden Feature Map können Sie die einzelnen Module (Funktionalitäten) freischalten.

 Die Lizenzen für einzelne Module können auch auf verschiedene *CmContainer* verteilt sein. So könnte z.B. die Basisversion mit Rechnerbindung laufen, während der Service-Techniker mit seinem *CmContainer* Zugriff auf erweiterte Funktionen erhält.

4.3.1.5 Miete, Leasing

Lizenzen im Bereich der Miete oder des Leasing von Software erlauben das Festlegen einer zeitlichen Frist, innerhalb der die Nutzung der Anwendung gestattet ist.

 Definieren Sie auf der Product Item Ebene ein Ablaufdatum oder einen Nutzungszeitraum über die Product Item Option Expiration Time bzw. Usage Period.

 Expiration Time bzw. Usage Period werden bei *CodeMeter®* gegen die interne Uhr im *CmContainer* verglichen und sind damit gegen Manipulationen sicher (zum Prüfungsmechanismus der Zeiten siehe [CodeMeter Zeitserver](#)⁴¹⁷).

4.3.1.6 Tatsächlich ausgeführte Aktionen

Lizenzen im Bereich pay-per-use basieren auf der Abrechnung tatsächlicher getätigter Aufrufe einer Software, oder deren Module. Beispielsweise pay-per-click, pay-per print, pay-per-start, etc.. Dies gewährleistet maximale Flexibilität, z.B. zusätzliche Kunden zu gewinnen, die die Software nach Nutzung bezahlen möchten.

 Definieren Sie auf der Product Item Ebene einen Unit Counter. Vor oder nach der entsprechenden Aktion in Ihrer Anwendung setzen Sie den Zähler um eine oder mehrere Einheiten herab. Wenn der Zähler bei einem Wert von Null (0) angelangt ist, wird die Lizenz ungültig und kann nicht mehr länger eingesetzt werden. Sie können den Gebrauch natürlich auch auf eine bestimmte Anzahl von Aufrufen begrenzen.

 Bei verschiedenen Aktionen können Sie wahlweise den gleichen Unit Counter oder verschiedene Unit Counter reduzieren.

4.3.1.7 Downgrade/Versionsmanagement

Downgrade

Beim Downgrade wird die Lizenz erteilt, anstatt der aktuell lizenzierten Version eine frühere Version des selben Produkts einzusetzen.

 Definieren Sie auf der Product Item-Ebene die Product Item Option Feature Map. Jedes Bit in der Feature Map steht dann für eine Version. So können Sie z.B. eine Floating Lizenz auf 3 PCs gleichzeitig inklusive einem Downgrade-Recht anbieten, d.h. der Lizenznehmer kann auf 3 PCs entweder die alte, oder die neue Version starten, aber beide zusammen auf maximal 3 PCs gleichzeitig.



In der Summe können aber nur so viele Anwendungen gestartet werden, wie in der Product Item Option License Quantity definiert wurden.

Versionskontrolle



Definieren Sie auf der Product Item-Ebene die Product Item Option Feature Map. Jedes Bit in der Feature Map steht für eine Version, die Sie einzeln freigeben oder sperren können.



Wenn Sie gleichzeitig die Product Item Option License Quantity auf einen Wert von 1 setzen, kann der Anwender nur eine der freigeschalteten Versionen gleichzeitig starten. Natürlich funktioniert dies auch im Netzwerk mit mehr als einer Lizenz.

4.3.1.8 Overflow

Overflow-Lizenzen umfassen die Bereitstellung zusätzlicher, nach Nutzung abgerechnete Lizenzen zur Absicherung eines kurzfristigen, erhöhten Lizenzbedarfs.



Definieren Sie zwei Lizenzentriäge auf der Product Item-Ebene mit zwei verschiedenen Product Codes für den Haupt-Eintrag und den Overflow-Eintrag. Der Haupt-Eintrag enthält keinen Unit Counter und eine License Quantity entsprechend der Anzahl der gekauften Lizenzen. Der Overflow-Eintrag enthält hingegen einen hohen Unit Counter und eine License Quantity in Höhe der gewünschten Overflow-Lizenzen.

Wenn alle Haupt-Einträge belegt sind, verwenden Sie in der Software die Overflow-Einträge. Sie können dann selbst entscheiden, ob Sie dies in der Software anzeigen und die Software in diesem Fall künstlich verlangsamten wollen. Zusätzlich können Sie regelmäßig den Unit Counter kontrollieren und so nachweisen, wie häufig (oder wie lange) die Overflow Lizenzen verwendet wurden.



Bei Overflow-Lizenzen müssen Sie nicht auf den Schutz durch AxProtector verzichten. Legen Sie diesen mit dem Product Code des Haupt-Eintrages um die Software und stellen Sie den Zugriffsmodus auf NoUserLimit.

4.3.1.9 Hot / Cold Standby

Lizenzmodelle im Bereich der Ausfallsicherheit umfassen Verfahren des Cold- und Hot Standby.

Cold-Standby

Beim Cold-Standby wird neben dem eigentlichen *CmDongle* ein zweiter Backup-*CmDongle* bereithalten, der aber nicht aktiv ist. Wenn der erste *CmDongle* ausfällt, kann der Backup-*CmDongle* verwendet werden.



Definieren Sie auf der Product Item-Ebene einen Nutzungszeitraum über die Product Item Option Usage Period. Sie geben Ihren Kunden einen Backup *CmDongle* mit einer Usage Period von einigen Tagen. Bei der ersten Nutzung dieses Lizenzentriags wird die Usage Period gestartet. Die Lizenz ist ab diesen Augenblick verfügbar und sperrt sich nach Ablauf automatisch selbst. Damit kann diese Lizenz zur Überbrückung verwendet werden, aber nicht als vollwertige zweite Lizenz. Die Überprüfung der Usage Period erfolgt mittels der Echtzeituhr im *CmDongle*.

Hot-Standby

Bei einem Hot-Standby steht ebenfalls ein Backup-*CmDongle* bereit, der aber neben dem eigentlichen

CmDongle parallel in Betrieb gehalten wird. Nur bei Ausfall wird der Backup-*CmDongle* verwendet.



Definieren Sie zwei Product Item-Ebenen mit zwei verschiedenen Product Codes für die Haupt-Lizenz und die Backup-Lizenz in zwei separate *CmDongles*. Die Hauptlizenz enthält keinen Unit Counter. Die Backup-Lizenz realisieren Sie über einen sehr hohen Unit Counter für den zweiten *CmDongle* auf einem zweiten Rechner. Den *CmDongle* mit der Haupt-Lizenz und ohne Unit Counter schließen Sie an den Lizenzserver, und den *CmDongle* mit der Backup-Lizenz und hohem Unit Counter an den Backup-Server.

Über die [Serversuchliste](#)⁴⁷¹ legen Sie die Reihenfolge der zu belegenden Lizenzen fest. Im Fall, dass der erste Server ausfällt, wird automatisch der zweite Server mit den Backup-Lizenzen verwendet. Sie kontrollieren regelmäßig den Zählerstand, um einen Missbrauch zu vermeiden.

4.3.1.10 Named User Lizenzen

Bei einer Named User-Lizenz ist die Verwendung einer Software namentlich an einen Nutzer gebunden, der sich zudem erfolgreich authentifizieren muss.



Definieren Sie auf der Product Item-Ebene die Product Item Option Protected Data und speichern Sie die User-ID dort ab.

In der Software vergleichen Sie dann, ob die eigen gespeicherte User-ID mit der aktuell für diesen Nutzer berechneten User-ID übereinstimmt.

4.3.1.11 Rechnergebundene Lizenzen

In einigen Fällen kann es notwendig sein, einen *CmContainer* an einen PC, eine Maschine oder einen Benutzer zu binden.



Definieren Sie auf der Product Item-Ebene die Product Item Option Protected Data und speichern Sie die ID dort ab.

In der Software vergleichen Sie, ob die eigene gespeicherte Host-ID mit der aktuell für den Rechner berechneten übereinstimmt.

4.3.1.12 Lizenzausleihe

Das Lizenzmodell der Lizenzausleihe ermöglicht die Nutzung von Software-Anwendungen auf Rechnern auch ohne Verbindung zu einem Lizenzserver im Netzwerk. Dabei wird die Lizenz zeitlich begrenzt ausgeliehen. Die Gesamtanzahl der im Netzwerk verfügbaren Lizenzen bleibt davon aber unberührt. Diese Lizenz-Mobilität ist beispielsweise gefragt, wenn Lizenzen auf einem separaten Laptop für unterwegs, oder im Home Office zur Verfügung stehen sollen.



Für die Lizenzausleihe benötigen Sie [vorprogrammierte](#)³⁷⁶ *CmContainer* auf der Server- und der Client-Seite.

Der Ausleih- und Rückgabevorgang einer Lizenz für den Anwender erfolgt über den Karteireiter "[Ausleihe](#)⁴⁵³" in [CodeMeter Kontrollzentrum](#). In [CodeMeter WebAdmin](#) wird die [Belegung der Lizenzen](#)⁴⁸⁸ angezeigt wobei Anzahl und die [maximale Ausleihdauer](#)⁴⁸² konfiguriert werden können.

4.4 Sicherheit durch Verschlüsselung

Die Sicherheit von *CodeMeter*[®] basiert auf Ver- und Entschlüsselungsvorgängen. Die zu schützende Software, bzw. Teile der Software oder Daten in der Software werden vor der Auslieferung durch den Lizenzgeber verschlüsselt. Der Schlüssel zum Entschlüsseln ist in der Lizenz enthalten, die der Lizenzgeber

für den Lizenznehmer erstellt. Auf der Seite des Lizenznehmers werden in der Software nur die Teile entschlüsselt, die gerade benötigt werden (On-Demand-Decryption). Nach der Verwendung können diese Teile wieder verschlüsselt werden.

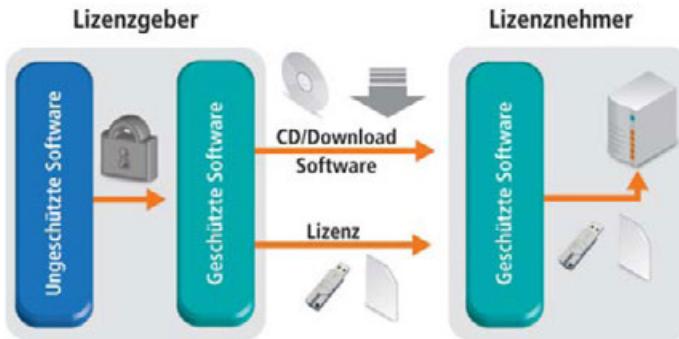


Abbildung 9: Sicherheit durch Verschlüsselung

4.4.1 Schlüssableitung: ein Lizenzeintrag - viele Schlüssel

Die Software wird zur Laufzeit auf dem PC des Lizenznehmers entschlüsselt. Die Kommunikation zwischen der Software und der Lizenz ist dabei bis in den *CmContainer* hinein verschlüsselt. Diese Schlüssel werden im *CmContainer* durch eine Ableitung gebildet.

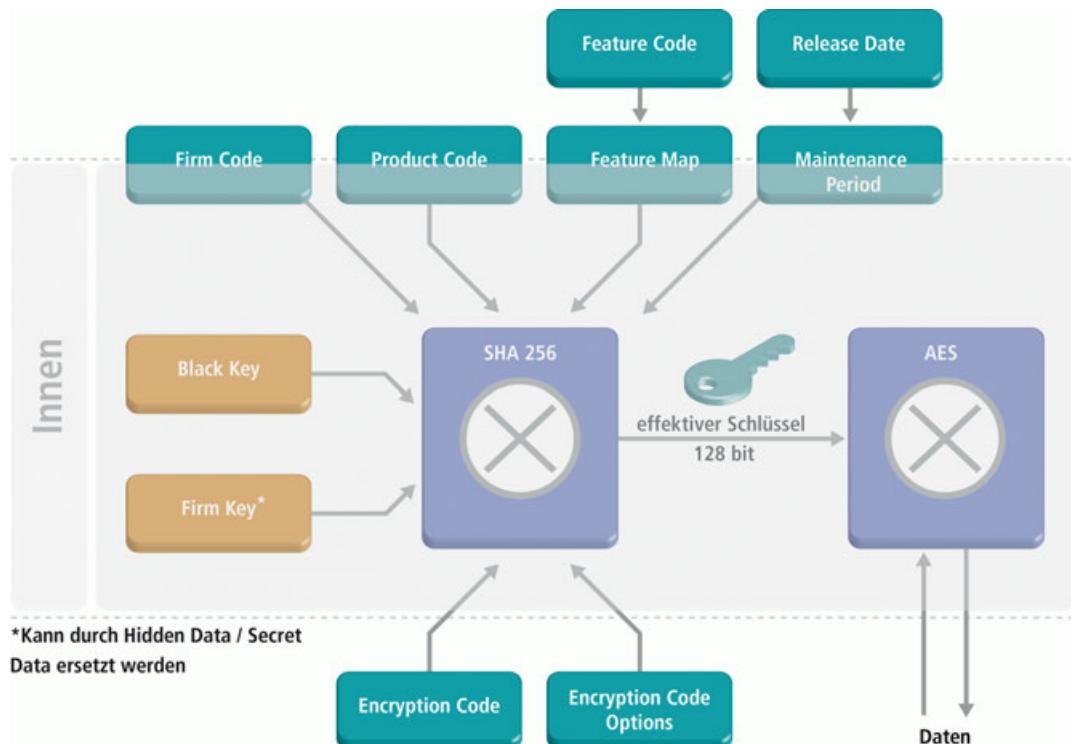


Abbildung 10: Schlüsselableitung

Der effektive Schlüssel

Der "effektive" Schlüssel zum Ver- und Entschlüsseln setzt sich aus mehreren *CmContainer*-internen und -externen Parametern zusammen. Innerhalb des *CmContainer* wird er dann über eine Hash-Funktion (SHA 256) berechnet.

Black Key, Firm Key

Zunächst gibt es zwei nicht auslesbare Parameter innerhalb des *CmContainer*. Der Black Key ist ein geheimer Schlüssel, der nur Wibu-Systems bekannt ist. Der Firm Key wird in der Regel von Wibu-Systems initial ausgeliefert, ist danach aber durch den Lizenzgeber änderbar.



Alternativ kann hier statt dem Firm Key, auch ein Secret Data- oder Hidden Data-Feldeintrag verwendet werden.

Firm Code, Product Code, Feature Code

Zusätzlich existieren vier weitere Parameter, die sich im *CmContainer* befinden. Sie können aber von außen in den *CmContainer* für eine Lizenz programmiert werden: Firm Code, Product Code, Wartungszeitraum (Maintenance Period), bei dem ein Erstelldatum (Release Date) festgelegt wird und die Feature Map, die das Setzen von Features über einzelne Bits mit der Festlegung des Feature Codes ermöglicht.

Encryption Code

Diese ersten Parameter fließen fest für einzelne Lizenzentitäten in die Schlüsselableitung ein. Im Gegensatz dazu sind die zwei weiteren Parameter Encryption Code und Encryption Code Options (ECO) dynamisch. Sie fließen variabel wie für die entsprechende Lizenz definiert zur Laufzeit der geschützten Anwendung in die Schlüsselableitung ein.

Der Encryption Code ist ein frei wählbarer fester Wert, der vom Lizenzgeber für Ver- und Entschlüsselungsvorgänge angegeben wird.

Encryption Code Options

Die Encryption Code Options enthalten Informationen darüber, ob bestimmte Product Item Options bei der Schlüsselableitung zur Laufzeit vorhanden sein müssen und wie sie überprüft werden. Sie umfassen u.a. die folgenden Bestandteile (für die detaillierte Beschreibung zur Verwendung der einzelnen Optionen siehe in der *CodeMeter API Guide* Online-Hilfe das Kapitel "[Functions | Encryption API | CmCrypt2](#)"):

- Product Item Options Unit Counter, Activation Time und Expiration Time
- Zugriffsmodi der gestarteten Instanzen auf verfügbare Lizizen
- Wert, um den ein Unit Counter bei einer bestimmten Aktion heruntergezählt werden soll (Delta-Maske),
- Abfrage, ob seit Anstecken des *CmDongle* eine Aktualisierung vom zertifizierten Zeitserver geholt wurde,
- Herunterzählen eines speziellen Zählers, der bei einer Debugger-Erkennung um den Wert 1 reduziert wird (Firm Access Counter, FAC).

Hash Funktion (SHA 256)

Aus diesen insgesamt sieben Parametern wird über eine Hash Funktion (SHA 256, Secure Hash Algorithmus) der effektive Schlüssel berechnet, der dann innerhalb des symmetrischen Verschlüsselungsverfahrens AES (Advanced Encryption Standard) verwendet wird.

Um zu überprüfen, dass Daten tatsächlich mit dem gleichen effektiven Schlüssel ver- und entschlüsselt wurden, wird beim Verschlüsseln eine Prüfsumme (CRC, Cyclic Redundancy Check) gebildet, die beim Entschlüsseln gegengeprüft wird. Finden nun nicht erlaubte Manipulationen statt, z.B. über den Versuch das Herunterzählen eines Unit Counter auszuhebeln, bedeutet dies eine Veränderung von Parametern zwischen Ver- und Entschlüsselung. Dies führt nachfolgend zu einer anderen Prüfsumme (CRC) und die Entschlüsselung wird mit einer Fehlermeldung abgebrochen.

Dieser Vorgang der Schlüsselableitung wird im *CmContainer* durchgeführt und die abgeleiteten Schlüssel können von allen *CodeMeter®*-Werkzeugen und -schnittstellen zur Ver- und Entschlüsselung verwendet werden.

4.5 Kryptographie

Kryptographische Methoden und Verfahren umfassen im Allgemeinen die folgenden Ziele:

- Integrität: Inhalte dürfen nicht geändert werden.
- Vertraulichkeit: Das Lesen eines eigentlichen Inhalts soll für Unbefugte praktisch unmöglich gemacht werden.
- Authentifizierung: Der Sender einer Nachricht beweist dem Empfänger gegenüber seine Identität.
- Verbindlichkeit: Der Empfänger kann den Nachweis erbringen, dass der Sender die Nachricht mit identischem Inhalt abgeschickt hat.

CodeMeter® bietet viele kryptographische Verfahren an, die diese Ziele erfüllen.



AxProtector und IxProtector arbeiten bei der symmetrischen Ver- und Entschlüsselung mit AES (Advanced Encryption Standard) mit einer Schlüssellänge von 128 Bit.

Asymmetrische Ver- und Entschlüsselungen werden über ECC (Elliptic Curve Cryptography) 224-Bit und RSA mit mindestens 2048-Bit durchgeführt..

Im CodeMeter Kern-API haben Sie die Möglichkeit über die **CmCrypt** Funktion unterschiedlichste Ver- und Entschlüsselungsalgorithmen anzuwenden. Es sind symmetrische Verfahren, aber auch asymmetrische Verfahren für Signaturen und Public-Key Strukturen einsetzbar.

4.5.1 Direkte und indirekte Verschlüsselung

Bei CodeMeter® greift zunächst die Unterscheidung, ob ein Ver- oder Entschlüsselungsvorgang direkt oder indirekt stattfindet. Dies hat Einfluss auf die Geschwindigkeit der Verschlüsselung.

Direkte Verschlüsselung

Bei der direkten Verschlüsselung erfolgt der Vorgang im *CmContainer* selbst. Die zu verschlüsselnden Daten mit einer exakten Datenlänge von 16 Byte werden über die Verschlüsselungseinheit im *CmContainer* verschlüsselt.



Sinnvoll ist die Verwendung bei zufälligen Prüfungen oder bei Sequenzen mit kurzer Länge.

Indirekte Verschlüsselung

Bei der indirekten Verschlüsselung wird zunächst ein Teil der Daten direkt im *CmContainer* verschlüsselt. Danach geht dieses Ergebnis als Initialisierungsvektor in den Rest der Verschlüsselung ein, die im Speicher des PC erfolgt (Schlüssel zur Verschlüsselung).



Die Mindestlänge der Daten beträgt 16 Byte, die Maximallänge 4 GByte.

4.5.2 Symmetrische Verschlüsselung

Bei symmetrischen kryptographischen Verfahren wird für Ver- und Entschlüsselungsvorgänge derselbe Schlüssel verwendet (siehe [Verschlüsselungs-API](#)³³⁵).

AES

CodeMeter® setzt für diese Verfahren den Standardalgorithmus zur symmetrischen Verschlüsselung von Daten ein: den AES (Advanced Encryption Standard).



In CodeMeter® wird AES mit einer Schlüssellänge von 128 Bit = 16 Byte verwendet.

Es gibt zwei grundlegende Arten von Algorithmen bei symmetrischen Ver- und Entschlüsselungsvorgängen: Stromchiffren und Blockchiffren.

Stromchiffren

Bei einem Stromchiffre wird der Klartext nicht in Blöcken, sondern zeichenweise verschlüsselt. Hierfür wird jedes Zeichen des Klartextes mit einem Zeichen des Schlüssels verknüpft.

4.5.2.1 Streamcipher (AES_STREAM)

CodeMeter® verwendet für diesen Modus den AES-Algorithmus als Zufallszahlengenerator. Die entstehende Folge wird über ein exklusives Oder (XOR) mit dem Klartext verknüpft.

Das bedeutet, dass keine separate Entschlüsselfunktion notwendig ist, denn Ver- und Entschlüsselung sind identisch. Wird eine Folge, d.h. ein Schlüssel, zweimal verwendet, so ist das Verfahren unsicher und kann gebrochen werden. Wird bei einer Übertragung nur ein Zeichen im Chiffrat geändert, so ändert sich im Klartext auch nur ein Zeichen. Das Verfahren hat daher die geringste Fehlerausbreitung. In der Fachliteratur wird dieser Modus oft auch als "Output Feedback Mode" beschrieben.

Blockchiffren

Bei Blockchiffren wird der Klartext in Blöcke mit festen Blöcken aufgeteilt. Auf jeden Block wird dann der Schlüssel angewendet, sodass aus jedem Klartextblock ein Geheimtextblock entsteht.

4.5.2.2 Electronic Codebook Mode (AES_ECB)

Der AES-Algorithmus mit Electronic Code Book Modus bewirkt, dass alle Klartextblöcke in Blöcken fester Länge unabhängig voneinander verschlüsselt werden. Auf jeden Block wird dann der Schlüssel angewendet, so dass aus jedem Klartextblock ein Geheimtextblock entsteht. Gleiche Klartextblöcke ergeben daher bei gleichem Schlüssel auch immer den gleichen Geheimtextblock, wodurch man bei hinreichend vielen Geheimtextblöcken und partiellen Annahmen über den Klartext Rückschlüsse auf den geheimen Schlüssel ziehen kann.



Ohne das Vorliegen zwingender Gründe sollte von einem Einsatz abgesehen werden.

4.5.2.3 AES - Cipher Block Chaining Mode (CBC) (empfohlen)

Beim AES-Algorithmus im Cipher Block Chaining Modus wird vor dem Verschlüsseln eines Klartextblocks dieser zunächst mit dem im letzten Schritt erzeugten Geheimtextblock per XOR (exklusives Oder) verknüpft. Wird ein Zeichen im Klartextblock geändert, so ändert sich der gesamte nachfolgende Geheimtextblock. Die Entschlüsselung muss jeweils am Anfang des verschlüsselten Blockes gestartet werden. Eine Entschlüsselung eines beliebigen Teils in der Mitte des Chiffrats ist nicht einfach möglich.

Sowohl zur Ver- als auch zur Entschlüsselung muss ein gleicher Startwert (IV = Initial Vector) angegeben werden. Der IV muss nicht vertraulich sein. Wird er konstant gewählt, so werden gleiche Klartexte (Anfänge) zu gleichen Chiffren verschlüsselt.



Wibu-Systems empfiehlt diesen Modus, wenn nicht aus zwingenden Gründen ein anderer Modus gewählt werden soll.

4.5.2.4 AES - Cipher Feedback Mode (CFB)

Beim AES-Algorithmus im Cipher Feedback werden Klartexte verschlüsselt, die länger als die Blocklänge des Chiffrierverfahrens sind. Eine Entschlüsselung ist aber nur vom Anfang des Chiffrates her möglich. Sowohl zur Ver- als auch zur Entschlüsselung muss der gleiche Startwert (IV = Initial Vector) angegeben werden. Der IV muss nicht vertraulich sein. Wird er konstant gewählt, so werden gleich Klartexte (Anfänge) zu gleichen Chiffren verschlüsselt. Der CFB ist im Vergleich zu den anderen Modi deutlich langsamer, da für jedes Byte ein getrennter Aufruf zur Verschlüsselung gestartet wird.



Dieser Modus sollte nur in begründeten Ausnahmefällen gewählt werden.

4.5.3 Asymmetrische Verschlüsselung

Neben der symmetrischen Verschlüsselung bietet CodeMeter® auch die Möglichkeit, Daten über private und öffentliche Schlüssel asymmetrisch zu ver- und entschlüsseln, Signaturen für die Authentizitätsprüfung zu erzeugen und zu verifizieren. CodeMeter API Guide bietet hierzu die notwendigen API Befehle und Funktionsblöcke: [Authentifizierungs-API](#)³³⁴, [Verschlüsselungs-API](#)³³⁵, [Blöcke](#)³⁴⁰ für die Durchführung verschiedener Ver- und Entschlüsselungsoperationen.

4.5.3.1 ECC - Elliptic Curve Cryptography

Die ECC (Elliptic Curve Cryptography) umfasst asymmetrische Kryptographie auf Basis elliptischer Kurven. Hierbei verfügen beide Seiten über unterschiedliche Schlüssel. Der private Schlüssel verlässt den CmDongle nie. Aus ihm kann der öffentliche Schlüssel berechnet werden. Dieser muss authentisch (jedoch nicht vertraulich) bei der Gegenseite hinterlegt und gespeichert werden.

4.5.3.2 ECIES - Elliptic Curve Integrated Encryption Scheme

Das ECIES (Elliptic Curve Integrated Encryption Scheme) ist ein Public-Key-Verschlüsselungsschema mit dem mit der Kenntnis des öffentlichen Schlüssels Daten an den Besitzer des privaten Schlüssels (im CmDongle) geschickt werden kann. Üblicherweise werden die Daten mit der Funktion **CmCryptEcies** verschlüsselt und mit **CmCrypt** und dem Algorithmus CM_CRYPT_ECIES_STD wieder entschlüsselt.

4.5.3.3 ECDSA - Elliptic Curve Digital Signature Algorithm

Der ECDSA (Elliptic Curve Digital Signature Algorithm) ist ein Signaturalgorithmus, der aus einem Dokument einen Hash-Wert (Digest) bildet. Dieser wird dann mit dem privaten Schlüssel signiert. Ein Dokument (bzw. einen Hash-Wert) mit einem privaten Schlüssel assoziiert. Im Gegensatz zum ECIES wird hier der private Schlüssel zur Erzeugung der Signatur eingesetzt und der öffentliche Schlüssel zur Verifikation.

Siehe die relevanten Funktionen **CmCalculateDigest**, **CmCalculateSignature**, **CmValidateSignature** und **CmGetPublicKey** des [Authentifizierungs API](#)³³⁴, die für die Durchführung von Authentifizierungsvorgängen benötigt werden.

4.5.3.4 RSA

Der RSA benannt nach seinen Erfindern (Ron Rivest, Adi Shamir und Leonard Adleman) ist ein asymmetrisches Kryptosystem, das sowohl zur Verschlüsselung als auch zur digitalen Signatur verwendet werden kann.

4.5.4 Zusätzliche Verschlüsselungen

Zusätzlich werden noch zwei weitere Verschlüsselungsverfahren eingesetzt.

Certified Time Encryption

Diese Verschlüsselung betrifft die zertifizierte Zeit (Certified Time) und stellt eine Spezialfunktion für den CodeMeter®-Zeitserver dar.

SHA - Secure Hash Algorithm 256

Der SHA ist ein kryptographischer Hash-Algorithmus, der eine 256 Bit (32 Byte) lange kryptographische Prüfsumme erzeugt, die als eine Art "Fingerabdruck" eingesetzt werden kann. Im Bereich der asymmetrischen Verschlüsselung wird der SHA-256 zur Vorbereitung einer Signatur verwendet, um aus der zu signierenden Nachricht einen Kontrollwert konstanter Länge zu berechnen.

5 CodeMeter Start Center

CodeMeter Start Center dient als Kommunikationszentrale und ermöglicht den Zugriff auf die zentralen CodeMeter®-Werkzeuge, Anwendungen und Schnittstellen.

5.1 Aufbau und Navigation

Sie erreichen CodeMeter Start Center über das Systemmenü "**Start | Alle Programme | CodeMeter**". Die Benutzeroberfläche teilt sich in zwei Bereiche: in eine obere Menüsteuerung und in ein unteres Fenster, das den Zugriff auf die einzelnen Anwendungen ermöglicht.

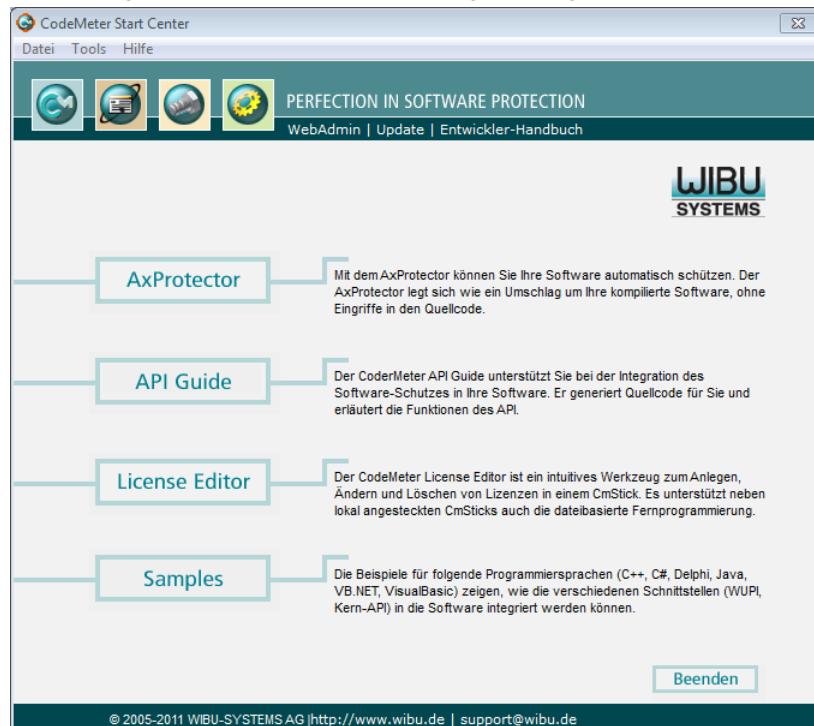


Abbildung 11: CodeMeter Start Center

5.1.1 Menüleiste

Das Datei-Menü

Element	Beschreibung
Sprache	CodeMeter Start Center bietet Ihnen verschiedene Sprachversionen für die Bedienoberfläche.

Element	Beschreibung
	che. Sie können derzeit unter 8 Sprachen wählen: Chinesisch, Deutsch, Englisch, Spanisch, Französisch, Japanisch, Niederländisch, Portugiesisch.
Beenden	Über den Menü-Eintrag " Beenden " beenden Sie <i>CodeMeter Start Center</i> . Alternativ können Sie das Fenster über die "x" Schaltfläche oder die Schaltfläche " Beenden " im <i>CodeMeter Start Center</i> schließen.

Das Tools-Menü

Neben den Werkzeugen, Anwendungen und Schnittstellen, die Sie auch über das *CodeMeter Start Center*-Fenster aufrufen, umfassen das Menü auch das Öffnen von *CodeMeter WebAdmin*, um vorhandene Lizenzen in *CmContainern* einzusehen

Das Hilfe-Menü

Element	Beschreibung
Update suchen	Informiert über Software-Aktualisierungen auf der Wibu-Systems Internet-Präsenz.
Entwickler-Handbuch	Öffnet dieses Dokument.
Wibu-Systems Homepage	Verweist auf die Wibu-Systems Internet-Präsenz.
Mail an Support	Öffnet eine E-Mail adressiert an den Support von Wibu-Systems.
Über	Öffnet ein Fenster mit Versionsinformationen 

Fenster

Element	Beschreibung
AxProtector	AxProtector ⁷¹ schützt Ihre Software automatisch. AxProtector legt sich wie ein Umschlag um Ihre kompilierte Software, ohne Eingriffe in den Quellcode vorzunehmen.
API Guide	CodeMeter API Guide ³³⁷ unterstützt Sie bei der Integration des Software-Schutzes in Ihre Software. Er generiert Quellcode für Sie und erläutert die Funktionen des <i>CodeMeter Kern-API</i> sowie des Softwareschutz-API (WUPI) ³²⁰ .
CodeMeter License Editor	CodeMeter License Editor ³⁴⁶ ist ein intuitives Werkzeug zum Anlegen, Ändern und Löschen von Lizenzen in einem <i>CmDongle</i> . Es unterstützt neben lokal angeschlossenen <i>CmDongles</i> auch die dateibasierte Fernprogrammierung ³⁹² .
Samples	Hier finden Sie für verschiedene Programmiersprachen Beispiele für die Integration der Schnittstellen in die Software (<i>Softwareschutz-API (WUPI)</i> , <i>Kern-API</i>). Durch Klicken der Schaltfläche gelangen Sie in die betreffenden Verzeichnisse und erhalten eine kurze Übersicht über bestehende Beispiele.
Beenden	Über die " Beenden " Schaltfläche schließen Sie <i>CodeMeter Start Center</i> .

6 CodeMeter Lizenzserver

Die zentrale Komponente von *CodeMeter®* ist *CodeMeter Lizenzserver* (*CodeMeter.exe*), der als Dienst auf jedem Rechner läuft, auf dem eine durch *CodeMeter®* geschützte Software verwendet werden soll. *CodeMeter Lizenzserver* stellt somit die Schnittstelle zwischen Ihrer Software und der Lizenz in einem *CmContainer* zur Verfügung.

Viele Dongle-Hersteller verwenden für direkte Zugriffe auf Dongles separate Bibliotheken. Wibu-Systems beschreitet andere Wege. Hier übernimmt der für die Lizenzierungssysteme *CmDongle* und *CmActLicense* standardisierte *CodeMeter Lizenzserver* als zentrale Drehscheibe diese Kommunikationsaufgaben. *CodeMeter Lizenzserver* kommuniziert nach unten über die betriebssystemeigenen USB-Treiber (als [Mass Storage oder Human Interface Device, HID](#)⁵¹⁰) mit dem *CmContainer*, und stellt nach oben die Schnittstelle zu der mit *CodeMeter®* geschützten Software zur Verfügung.

Nahtlose, integrierte und sichere Zugriffsverwaltung

CodeMeter Lizenzserver verwaltet die Zugriffe von Anwendungen auf *CmContainer*. Dabei spielt es keine Rolle, ob mehrere Anwendungen gleichzeitig auf einen *CmContainer* zugreifen, oder Anwendungen Lizenzinformationen benötigen, die auf verschiedene *CmContainern* verteilt sind. *CodeMeter Lizenzserver* läuft als Dienst im Hintergrund und die gesamte Kommunikation von und zu *CodeMeter Lizenzserver* ist sicher verschlüsselt.

Zukunftssicherheit

Den Einsatz zukünftiger Schnittstellen deckt *CodeMeter Lizenzserver* ebenfalls ab. Eine heute verschlüsselte Software läuft in Zukunft z.B. auch mit einer SD-oder CF Card. Das Anpassen Ihrer Software durch das Programmieren einer neuen Schnittstelle entfällt. *CodeMeter Lizenzserver* gewährleistet automatisch, dass Ihre Anwendung lauffähig ist. Darüber hinaus ist die Abwärtskompatibilität in jedem Fall garantiert. Mit der jeweils neuesten Version von *CodeMeter Lizenzserver* sind alle bereits ausgelieferten Versionen Ihrer Software lauffähig. Ohne dass sie dafür Ihre Software neu kompilieren müssen.

Automatische Lizenzvergabe - lokal und im Netz

CodeMeter Lizenzserver sorgt nicht nur am lokalen PC automatisch für die Verwaltung von Lizzenzen. Als [Netzwerkserver eingerichtet](#)⁴⁷⁴, ist er auch in der Lage die verfügbaren Lizzenzen im gesamten Netz zur Verfügung zu stellen. Wird die maximale Lizenzanzahl erreicht, startet eine weitere Anwendung nicht. Die Zuteilung der Lizzenzen kann in verschiedenen Betriebsmodi erfolgen. Im Normalfall belegt jede gestartete Anwendung eine Lizenz. Über die Lizenzoption "station share" können Sie aber beispielsweise auch festlegen, dass pro Rechner die Anwendung beliebig häufig gestartet werden kann, aber nur als eine Lizenz pro Rechner gezählt wird. In diesem Modus wird jede Terminalserver-Session und jede Virtual Machine wie ein separater PC gezählt.

Seit der *CodeMeter®*-Version 5.0 unterstützt die [Netzwerkkommunikation](#)⁴⁷⁴ auch Weitverkehrsnetze WAN (Wide Area Networks). Für weitere Details siehe [hier](#)⁴²³.

Automatische und manuelle Freigabe von Lizzenzen

Sollte es in Ausnahmefällen zu ungewollten Abstürzen Ihrer Anwendung kommen, so gewährleistet *CodeMeter Lizenzserver* durch die ständige Überprüfung der angemeldeten Anwendungen automatisch, dass die Lizzenzen wieder freigegeben werden. Zusätzlich hat der Administrator die Möglichkeit, die Lizzenzen auch manuell wieder [freizugeben](#)⁴⁹⁰.

CodeMeter Kontrollzentrum und CodeMeter WebAdmin

Lokale Konfigurationseinstellungen für *CodeMeter Lizenzserver* können Sie im [CodeMeter](#)

[Kontrollzentrum](#)⁴⁴³ vornehmen. [CodeMeter WebAdmin](#)⁴⁶⁴ erlaubt zusätzlich, Informationen über belegte Lizenzen im Netzwerk einzusehen.

Zur Kommunikation zwischen allen Komponenten wird das TCP/IP Netzwerkprotokoll eingesetzt.

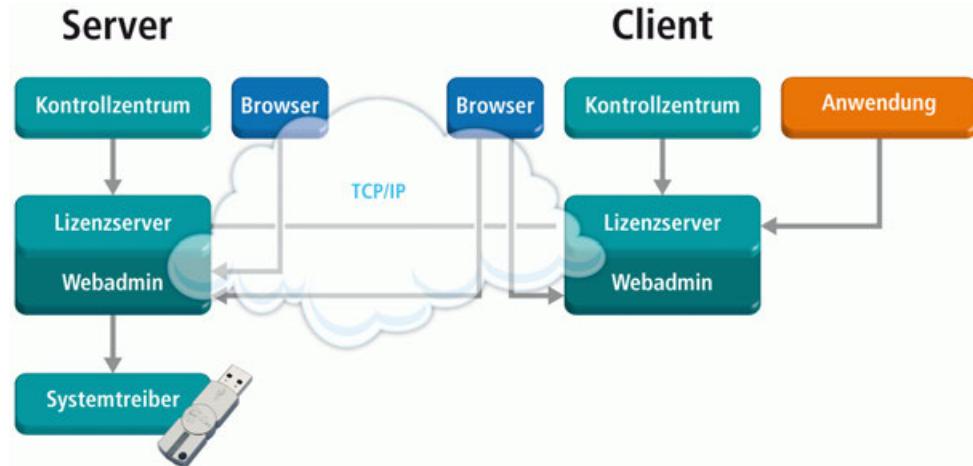


Abbildung 12: CodeMeter Lizenzserver

Vielfalt der Betriebssysteme

CodeMeter Lizenzserver ist verfügbar für Betriebssysteme, wie Windows 7, 8, Vista, Windows XP, Windows CE, Windows Embedded, Mac OS X, Linux (verschiedene 32- und 64-Bit Derivate), Sun Solaris 10 und VxWorks.

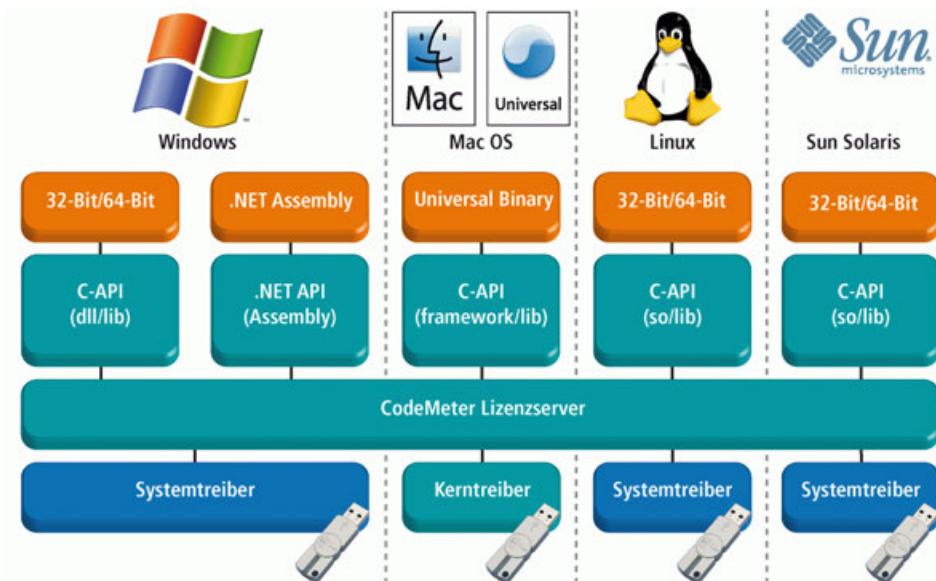


Abbildung 13: *CodeMeter Lizenzserver* und Betriebssysteme

Auch in heterogenen Systemlandschaften läuft *CodeMeter Lizenzserver* problemlos. So kann z.B. *CodeMeter Lizenzserver* als Netzserver unter Linux laufen, während Ihre Software auf den Betriebssystemen Windows und Mac OS verwendet wird.

7 Automatischer Softwareschutz mit AxProtector

Keine Programmierkenntnisse erforderlich

Mit *AxProtector* steht Ihnen ein Werkzeug zur Verfügung, das automatisch ausführbaren Code von Anwendungen verschlüsselt. Damit integrieren Sie *CodeMeter®* problemlos und schnell in Ihre Anwendung, ohne dafür Eingriffe in den Quelltext der Anwendung vornehmen zu müssen. Über eine einfach zu bedienende Anwendungsoberfläche mit den wichtigsten Einstellungsoptionen bedeutet automatischer Softwareschutz mit *AxProtector* sichere Integration ohne Programmierkenntnisse.

AxProtector verschlüsselt und schützt Ihre Anwendung in wenigen Minuten und dies für verschiedene Projekttypen⁷⁵:

AxProtector gibt es ebenfalls in einer Kommandozeilen-Variante²⁹² für Windows 32-Bit / 64-Bit, .NET, Linux-, Mac OS X- und Java-Anwendungen. Die Verschlüsselung über die *AxProtector*-Anwendungsoberfläche erzeugt dabei eine Kommandozeile, die ergänzt und weiterverwendet werden kann.

Die folgende Tabelle fasst zusammen, welche Anwendungen wie über unterschiedliche Projekttypen und Werkzeuge für verschiedene Betriebssysteme mit *AxProtector* verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttypen	GUI Windows	Kommandozeile
Windows Anwendung oder DLL	 AxProtector Windows ⁷⁶  IxProtector Windows ²¹⁸	✓	Windows Kommandozeile
.NET Assembly	 AxProtector .NET ¹¹⁰  IxProtector .NET ²³²	✓	.NET Kommandozeile
Mac OS X Anwendung oder Dylib	 AxProtector Mac OS X ¹⁴⁰  IxProtector Mac OS X ²⁴⁷	✓	Windows Kommandozeile Kommandozeile für Mac OS X (auf Mac OS X Betriebssystemen verwendbar)
Java Anwendung (Archiv-Format *.jar, Webarchiv-Format *.war)	 AxProtector Java ¹⁶⁸	✓	Windows Kommandozeile Kommandozeile für Java (auf Windows, Mac OS X, Linux und Solaris Betriebssystemen verwendbar)
Linux Anwendung oder Shared Object	 AxProtector Linux ¹⁹⁰  IxProtector Linux ²⁶¹	✓	Windows Kommandozeile Kommandozeile für Linux (auf Linux Betriebssystemen verwendbar)
Daten-Dateien, die Ihre geschützte Anwendung verwendet	 AxProtector Dateiverschlüsselung ²⁷⁵	✓	Windows Kommandozeile

Tabelle 5: *AxProtector* – Zu verschlüsselnde Anwendungen, Projekttypen und Verschlüsselungswerzeuge

AxProtector:

- unterstützt alle vorhandenen CodeMeter®-Lizenzoptionen (Product Item Options). Dadurch fließen alle notwendigen Lizenzinformationen in die Verschlüsselung ein, wie zum Beispiel die Verwendung von Lizenzen im Netz, oder die Durchführung von Lizenzüberprüfungen während der Laufzeit der Anwendung.
- verfügt über Funktionen zur Erkennung von Debuggern: wird ein Debugger erkannt, so kann ein CmContainer gesperrt werden.
- bietet bei der Entschlüsselung das Feature der "On-Demand-Decryption" an, d.h. erst bei Bedarf werden benötigte Teile der geschützten Anwendung im Speicher entschlüsselt, der Rest verbleibt verschlüsselt im Speicher. Das verhindert die Extraktion einer ungeschützten Version der Software. Ein Memory Dumping ist somit nicht möglich.
- ermöglicht die Verwendung frei konfigurierbarer Benutzerdialoge, die das Erstellen eigener Texte für Kaufoptionen oder Fehler, aber auch die Einbettung eines Firmenlogos umfassen.

7.1 Aufbau und Navigation

Am besten öffnen Sie AxProtector über [CodeMeter Start Center](#)⁶⁶ oder alternativ über das Systemmenü "Start | Alle Programme | AxProtector".

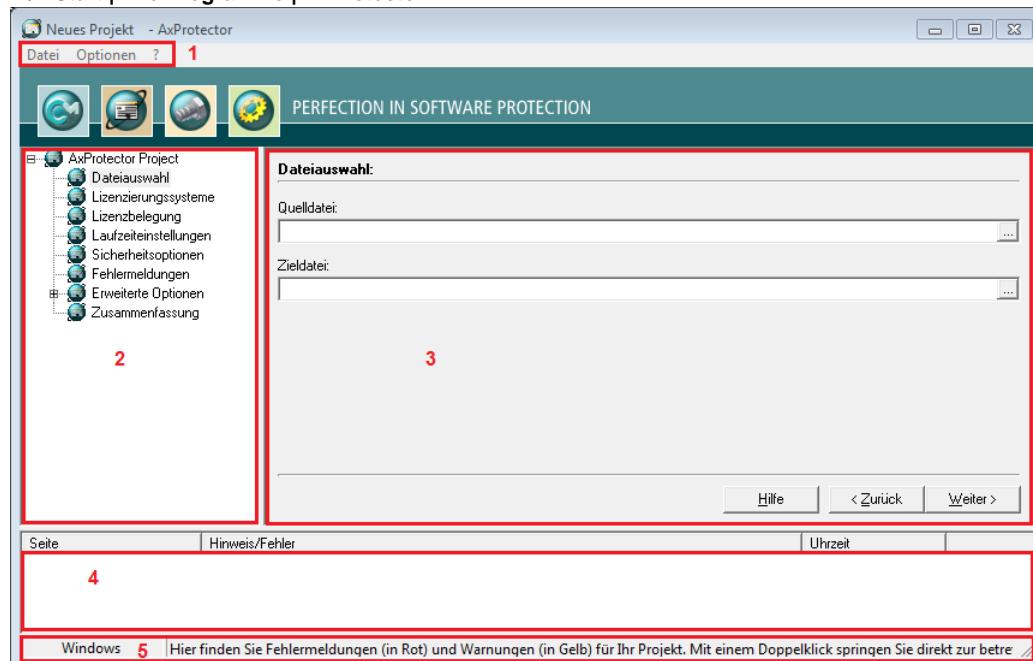


Abbildung 14: AxProtector – Oberfläche und Navigation

Die AxProtector-Benutzeroberfläche setzt sich aus fünf separaten Bereichen zusammen:

- [Menüleiste⁷³](#) (1)
- [Navigationsfenster⁷⁴](#) (2)
- [Eingabefenster⁷⁴](#) (3)
- [Warnungs-/Fehler-Fenster⁷⁴](#) (4)
- [Projektyp-Fenster⁷⁵](#) (5)

7.1.1 Menüleiste

Das Datei-Menü

Element	Beschreibung
Neues Projekt	<p>Um ein neues Projekt anzulegen, gehen Sie wie folgt vor:</p> <p>1. Wählen Sie den "Datei Neues Projekt" Menü-Eintrag. Oder drücken Sie alternativ die <STRG+N> Tastenkombination. Der "Neues Projekt" Dialog öffnet sich zur Auswahl des Projektyps.</p> <p>Projekt öffnen</p> <p>Um ein vorhandenes Projekt zu öffnen, gehen Sie wie folgt vor:</p> <p>1. Wählen Sie den "Datei Projekt öffnen" Menü-Eintrag. Oder drücken Sie alternativ die <STRG+O> Tastenkombination. Es erscheint der "Öffnen" Systemdialog, aus dem Sie die gewünschte Datei wählen können.</p> <p>2. Klicken Sie den Projekt-Dateinamen an, der geöffnet werden soll und klicken Sie dann auf die "Öffnen" Schaltfläche.</p> <p>Projekt speichern</p> <p>Um ein erstelltes oder von Ihnen geändertes Projekt zu speichern, gehen Sie wie folgt vor:</p> <p>1. Wählen Sie den "Datei Projekt speichern" Menü-Eintrag. Oder drücken Sie alternativ die <STRG+S> Tastenkombination.</p> <p>Projekt speichern unter</p> <p>Um ein geöffnetes Projekt unter einem neuen Namen zu speichern, gehen Sie wie folgt vor:</p> <p>1. Wählen Sie den "Datei Projekt speichern unter" Menü-Eintrag.</p> <p>2. Wählen Sie einen Zielordner im erschienenen "Speichern unter"-Fenster und geben Sie den Namen ein unter dem das Projekt gespeichert wird.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Falls eine solche Datei bereits existiert, fragt AxProtector nach, ob Sie diese überschreiben wollen. Klicken Sie die "Nein" Schaltfläche und speichern sie das Projekt unter einem anderen Namen, um die existierende Datei beizubehalten. </div>
Wbc-Datei exportieren	<p>Das Wählen dieses Menü-Eintrags exportiert die Schutzeinstellungen in eine *.wbc-Datei, die Sie frei benennen und abspeichern können. Diese Datei können Sie später in der AxProtector²⁹² -Kommandozeilen²⁹²-Variante verwenden.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Dieser Menü-Eintrag steht Ihnen nur dann zur Verfügung, wenn das Projekt alle notwendigen Überprüfungen bestanden hat. </div>
Beenden	Über den " Datei Beenden " Menü-Eintrag beenden Sie AxProtector. Alternativ schließen Sie das Fenster über das "x" Bedienelement, oder die <ALT+F4> Tastenkombination. Vor dem Verlassen werden Sie aufgefordert, vorgenommene Änderungen abzuspeichern.

Das Optionen-Menü

Element	Beschreibung
Sprache	AxProtector bietet Ihnen verschiedene Sprachversionen für die Bedienoberfläche. Wählen

Element	Beschreibung
	Sie unter 8 Spracheinstellungen: Chinesisch, Deutsch, Englisch, Spanisch, Französisch, Japanisch, Niederländisch, Portugiesisch.

Das ?-Menü

Element	Beschreibung
Inhalt	Das Wählen dieses Menü-Eintrags öffnet die AxProtector Online-Hilfe.
Über	Das Wählen dieses Menü-Eintrags öffnet ein Fenster mit AxProtector Versionsinformationen.

7.1.2 Navigationsfenster

Das Navigationsfenster zeigt Ihnen für jeden Projekttyp die einzelnen Schutzschritte in einer Baumnavigation an. Über die Navigation springen Sie direkt zu jedem Schritt.

7.1.3 Eingabefenster

Das Eingabefenster ermöglicht die Eingabe von Schutzeinstellungen über entsprechende Felder und Dialoge. Sie navigieren durch die einzelnen Schritte über die "Weiter" oder "Zurück" Schaltflächen am Ende jedes Schrittes.

	Dieses Symbol zeigt Ihnen an, dass Sie über die "Erweitert" Schaltfläche zusätzliche Schutzeinstellungen vorgenommen haben.
--	---

7.1.4 Hinweis- und Fehler-Fenster

Dieses Fenster zeigt Ihnen über Symbole Hinweise, Fehler oder Warnungen zu den einzelnen Schritten an. Die Symbole sehen Sie ebenfalls vor jedem Schritt im Navigationsfenster.

Symbol	Beschreibung
	Beim Setzen der Einstellungen ist ein Fehler aufgetreten. Der jeweilige Schutzschritt wird nicht durchgeführt. Ein Text klärt Sie über die mögliche Fehlerursache auf. Sie haben dann die Möglichkeit, nochmals Ihre Eingaben zu überprüfen.
	Beachten Sie eine Warnung im Zusammenhang mit Ihren Einstellungen zum Schutz Ihrer Anwendung.
	Alle Einstellungen sind korrekt vorgenommen. Dieser Schutzschritt wird durchgeführt.

	Mit einem Doppelklick auf die und Symbole springen Sie automatisch zur Seite, auf die sich die Information bezieht.
--	---

7.1.5 Projekttyp-Bereich

Dieser Bereich zeigt Ihnen, welchen Projekttyp Sie gerade verwenden und zeigt vorhandene Tooltip-Texte an, wenn Sie mit der Maus über Elemente in den Dialogen fahren.

7.2 Projekt Dialog

Beim Aufruf von AxProtector und beim Anlegen eines neuen Projektes in AxProtector öffnet sich ein Projekt-Dialog, der Ihnen verschiedene Projekttypen zur Auswahl anbietet.

Über die Karteireiter "**AxProtector**", "**IxProtector**" und "**Sonstige**" bekommen Sie die verfügbaren Projekttypen angezeigt.



Unterstützung bekommen Sie über die Schaltfläche "**Hilfe**".

7.3 Projekttypen

AxProtector kennt die folgenden Projekttypen:

Icon	Projekttyp
	AxProtector
	Windows Anwendung oder DLL <small>76</small>
	.NET Assembly <small>110</small>
	Mac OS X Anwendung oder Dylib <small>140</small>
	Java Anwendung (jar-Datei) <small>168</small>
	Linux Anwendung oder Shared Object <small>190</small>
	IxProtector
	Windows Anwendung oder DLL <small>218</small>

Icon	Projekttyp
	.NET Assembly ²³²
	Linux Anwendung oder Shared Object ²⁶¹
	Mac OS X Anwendung oder Dylib ²⁴⁷
	Andere
	Dateiverschlüsselung ²⁷⁵

7.4 AxProtector Karteireiter

7.4.1 Windows Anwendung oder DLL

AxProtector schützt ausführbare Dateien (Anwendungen *.exe und Bibliotheken *.dll) im PE Format (Portable Executable): Die ausführbaren Dateien können z.B. mit gängigen Compilern (C, C++; Delphi, VB 6.0, FORTRAN, ...) oder Autorenwerkzeugen (AdobeFlash, ...) erstellt werden.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Windows mit AxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Windows Anwendung oder DLL	AxProtector Windows ⁷⁶	✓	Windows Kommandozeile ²⁹²

7.4.1.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

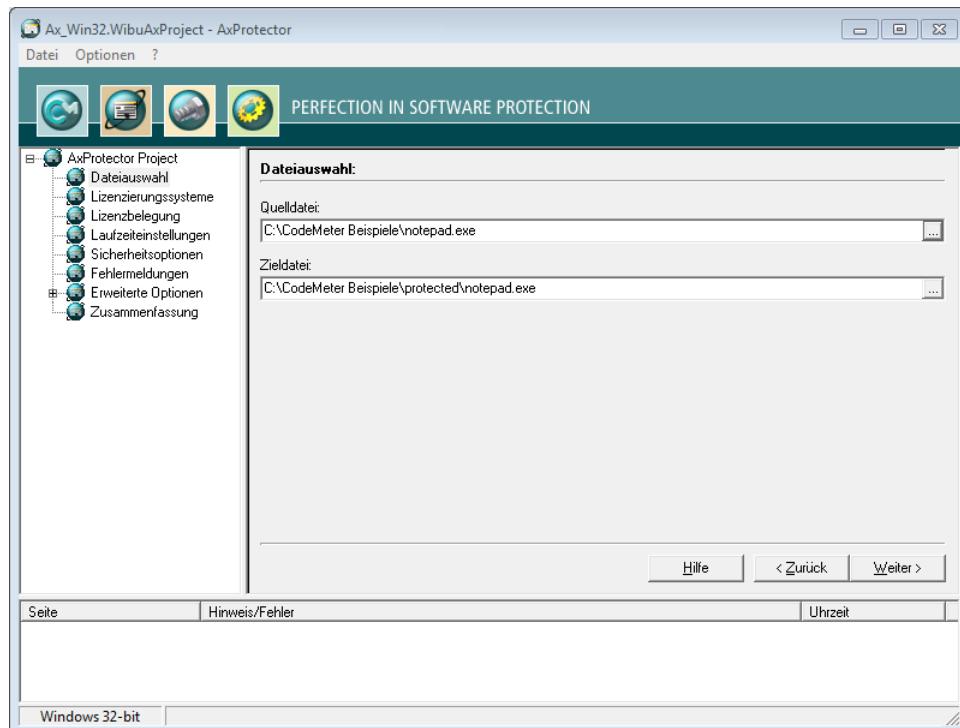


Abbildung 15: AxProtector - Windows "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein. i Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [...] \protected\...]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.4.1.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Datei nehmen Sie hier Einstellungen zum verwendeten Lizenziereingssystem *CmDongle* und/oder *CmActLicense* vor.

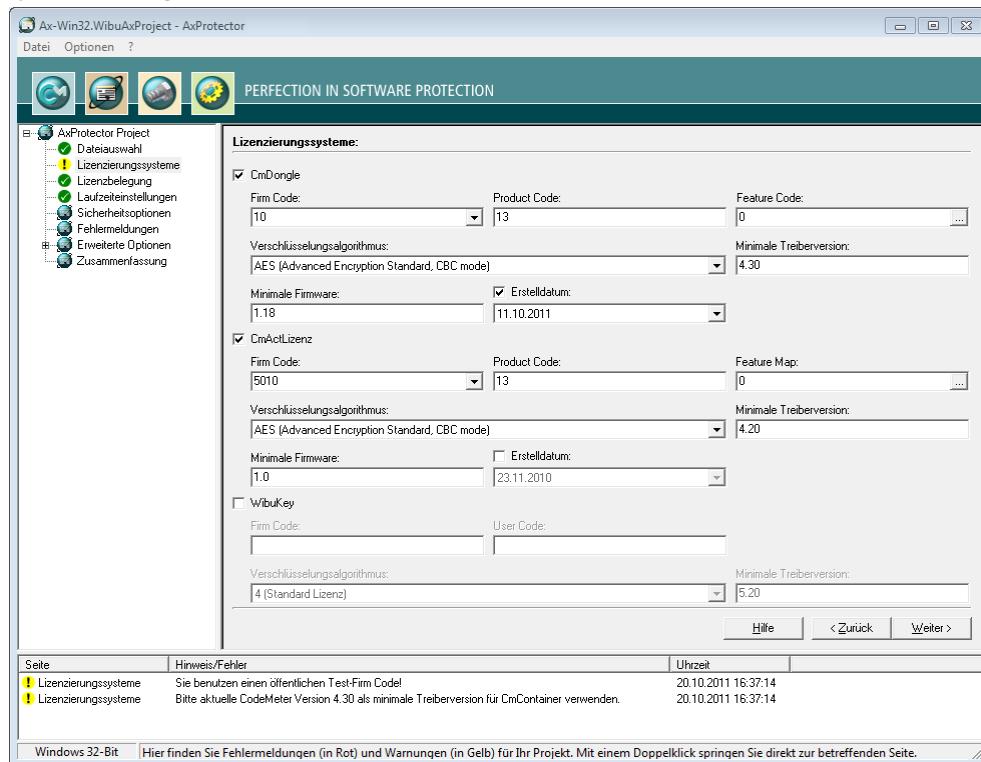


Abbildung 16: AxProtector - Windows "Lizenzierungssysteme"

Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenziierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*-Lizenz.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenziereingssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die Wibu-Systems Interne Seiten.

Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich:

Element	Beschreibung
Firm Code	<p>Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird.</p> <p> Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle Evaluation-Firm Code</i> des <i>CodeMeter® Software Development Kits (SDK)</i> und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010. Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bewirkt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eintragen.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
<p>Abbildung 17: AxProtector - .NET Feature Map Eingabe</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>	
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. <i>CodeMeter®</i> unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Minimale Treiberversion	<p>Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i> an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt AxProtector automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizenzen.</p> <p> Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen</p>

Element	Beschreibung
	<p>beim Starten Ihrer geschützten Software.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden.</p> <p>Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <p>Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können.</p> <p>Beachten Sie auch, dass Sie hier das Kontrollkästchen aktivieren müssen, um Überprüfungsoptionen des Wartungszeitraumes (Maintenance Period) im Dialog zu den erweiterten Laufzeiteinstellungen¹²¹ vornehmen zu können.</p>
Minimale Firmware	<p>Kommandozeilen-Option siehe hier²⁹⁴.</p> <p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem WibuKey informiert separat das WibuKey Entwicklerhandbuch.

7.4.1.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Anwendung vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

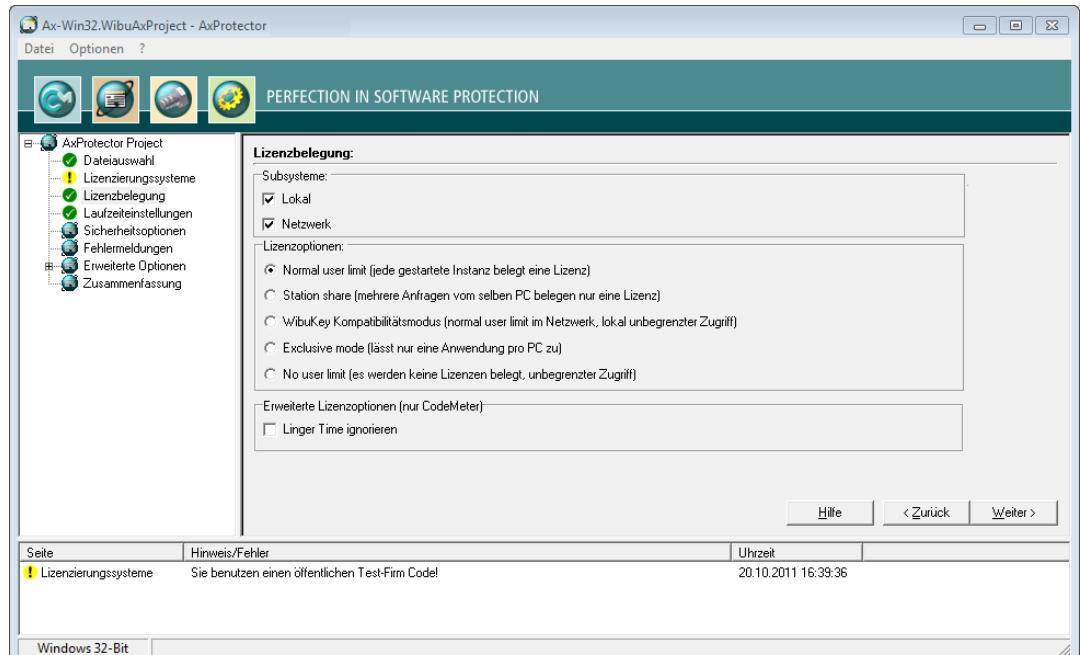


Abbildung 18: AxProtector - Windows "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#)²⁹⁴).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der CodeMeter Lizenzserver mit einem aktvierten Netzwerkzugriff läuft.



Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizenzoptionen

Im Bereich Lizenzoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier](#)²⁹⁵).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz.  Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.
WibuKey Kompatibilitäts-Modus	Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).  Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu <i>WibuKey</i> . <i>Wibu Systems</i> empfiehlt die Einstellungen 'Normal user limit' und 'Station Share'.
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizenzen belegt werden. Belegte Lizenzen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzeigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).

7.4.1.4 Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie das Verhalten der Anwendung zur Laufzeit fest, z.B. Abfrage der Lizenz in CmContainern, Ausgabe von Warnmeldung, etc.

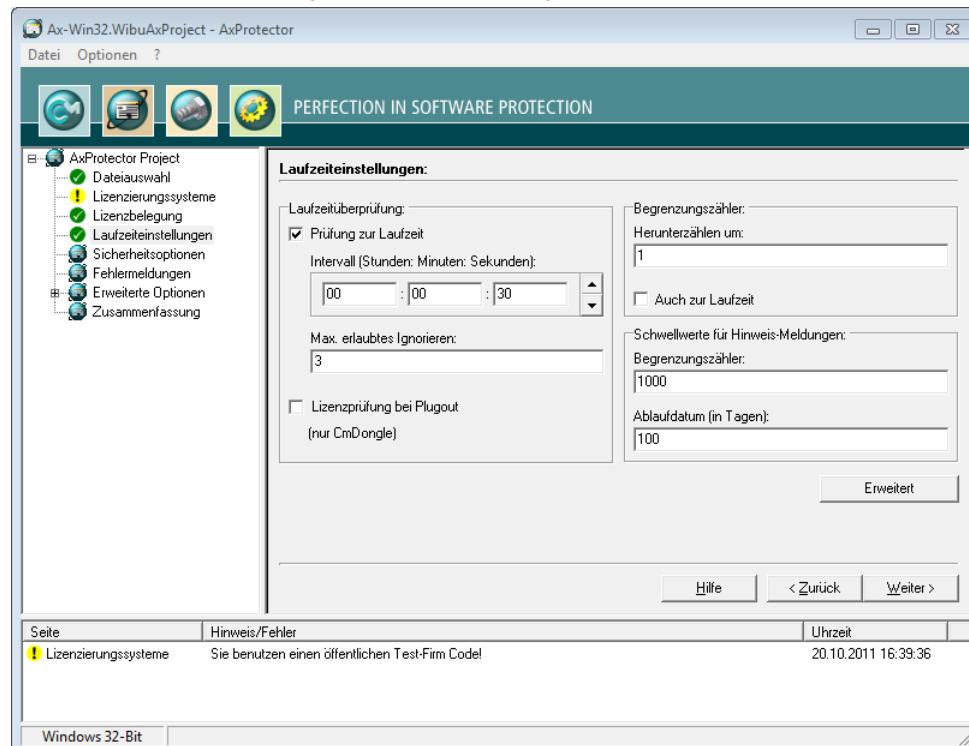


Abbildung 19: AxProtector - Windows "Laufzeiteinstellungen"

Laufzeitüberprüfung

In diesem Bereich können Sie definieren, ob und wie oft die geschützte Anwendung die Lizenz während der Laufzeit überprüft..

Elemente	Beschreibung
Prüfung zur Laufzeit	Aktiviert oder deaktiviert die Überprüfung während der Laufzeit der geschützten Anwendung. Kommandozeilen-Option siehe hier .
Intervall	Legt das Intervall zwischen zwei Überprüfungen fest. Angabe im Format Stunden: Minuten: Sekunden.
Max. erlaubtes Ignorieren	Gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann. Schlägt die Verbindung zum CmContainer fehl, d.h. kann nicht mehr auf die Lizenz zugriffen werden, geben Sie dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit, auch ohne die Lizenz noch weiterzuarbeiten.

Elemente	Beschreibung
Lizenzprüfung bei Plug-Out (nur CmDongle)	Beendet die geschützte Anwendung, wenn der <i>CmDongle</i> während der Ausführung abgezogen wird und eine sofortige Fehlermeldung wird ausgegeben. Kommandozeilen-Option siehe hier ²⁹⁷ .

Begrenzungszähler

Begrenzungszähler (Unit Counter) können u.a. dazu dienen, die Gültigkeit von Lizzenzen in einem *CmContainer* festzustellen. In diesem Bereich können Sie dieses Verhalten definieren (Kommandozeilen-Option siehe [hier](#)²⁹⁸).

Element	Beschreibung
Herunterzählen um	Gibt den Wert an, um den der Begrenzungszähler (Unit Counter) heruntergezählt wird. Diese Option bewirkt das Herunterzählen des Zählers beim Start der geschützten Anwendung. Ist die "Auch zur Laufzeit" Option aktiviert und sind die Einträge wie in der obigen Abbildung dargestellt gesetzt, wird alle 30 Sekunden (siehe das festgelegt Intervall) ein gesetzter Begrenzungszähler (Unit Counter) um den Wert 1 heruntergezählt.
Auch zur Laufzeit	Zählt den Begrenzungszähler (Unit Counter) auch während der Laufzeit der geschützten Anwendung herunter.



Diese Option greift nur, wenn die "Prüfung zu Laufzeit" Option im Bereich "[Laufzeit-überprüfung](#)"²⁸³ aktiviert ist.

Schwellenwerte für Hinweismeldungen

In diesem Bereich können Sie definieren, wann eine Hinweismeldung zur Gültigkeit der Lizenz ausgegeben wird.

	Zur individuellen Gestaltung des Textes siehe Hinweismeldungen ⁹² .
--	--

Element	Beschreibung
Begrenzungszähler	Wird der angegebene Schwellenwert unterschritten, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .
Ablaufdatum (in Tagen)	Wird das angegebene Ablaufdatum in Tagen innerhalb der vorgegebenen Schwelle erreicht, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .

7.4.1.4.1 Erweiterte Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie zusätzliche Einstellungen zur Laufzeit der verschlüsselten Anwendung fest.

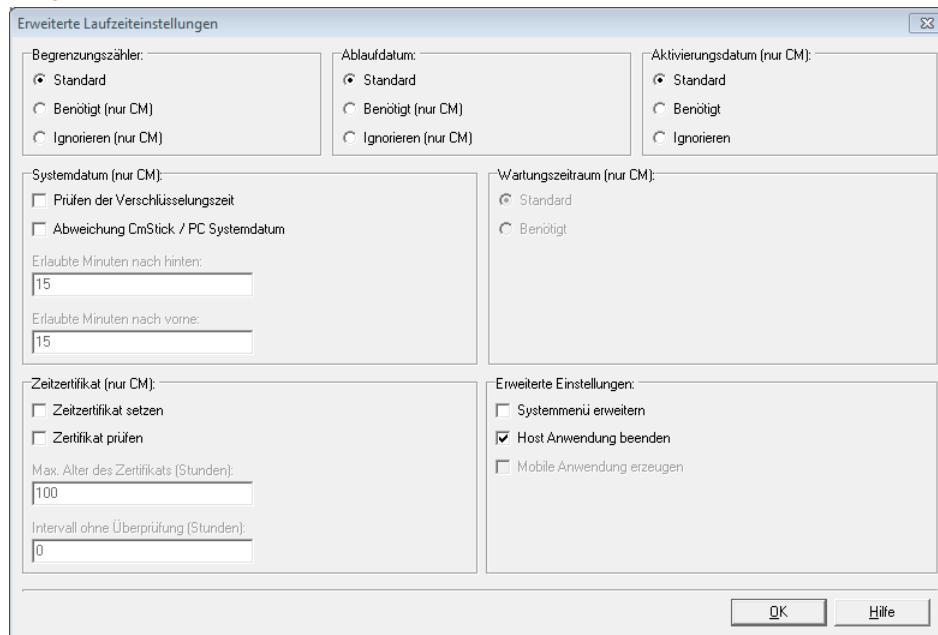


Abbildung 20: AxProtector - Windows "Erweiterte Laufzeiteinstellungen"

Für die Abfrage der in die Lizenz eingetragenen Optionen Begrenzungszähler (Unit Counter), Ablaufdatum (Expiration Time) und Aktivierungsdatum (Activation Time) gilt die folgende Handhabung.

Status	Standard	Benötigt	Ignorieren
= 0	X	X	✓
< > 0	✓	✓	✓
nicht angegeben	✓	✓	✓

Begrenzungszähler (Unit Counter)

Definiert die Handhabung eines Unit Counter (Begrenzungszählers), der in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)^{▷ 308}).

Element	Beschreibung
Standard	Zählt einen vorhandenen Unit Counter-Eintrag in der Lizenz beim Start und/oder zur Laufzeit um den auf der vorherigen Seite definierten Wert herunter. Wenn der Unit Counter Null erreicht startet die verschlüsselte Anwendung nicht.
Benötigt	Ein Unit Counter-Eintrag < > 0 in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag star-

Element	Beschreibung
	tet die verschlüsselte Anwendung nicht.
Ignorieren	Ein vorhandener Unit Counter-Eintrag in der Lizenz wird ignoriert. Die Anwendung setzt den Unit Counter nicht herunter. Die Anwendung startet auch bei einem Unit Counter-Eintrag = 0.

Ablaufdatum (Expiration Time)

Definiert die Handhabung einer Expiration Time (Ablaufdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Expiration Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn keine Expiration Time vorhanden ist, oder das aktuelle Datum vor der Expiration Time liegt.
Benötigt	Ein Expiration Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten.
Ignorieren	Ein vorhandener Expiration Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum nach der Expiration Time liegt.

Aktivierungsdatum (Activation Time)

Definiert die Handhabung einer Activation Time (Aktivierungsdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Activation Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn kein solcher Eintrag vorhanden ist, oder die zertifizierte Zeit ⁴¹⁷ nach Activation Time liegt.
Benötigt	Ein Activation Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten. Beachten Sie, dass dann eine Internet-Verbindung zum Abholen der zertifizierten Zeit erforderlich ist.
Ignorieren	Ein vorhandener Activation Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum vor der Activation Time liegt.

Wartungszeitraum (Maintenance Period)

Definiert die Handhabung eines Wartungszeitraumes (Maintenance Period), der in der Lizenz eingetragen ist. Eine Lizenz berechtigt dann zur Verwendung aller Softwareversionen, die innerhalb des definierten Wartungszeitraumes (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der Applikation hinterlegt und zur Laufzeit der geschützten Anwendung geprüft, ob das Erstelldatum (Release Date) innerhalb des Wartungszeitraumes (Maintenance Period) liegt (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

	Die Optionen sind nur auswählbar, wenn auf der Seite "Lizenzerierungssysteme" das Erstelldatum (Release Date) aktiviert ⁸⁰ worden ist.
--	---

Es bestehen zwei Überprüfungsoptionen:

Element	Beschreibung
Standard	Während der Laufzeit der geschützten Anwendung wird gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist. Dies entspricht der Standardeinstellung auch wenn auf der Seite "Lizenzerierungssysteme" das Erstelldatum (Release Date) nicht aktiviert ⁸⁰ wor-

Element	Beschreibung
	den ist.
Benötigt	Während der Laufzeit der geschützten Anwendung ist das Prüfen des Wartungszeitraumes (Maintenance Period) gegen das Erstelldatum (Release Date) zwingend erforderlich. Die PIO Wartungszeitraum (Maintenance Period) muss vorhanden sein.

Zeitzertifikat

In jedem *CmContainer* ist eine laufende Uhr integriert, die läuft, wenn der *CmContainer* mit dem Rechner verbunden ist. Die Uhrzeit synchronisiert sich dabei beim Aktivieren des *CmContainers* nach vorne und nutzt ansonsten die letzte gespeicherte Zeit.

Wenn gewünscht, kann die zertifizierte Uhrzeit durch die Synchronisation mit dem *CodeMeter®* Zeitserver aktualisiert werden. Die Zeitserver sind von Wibu-Systems bereitgestellte Rechner, die über die Welt verteilt sind und eine zertifizierte Zeit zur Verfügung stellen. Bei einer Aktualisierung der zertifizierten Uhrzeit wird die interne *CmContainer*-Zeit synchronisiert (Kommandozeilen-Option siehe [hier](#)⁴¹⁷).

 Für Informationen zur Manipulationssicherheit von Aktivierungs- und Ablaufdatum siehe hier ⁴¹⁷ .

Element	Beschreibung
Zeitzertifikat setzen	Mit dieser Option wird versucht die zertifizierte Zeit im <i>CmContainer</i> zu aktualisieren. Die zertifizierte Zeit wird beim Zeitserver angefordert.  Diese Option erfordert eine Internet-Verbindung.
Zertifikat prüfen	Diese Option überprüft, ob die zertifizierte Zeit älter ist, als das hier festlegbare maximale Alter. Ist das maximale Alter des Zeitzertifikats überschritten, so lässt sich die Anwendung nicht starten.
Max. Alter des Zertifikats (in Stunden)	Bei ausgewählter "Prüfung" des Zeitzertifikats können Sie hier das maximale Alter des Zertifikats in Stunden angeben. Das Alter des Zertifikates berechnet sich aus der Differenz der laufenden System-Zeit und der zertifizierten Zeit.
Intervall ohne Überprüfung (Stunden)	Gibt an, innerhalb welchen Intervalls <u>keine</u> Überprüfung des Zeitzertifikats stattfindet. Ist dieses Intervall noch nicht erreicht, findet keine Überprüfung statt. Befindet sich das Zeitzertifikat zwischen diesem Intervall und dem max. Alter des Zertifikats, wird versucht, das Zeitzertifikat zu aktualisieren. Gelingt dies nicht, läuft die Anwendung jedoch bis zum Erreichen des max. Alters des Zeitzertifikats weiter. Erst danach ist zwingend ein aktualisiertes Zeitzertifikat notwendig.

System Datum

In diesem Bereich nehmen Sie Einstellungen vor, die dem zusätzlichen Schutz dienen, eine Lizenz über ein bewusstes Falschstellen der PC-Zeit zu manipulieren (Kommandozeilen-Option siehe [hier](#)⁴²⁹).

Element	Beschreibung
Prüfen der Verschlüsselungszeit	Diese Option speichert die Verschlüsselungszeit (PC Time) in der geschützten Anwendung. Die Anwendung läuft auf dem Kunden-PC dann nur, wenn die <i>CmContainer</i> Systemzeit neuer ist als die Verschlüsselungszeit.

Element	Beschreibung
	 Erfordert mindestens <i>CodeMeter® 4.10</i> .
Abweichung CmContainer / PC Systemzeit	Wird diese Option aktiviert, ist die Festlegung eines Zeitkorridors möglich, innerhalb dessen sich die Abweichung zwischen <i>CmContainer</i> Systemzeit und der PC-Zeit bewegen darf. Wird dieser unter- bzw. überschritten läuft die geschützte Anwendung auf dem Kunden-PC nicht.
Erlaubte Minuten nach hinten	Gibt in Minuten an, um wieviele Minuten die PC Zeit älter als die <i>CmContainer</i> Systemzeit sein darf.
Erlaubte Minuten nach vorne	Gibt in Minuten an, um wieviele Minuten die PC Zeit vor der <i>CmContainer</i> Systemzeit liegen darf.

Erweiterte Optionen

Dieser Bereich lässt die Auswahl weiterer Optionen zu.

Element	Beschreibung
Systemmenü erweitern	Fügt Ihrer Anwendung die Menüeinträge "Über" und "Systemmenü" hinzu (Kommandozeilen-Option siehe hier ³⁰⁰).
Host Anwendung beenden	Wenn keine gültige Lizenz gefunden wird, wird im Falle geschützter Dll-Anwendungsdateien die aufrufende *.exe beendet (Kommandozeilen-Option siehe hier ³¹⁰).
Mobile Anwendung erzeugen	[noch nicht implementiert]

7.4.1.5 Sicherheitsoptionen

Über diese Seite treffen Sie eine Auswahl aus verschiedenen Schutzmechanismen und -methoden für Ihre Anwendung. Sie können hier den Grad der Sicherheit selbst skalieren. Dabei legen Sie z.B. selbst fest, wie intensiv nach Debuggern gesucht werden soll, bis hin zum Sperren des *CmContainers*.

 Sollten sich die gesetzten Optionen inkompatibel zu Ihrer geschützten Anwendung verhalten, so können die einzelnen Sicherheitsoptionen auch einzeln deaktiviert werden.

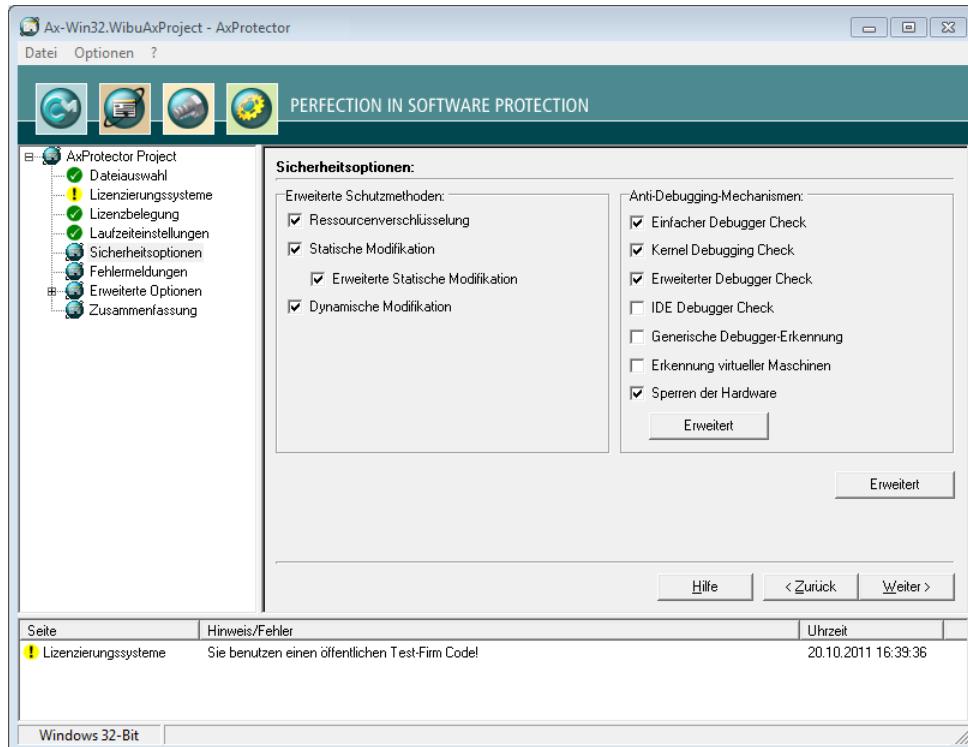


Abbildung 21: AxProtector - Windows "Sicherheitsoptionen"

Erweiterte Schutzmethoden

Die erweiterten Schutzmethoden greifen tief in die Anwendung ein. Dies kann in manchen Fällen dazu führen, dass einige Mechanismen aus Kompatibilitätsgründen nicht funktionieren (Kommandozeilen-Option siehe [hier](#)^{D₂₀}).

Element	Beschreibung
Ressourcen-Verschlüsselung	Verschlüsselt auch die Ressourcen Ihrer geschützten Anwendung. Die Ressourcen liegen nach Start Ihrer Anwendung verschlüsselt im Speicher und werden erst "on demand" entschlüsselt.
Statische Modifikation	Modifiziert Ihre Software in einer Weise, die sie gegen Debugging, Dumps und Reverse Engineering schützt. Diese Modifikationen werden <u>bei</u> der Verschlüsselung zu ihrer Anwendung hinzugefügt.
Erweiterte Statische Modifikation	Fügt der statischen Modifikation bei der Verschlüsselung Ihrer Anwendung zusätzlich mehrfach verschachtelte Sicherheitsmechanismen hinzu.
Dynamische Modifikation	Der Programmcode der zu schützenden Anwendung wird <u>zur Laufzeit</u> dynamisch modifiziert.

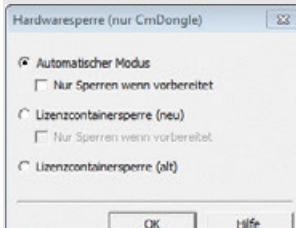


Die Optionen "Statische Modifikation" und "Erweiterte Statische Modifikation" greifen nicht, wenn auf der Seite "Erweiterte Optionen" die Option "[Dateiverschlüsselung aktivieren](#)"⁹⁶ aktiviert ist.

Anti-Debugging Mechanismen

Debugger-Programme dienen der Fehlersuche und Fehlerbeseitigung, können aber auch von Hackern zur Analyse der Software verwendet werden. In diesem Bereich legen Sie die Optionen fest, wie auf Debugger-Programme reagiert werden soll (Kommandozeilen-Option siehe [hier](#)²⁹⁷).

Element	Beschreibung						
Einfacher Debugger Check	Überprüft ob ein Debugger an Ihre Anwendung angehängt (attached) ist. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.						
Kernel Debugging Check	Überprüft zusätzlich auf Kernel-Debugger-Programme, wie z.B. "SoftICE". Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet. Die beiden nächsten Mechanismen umfassen Methoden zur Erkennung von spezifischen Debugger-Programmen und Werkzeugen.						
Erweiterter Debugger Check	Überprüft in einer erweiterten Suche auf Debugger-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.						
IDE Debugger Check	Überprüft auf sämtliche Debugger-Programme. Dabei sind keine Debugger-Programme mehr erlaubt, d.h. auch keine innerhalb von Entwicklungsumgebungen (z.B. Visual Studio, Delphi). Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.						
Generische Debugger	Fügt der Anwendung einen Mechanismus hinzu, der das Anhängen eines Debuggers an die laufende Anwendung verhindert.						
Virtuelle Maschinen	Erkennt, ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies.						
Sperren des Lizenz-Zugriffs	Mit dieser Option kann das genutzte Firm Item im <i>CmContainer</i> gesperrt werden sobald ein Debugger-Programm entdeckt wird. Wird die Option aktiviert, werden die Einstellungen übernommen, die Sie in einem Dialog setzen, der sich über die " Konfiguration "-Schaltfläche öffnet.						
<div style="border: 1px solid #ccc; padding: 5px; text-align: center;"> Diese Schaltfläche ist nur aktiviert für <i>CodeMeter</i>. </div>							
Konfiguration	<p>Wenn die Option "Sperren des Lizenz-Zugriffs" aktiviert wird, können Sie über die "Erweitert"-Schaltfläche im folgenden Dialog weitere Einstellungen vornehmen: Der Dialog erlaubt in Abhängigkeit von der Firmware, die bei der Verschlüsselung verwendet wird, die Auswahl verschiedener Sperrszenarien.</p> <table border="1"> <thead> <tr> <th>Sperrszenario</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>sofortiges Sperren</td><td>erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.</td></tr> <tr> <td>vorbereitetes Sperren</td><td>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte</td></tr> </tbody> </table>	Sperrszenario	Beschreibung	sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.	vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte
Sperrszenario	Beschreibung						
sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.						
vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte						

Element	Beschreibung
	<p>Sperrszenario</p> <p>programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</p> 
	<p>Abbildung 22: AxProtector - Windows "Sicherheitsoptionen - Hardwaresperre"</p> <p>Die folgenden Einstellungen sind verfügbar</p>
Einstellung	Beschreibung
"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)	<p>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt. Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items. Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</p>
"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert	<p>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft. Ist die Firmware 1.14 und höher, dann wird gleichzeitig geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.</p>
"Lizenzcontainersperre (neu)" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert	<p>Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Dies ist sicherheitstechnisch gesehen die empfohlene Einstellung. Voraussetzung ist jedoch, dass alle CmContainer im Feld mit einer Firmware Version 1.14 und höher ausgestattet sind.</p>
"Lizenzcontainersperre (neu)" markiert und das Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert	<p>Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Gleichzeitig wird geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.</p>

Element	Beschreibung	
	Einstellung	
	Option "Lizenzcontainersperre (alt)" markiert	Gilt für alle Firmware-Versionen. Ist ein vorbereitetes Sperren programmiert, wird der FAC um den Wert 1 heruntergezählt.

7.4.1.5.1 Erweiterte Sicherheitsoptionen

Ermöglicht die Auswahl zusätzlicher Sicherheitseinstellungen.



Abbildung 23: AxProtector - Windows "Erweiterte Sicherheitsoptionen"

Erweiterte Einstellungen

Dieser Bereich lässt die Auswahl weitere Optionen zu.

Element	Beschreibung
Virusprüfung hinzufügen	Der geschützten Anwendung wird eine Virenprüfung über eine Prüfsumme hinzugefügt (Kommandozeilen-Option siehe hier ³⁰¹).
API statisch linken	Das <i>CodeMeter Kern-API</i> wird statisch zur geschützten Anwendung hinzugelinkt. Diese Option erhöht die Sicherheit, sie vergrößert jedoch auch die ausführbare Datei (Kommandozeilen-Option siehe hier ³⁰²).
Zu verschlüsselnder Code (in %)	Hier kann die Menge des zu verschlüsselnden Codes (in %) angegeben werden (Kommandozeilen-Option siehe hier ³⁰⁰).

7.4.1.6 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisenfenster angezeigt werden sollen.

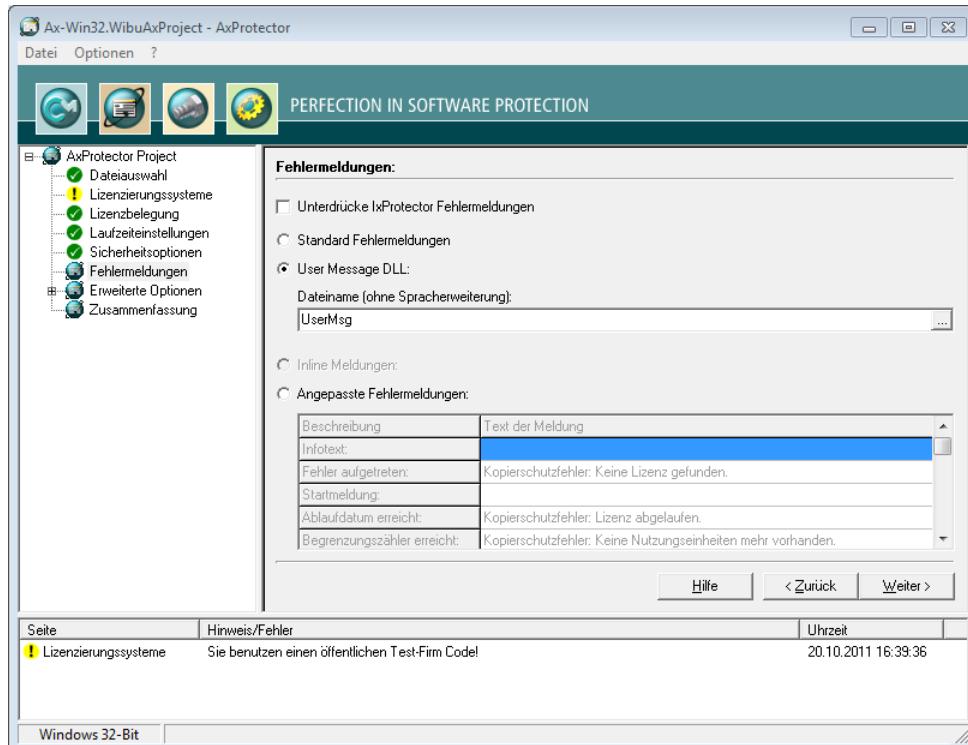
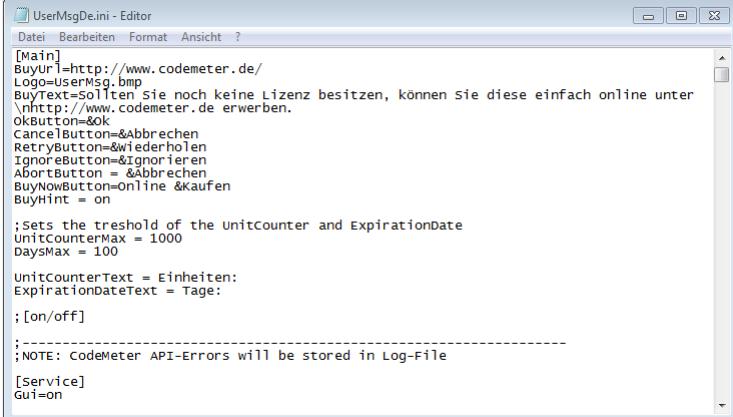


Abbildung 24: AxProtector - Windows "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Unterdrücke IxProtector Fehlermeldungen	Unterdrückt die Ausgabe von IxProtector Fehlermeldungen (Kommandozeilen-Option siehe hier ³⁰⁴).  Setzen Sie diese Option nicht, so werden bei der Verwendung von IxProtector im Fehlerfall zusätzliche Meldungsfenster angezeigt, und zwar zusätzlich zu den im Projekt selbst ausprogrammierten Meldungen.
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlermeldungen können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen (Kommandozeilen-Option siehe hier ³¹²).  Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector

Element	Beschreibung
	<p>geschützte Anwendung befindet.</p>  <pre> [UserMsgDe.ini - Editor] Datei Bearbeiten Format Ansicht ? [Main] BuyUr = http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten Sie noch keine Lizenz besitzen, können Sie diese einfach online unter \nhttp://www.codemeter.de erwerben. OkButton=&OK CancelButton=&Abbrechen RetryButton=&Wiederholen IgnoreButton=&Ignorieren AbortButton = &Abbrechen BuyNowButton=Online &Kaufen BuyHint = on ;Sets the threshold of the UnitCounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 UnitCounterText = Einheiten: ExpirationDateText = Tage: ; [on/off] ;-----[NOTE: CodeMeter API-Errors will be stored in Log-File] [Service] Gui=on </pre>
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.4.1.7 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung mit IxProtector und für die Dateiverschlüsselung vorzunehmen.

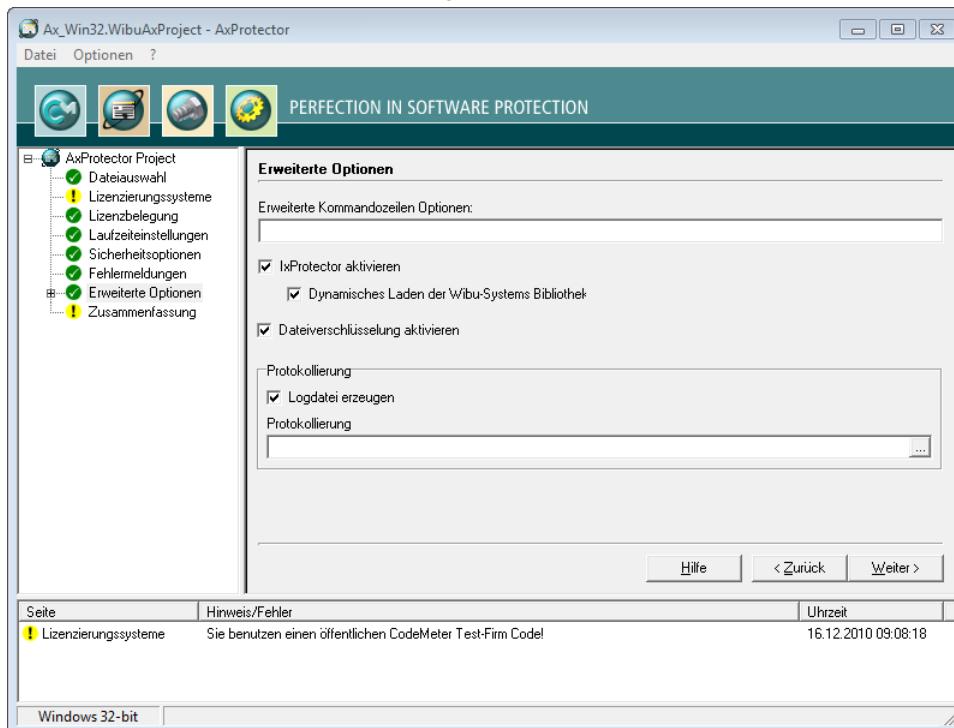


Abbildung 26: AxProtector - Windows Erweiterte Optionen

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen.  Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
IxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das Softwareschutz-API (WUPI) verwenden (Kommandozeilen-Option siehe hier).
Dynamisches Laden der Wibu-Systems Bibliotheken	Das Aktivieren des Auswahlkästchens bewirkt, dass für VB6-Anwendungen oder beim dynamischen Laden von Wibu-Systems Bibliotheken ein besonderes, laufzeitintensives Verfahren eingesetzt wird (Kommandozeilen-Option siehe hier)

Element	Beschreibung
Dateiverschlüsselung aktivieren	Das Aktivieren des Auswahlkästchens veranlasst das automatische Entschlüsseln von Daten-Dateien durch die AxProtector-Engine. Diese Option muss gesetzt werden, wenn Ihre verschlüsselte Anwendung auf die verschlüsselten Daten-Dateien zugreifen können soll (Kommandozeilen-Option siehe hier ²⁹⁷).
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.  Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.

7.4.1.7.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *AxProtector* über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

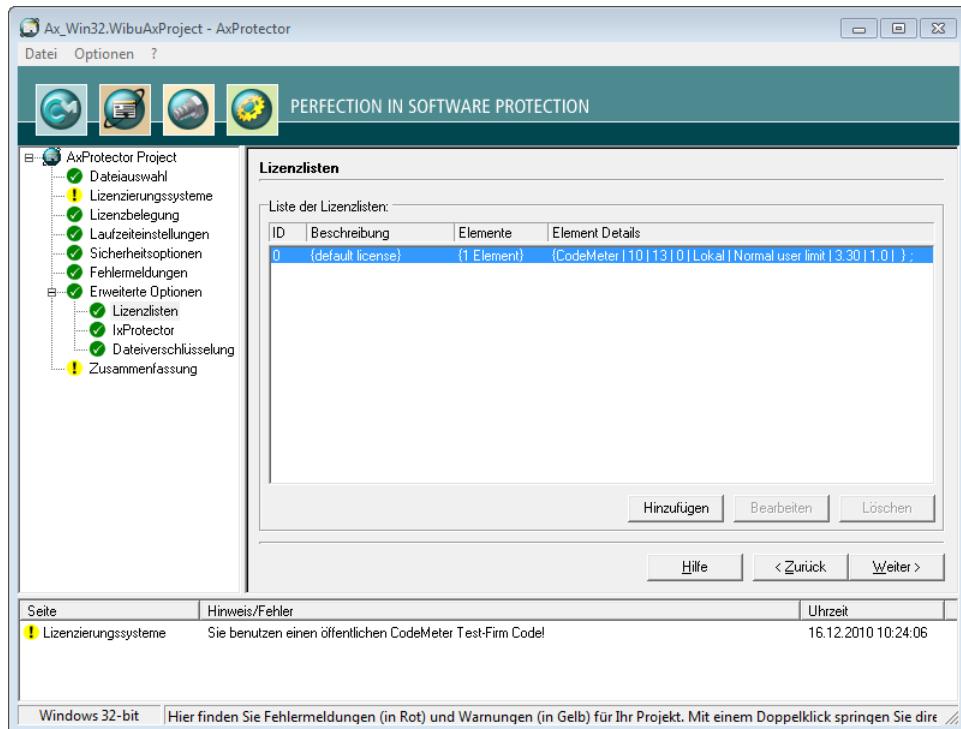
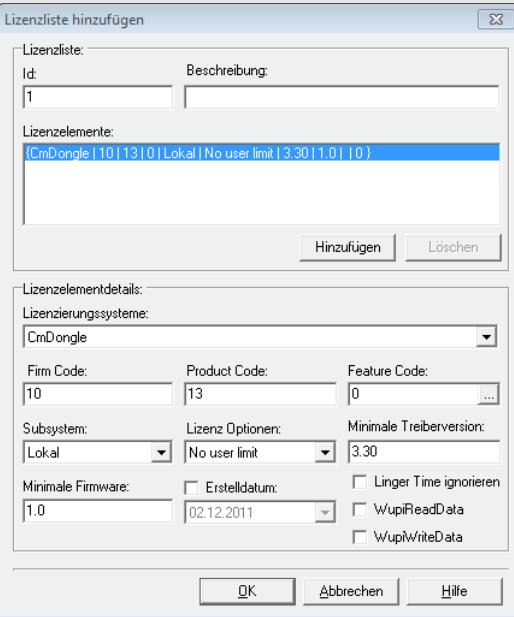


Abbildung 27: AxProtector - Windows Lizenzlisten

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.  Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.

Element	Beschreibung
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag.</p> <p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
Lizenzierungs-systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (<i>CmDongle</i> , <i>CmActLicense</i> oder <i>WibuKey</i>).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt. Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich. 
Subsystem	Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal). Lizenz Optionen Auswahl der Lizenz Optionen zur Belegung von Lizenzen: <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen. Zum Hinterlegen des Erstelldatum (Release Date) gehen Sie wie folgt vor: <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzeigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

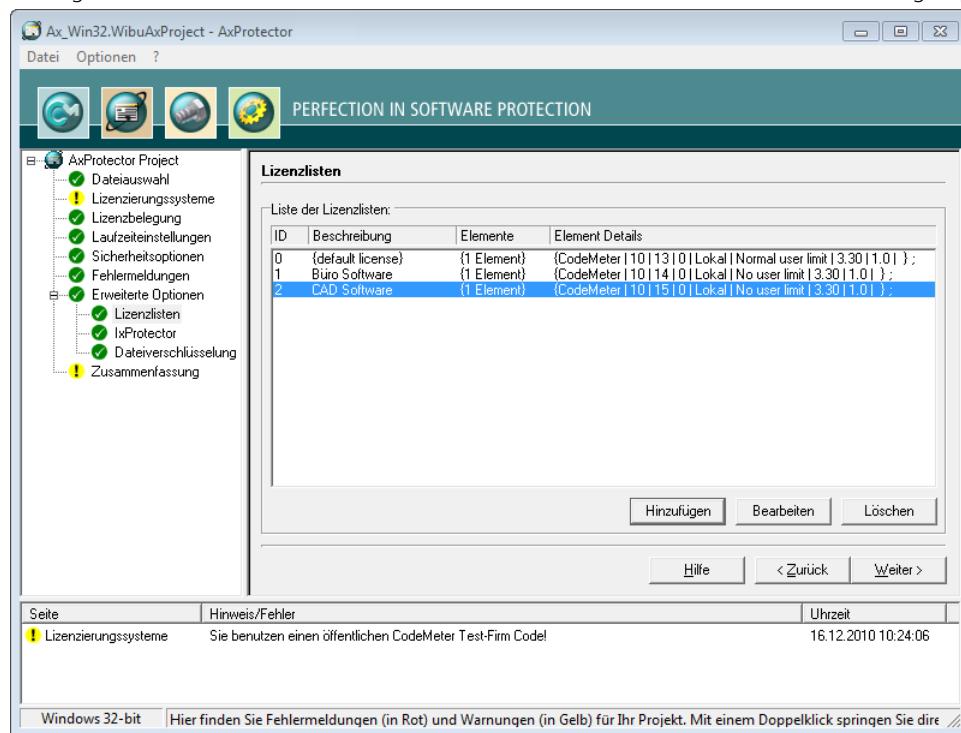


Abbildung 29: *AxProtector - Windows* aus gefüllte Lizenzliste

7.4.1.7.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

 In diesem Fall werden *CodeMeter®* und *WibuKey* API-Aufrufe über die dynamische Bibliothek (*.d11) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

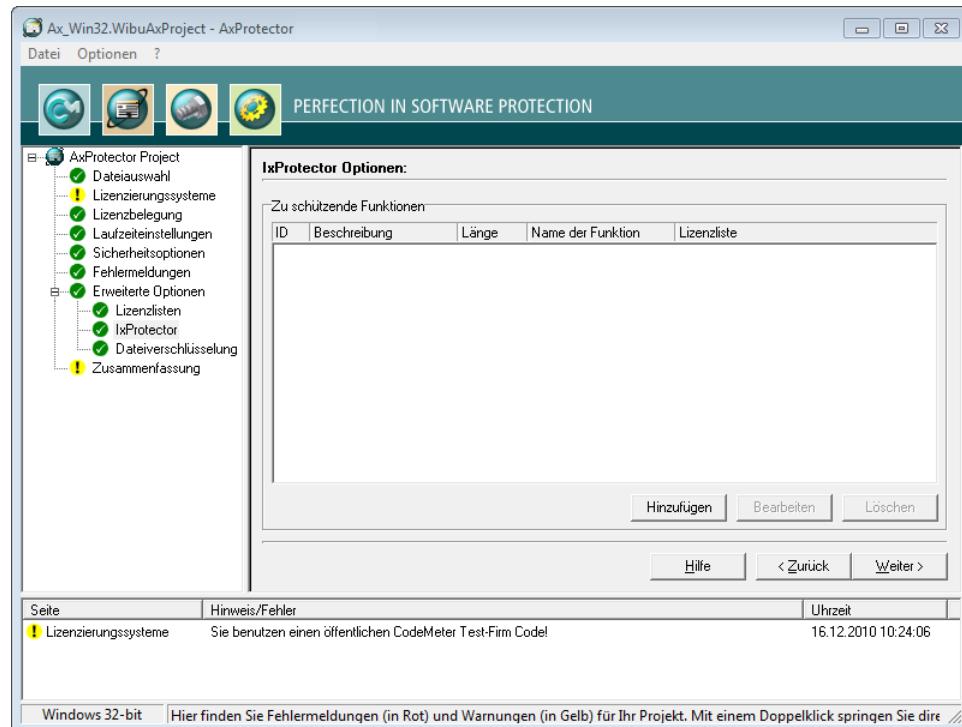
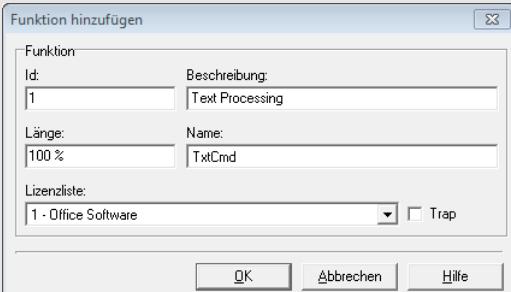


Abbildung 30: AxProtector - Windows IxProtector – Funktionsliste

Element	Beschreibung
Zu schützende Funktionen	Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor: 1. Betätigen Sie im Bereich IxProtector Optionen die "Hinzufügen" Schaltfläche. 2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.

Element	Beschreibung														
	 <p>Abbildung 31: AxProtector - Windows IxProtector – Funktion hinzufügen</p> <table border="1"> <thead> <tr> <th>Element</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>Id</td><td> <p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode und WupiEncryptCode verwenden.</p> </td></tr> <tr> <td>Beschreibung</td><td>Beschreibt die Funktion durch einen Texteintrag.</td></tr> <tr> <td>Länge</td><td> <p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an.</p> <p>Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p> </td></tr> <tr> <td>Name</td><td> <p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p> </td></tr> <tr> <td>Lizenzliste</td><td>Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.</td></tr> <tr> <td>Trap</td><td>Aktiviert die Trap-Funktion für die Funktion.</td></tr> </tbody> </table>	Element	Beschreibung	Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode und WupiEncryptCode verwenden.</p>	Beschreibung	Beschreibt die Funktion durch einen Texteintrag.	Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an.</p> <p>Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>	Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p>	Lizenzliste	Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.	Trap	Aktiviert die Trap-Funktion für die Funktion.
Element	Beschreibung														
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode und WupiEncryptCode verwenden.</p>														
Beschreibung	Beschreibt die Funktion durch einen Texteintrag.														
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an.</p> <p>Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>														
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p>														
Lizenzliste	Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.														
Trap	Aktiviert die Trap-Funktion für die Funktion.														

Element	Beschreibung
	Kommandozeilen-Option siehe hier ³⁰⁸ .
	3. Betätigen Sie die "OK" Schaltfläche. Die neuen Funktionen werden der Funktionsliste hinzugefügt.

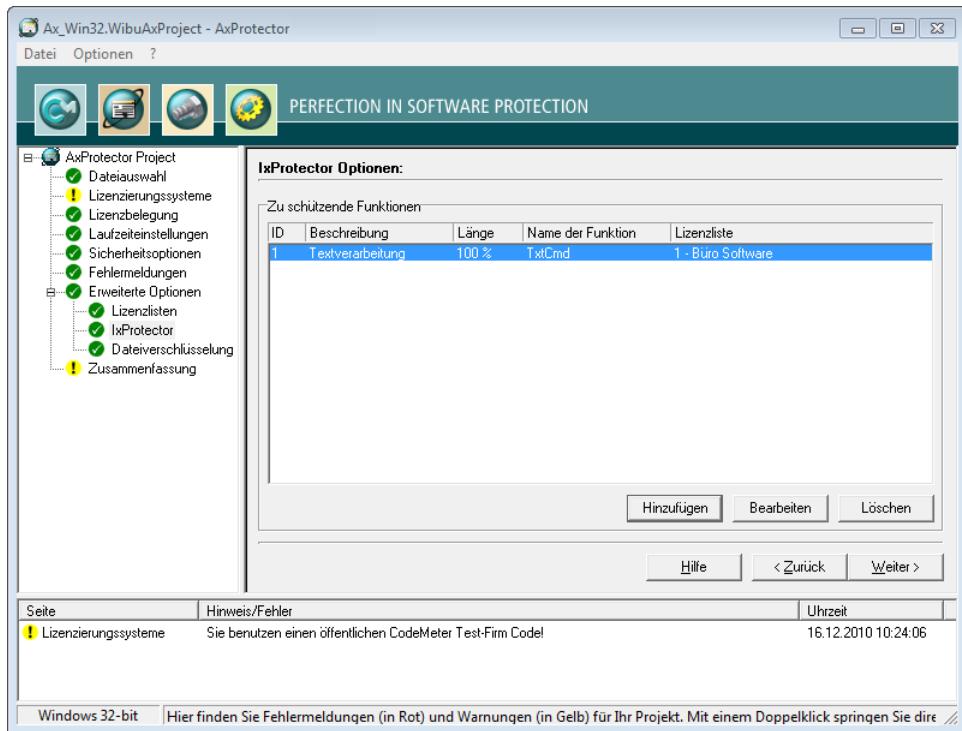


Abbildung 32: AxProtector - Windows IxProtector – Funktionsliste gefüllt

7.4.1.7.3 Dateiverschlüsselung

Über diesen Menü-Eintrag legen Sie fest, nach welchen Regeln eine Anwendung auf verschlüsselte Dateien zugreift. Außerdem haben Sie Möglichkeit dies für unterschiedliche Dateitypen in einer Liste zu definieren. Es können beliebig viele Dateitypen hinzugefügt werden. Für eine Datendatei wird lediglich ein Dateityp benötigt.

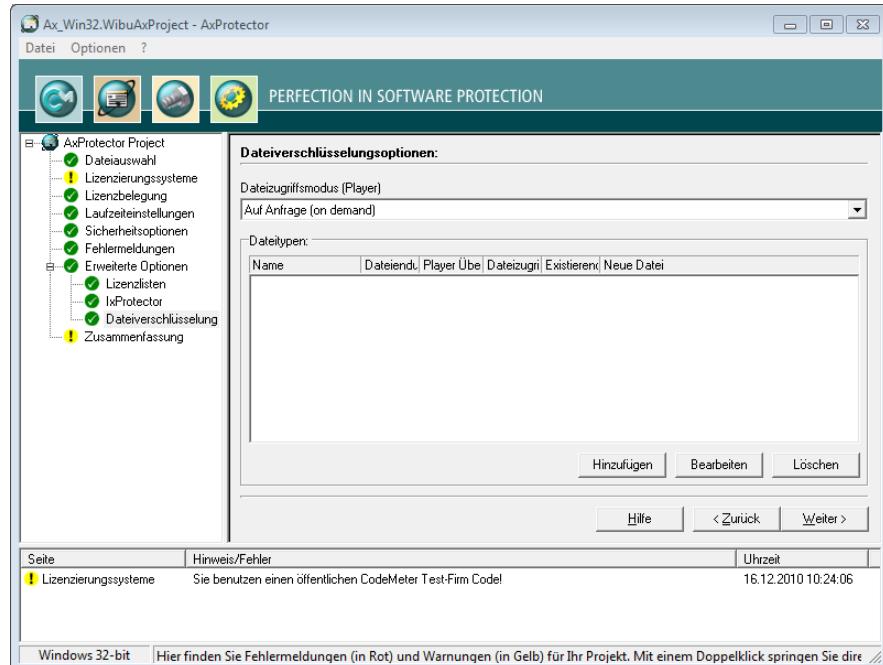


Abbildung 33: AxProtector - Windows Dateiverschlüsselung

Element	Beschreibung												
Dateityp hinzufügen	<p>1. Klicken Sie auf die "Hinzufügen"-Schaltfläche, um der Liste einen neuen Dateityp hinzuzufügen.</p> <div style="border: 1px solid #ccc; padding: 10px;"> <p>Dateityp hinzufügen</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Name</td> <td style="width: 50%;">Dateiendung</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>Player Überprüfung</td> <td>Dateizugriffsmodus</td> </tr> <tr> <td><input type="button" value="0 - (default license)"/></td> <td><input type="button" value="Auf Anfrage (on demand)"/></td> </tr> <tr> <td colspan="2">Schreiboptionen:</td> </tr> <tr> <td>Existierende Datei</td> <td><input type="button" value="Neue Datei"/></td> </tr> </table> <p style="text-align: center;"><input type="button" value="OK"/> <input type="button" value="Abbrechen"/> <input type="button" value="Hilfe"/></p> </div> <p>2. Geben Sie im "Name"-Feld einen beschreibenden Namen des Dateityps an. Dieser hat keinen Einfluss auf die Verschlüsselung.</p> <p>3. Geben Sie im "Dateiendung"-Feld die Dateierweiterung des Dateityps an, den Sie anlegen möchten, z.B. txt für Textdateien.</p> <p>4. Legen Sie im "Player Überprüfung"-Auswahlfeld fest, ob bei der Entschlüsselung eine Überprüfung der Lizenzoptionen der zugreifenden Anwendung stattfindet.</p> <p style="border: 1px solid #ccc; padding: 5px;">Lizenzliste Der Player (zugreifende Anwendung) muss mit einer Lizenz aus dieser Lizenzliste</p>	Name	Dateiendung	<input type="text"/>	<input type="text"/>	Player Überprüfung	Dateizugriffsmodus	<input type="button" value="0 - (default license)"/>	<input type="button" value="Auf Anfrage (on demand)"/>	Schreiboptionen:		Existierende Datei	<input type="button" value="Neue Datei"/>
Name	Dateiendung												
<input type="text"/>	<input type="text"/>												
Player Überprüfung	Dateizugriffsmodus												
<input type="button" value="0 - (default license)"/>	<input type="button" value="Auf Anfrage (on demand)"/>												
Schreiboptionen:													
Existierende Datei	<input type="button" value="Neue Datei"/>												

Abbildung 34: AxProtector - Dateiverschlüsselung "Dateityp hinzufügen"

2. Geben Sie im "Name"-Feld einen beschreibenden Namen des Dateityps an. Dieser hat keinen Einfluss auf die Verschlüsselung.
3. Geben Sie im "Dateiendung"-Feld die Dateierweiterung des Dateityps an, den Sie anlegen möchten, z.B. txt für Textdateien.
4. Legen Sie im "Player Überprüfung"-Auswahlfeld fest, ob bei der Entschlüsselung eine Überprüfung der Lizenzoptionen der zugreifenden Anwendung stattfindet.

Element	Beschreibung
	<p>verschlüsselt sein.</p> <p>i Dies erlaubt Ihnen beispielsweise festzulegen, dass auf einen bestimmten Datentyp ausschließlich mit einer von Ihnen verschlüsselten Anwendung zugegriffen werden kann.</p> <p>No player check Hier findet keine Überprüfung der zugreifenden Anwendung statt.</p> <p>5. Legen Sie im "Dateizugriffsmodus"-Auswahlfeld fest, wie der für den geschützten Dateizugriff vorbereitete Player (Anwendung zur Anzeige der Datendatei) auf die verschlüsselte Datendatei zugreift. Mit dem Dateizugriffsmodus können Sie den Speicherbedarf und das Laufzeitverhalten konfigurieren.</p> <p>i Die Wahl des passenden Modus hängt von der Art der Anwendung (des Players) und der Größe der Datei ab. Bei großen Videodateien sollte zum Beispiel die "Modus für große Dateien" Option verwendet werden. Bei kleinen Dateien (Konfigurationsdateien), auf die mehrmals zugegriffen wird bietet sich der "Auf einmal" Modus an. Da bei der Dateiverschlüsselungen auch unterschiedliche Laufzeiteinstellungen für zugreifende Anwendung und die Daten gewählt werden können, gelten zur Laufzeit die jeweils restriktiveren Einstellungen.</p>
Auf Anfrage	<p>Der Player reserviert im Hauptspeicher Platz für die komplette zu lesende Datei, liest aber nur den benötigten Teil - genaugenommen alle 4 kByte Blöcke, in denen dieser Teil enthalten ist - ein und entschlüsselt diese Blöcke. Bei weiteren Zugriffen auf die geschützte Datei werden weitere benötigte Blöcke (on demand) nachgeladen und entschlüsselt. Ist der benötigte Teil in bereits geladenen Blöcken, wird das entschlüsselte Abbild im Speicher verwendet. So baut der Player nach und nach ein komplettes Speicherabbild der benötigten Datei auf.</p> <p>i Dieser Modus benötigt viel Speicher (genauso viel wie die zu ladende Datei), bietet aber durch das Caching der entschlüsselten Daten eine gute Performance zur Laufzeit, wenn auf einen bereits entschlüsselten Block zugegriffen. Dieser Modus ist für lesenden und schreibenden Zugriff möglich.</p>
Auf einmal	<p>Der Player reserviert im Hauptspeicher Platz für die komplette zu lesende Datei, liest diese komplett ein und entschlüsselt sie komplett. Weitere Zugriffe auf die geschützte Datei erfolgen über das entschlüsselte Abbild im Speicher.</p> <p>i Dieser Modus benötigt viel Speicher (genauso viel wie die zu ladende Datei), bietet aber durch das Caching der entschlüsselten Daten eine gute Performance zur Laufzeit. Im Vergleich zum "Auf Anfrage" Modus benötigt dieser Modus mehr Zeit beim ersten Zugriff (wenn die Datei komplett geladen und entschlüsselt wird). Dafür liegt die Datei aber danach komplett entschlüsselt im Speicher und jeder weitere Zugriff ist performant. Dieser Modus ist für lesenden und schreibenden Zugriff möglich.</p>
Modus für große Dateien	<p>Der Player liest die gerade benötigten Teile der geschützten Datei ein und entschlüsselt diese. Die Daten werden im Speicher nicht als Cache gehalten.</p> <p>i Dieser Modus benötigt keinen zusätzlichen Speicher. Wenn auf die gleichen Daten mehrmals zugegriffen wird, werden die Daten jedes Mal neu gelesen und neu entschlüsselt. In diesem Modus ist nur lesender Zugriff möglich.</p>

Element	Beschreibung												
	<p>6. Legen Sie im "Schreiboptionen"-Bereich fest, wie die Dateien gespeichert werden.</p> <p>Existierende Datei</p> <p>Dieser Bereich regelt über Einstellungen, wie Änderungen an einer bestehenden Datei gespeichert werden.</p> <table border="1"> <tr> <td>Original</td><td>Hier sind Änderungen zugelassen. War die Datei verschlüsselt wird sie wieder verschlüsselt. Unverschlüsselte Dateien werden unverschlüsselt gespeichert.</td></tr> <tr> <td>No writing</td><td>Hier sind Schreibvorgänge nicht erlaubt, es besteht ausschließlich ein Read-Only-Zugriff.</td></tr> <tr> <td>Lizenzliste</td><td>Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.</td></tr> </table> <p>Neue Datei</p> <p>Dieser Bereich regelt über Einstellungen, wie Änderungen an einer neuen Datei gespeichert werden.</p> <table border="1"> <tr> <td>Plain</td><td>Neue Dateien werden grundsätzlich unverschlüsselt gespeichert.</td></tr> <tr> <td>No writing</td><td>Neue Dateien können nicht gespeichert werden. Es wird zwar eine neue Datei angelegt, darin aber keinerlei Daten abgespeichert.</td></tr> <tr> <td>Lizenzliste</td><td>Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.</td></tr> </table>	Original	Hier sind Änderungen zugelassen. War die Datei verschlüsselt wird sie wieder verschlüsselt. Unverschlüsselte Dateien werden unverschlüsselt gespeichert.	No writing	Hier sind Schreibvorgänge nicht erlaubt, es besteht ausschließlich ein Read-Only-Zugriff.	Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.	Plain	Neue Dateien werden grundsätzlich unverschlüsselt gespeichert.	No writing	Neue Dateien können nicht gespeichert werden. Es wird zwar eine neue Datei angelegt, darin aber keinerlei Daten abgespeichert.	Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.
Original	Hier sind Änderungen zugelassen. War die Datei verschlüsselt wird sie wieder verschlüsselt. Unverschlüsselte Dateien werden unverschlüsselt gespeichert.												
No writing	Hier sind Schreibvorgänge nicht erlaubt, es besteht ausschließlich ein Read-Only-Zugriff.												
Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.												
Plain	Neue Dateien werden grundsätzlich unverschlüsselt gespeichert.												
No writing	Neue Dateien können nicht gespeichert werden. Es wird zwar eine neue Datei angelegt, darin aber keinerlei Daten abgespeichert.												
Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.												

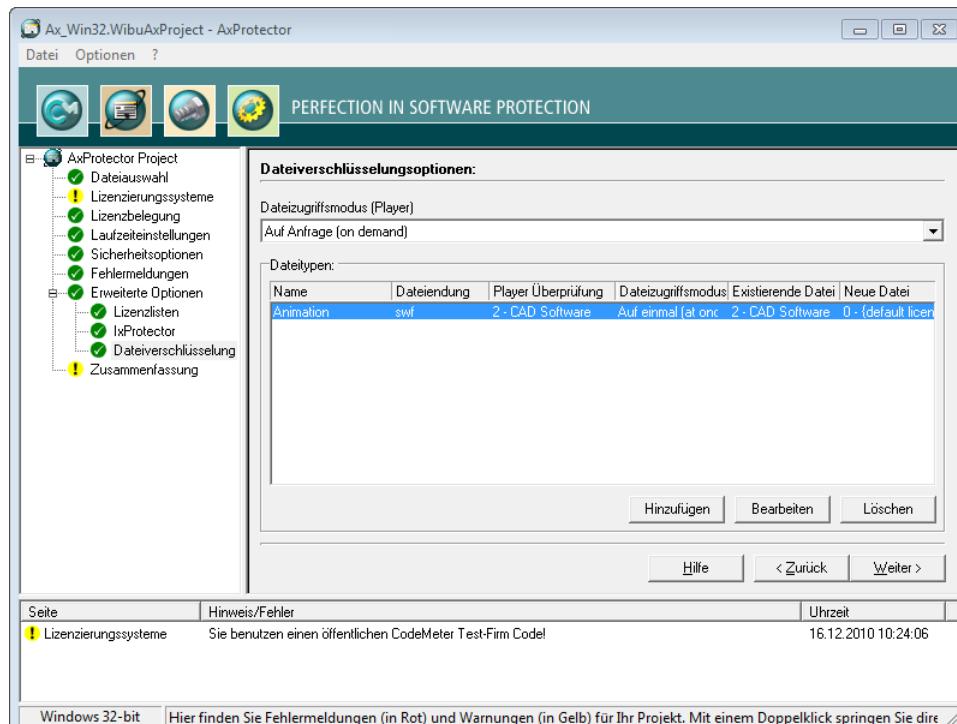


Abbildung 35: AxProtector - Dateiverschlüsselung "ausgefüllte Optionsliste"

7.4.1.8 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`

319

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

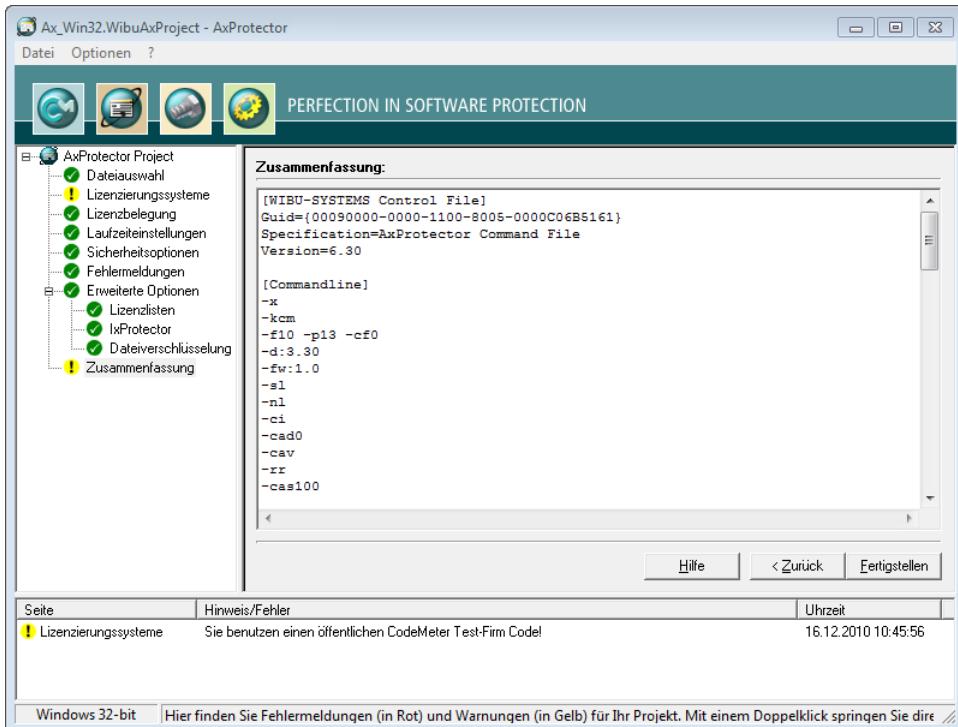


Abbildung 36: AxProtector - Windows "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

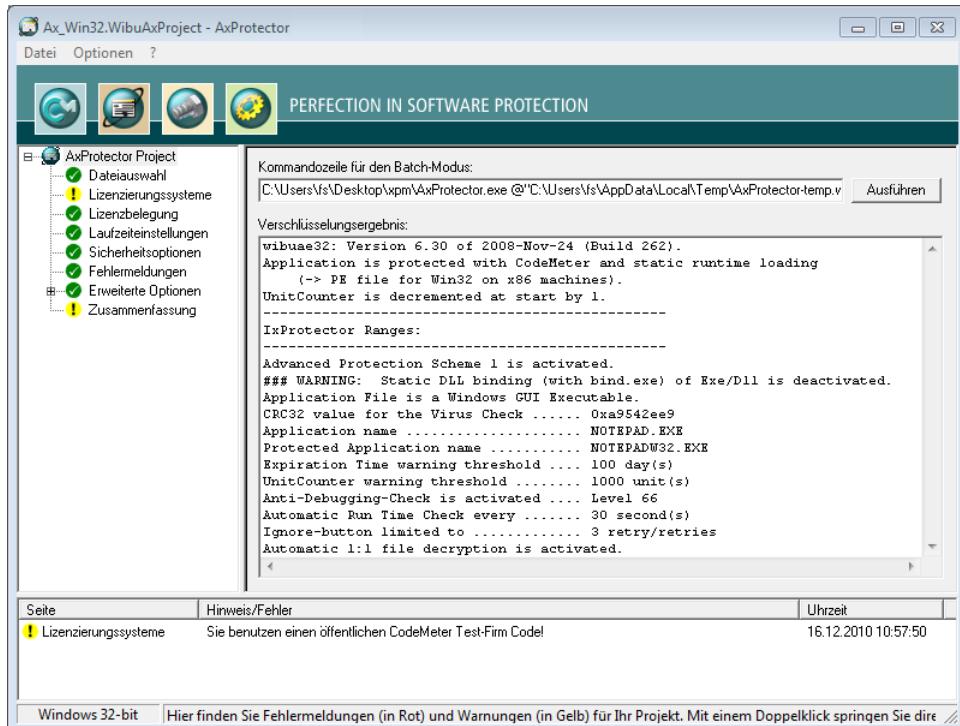


Abbildung 37: AxProtector - Windows "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	<p>Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector-Kommandozeile für den Batch-Modus ausgeführt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Die AxProtector-Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.4.2 NET Assembly

Eine .NET Assembly ist im Prinzip ein offenes Buch. Mit geeigneten Tools (z.B. Reflector) ist ein Disassembler und damit ein Reverse Engineering Ihres Codes einfach möglich. Um zu verhindern, das Unberechtigte Ihren Code analysieren und verändern, sollte der ausführbare Code also vor der Auslieferung auf jeden Fall verschlüsselt werden.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für .NET mit AxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
.NET Assembly	 AxProtector .NET ¹¹²		.NET Kommandozeile ¹²²



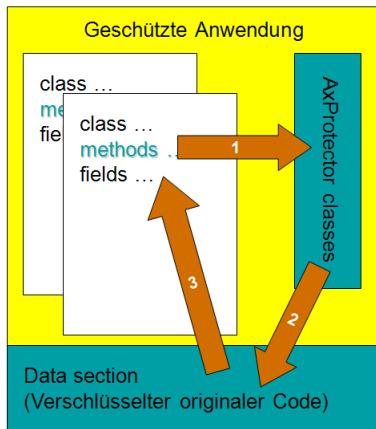
Beginnend mit der Version 4.20c wird auch das .NET 4.0 Framework unterstützt. Die neue Kommandozeilen-Variante `AxProtectorNet4.exe` ist in der Lage .NET 4.0 Assemblies zu benutzen. `AxProtector .NET 2.0` startet automatisch den `AxProtector .NET 4.0` wenn eine .NET 4.0 Assembly verschlüsselt wird.

Dabei geht AxProtector wie folgt vor:

Vorgehen

- Ihre Assembly wird durch `AxProtector .NET` disassembliert,
- Klassen, Methoden und Felder werden von der originalen Assembly extrahiert,
- eine neue Assembly wird erzeugt,
- es werden Klassen mit denselben Namen, Methoden und Feldern erzeugt,
- die neu erzeugten Methoden enthalten aber nicht den originalen Code, sondern Aufrufe auf die `AxEngine`,
- der originale Code wird mit einer von Ihnen gewählten Lizenz verschlüsselt in die Datensektion hinzugefügt,

Wenn nun zur Laufzeit die Methoden aufgerufen werden, entschlüsselt die `AxEngine` den originalen Code und führt ihn aus – natürlich nur, wenn Ihr Kunde die passende Lizenz hat. Weil die Methoden ihren ursprünglichen Namen behalten, können Sie diese auch weiterhin von außerhalb aufrufen. Selbst die Übergabeparameter (Typ und Bezeichnung) bleiben gleich!



Ein Disassemblieren des verschlüsselten Codes ist aber nicht mehr möglich

Sie selbst können festlegen, welche Methoden verschlüsselt werden und welche unverschlüsselt in der Assembly liegen. Die Festlegung treffen Sie wahlweise für einen kompletten Namespace, eine ganze Klasse, oder eine einzelne Methode.



Eine Festlegung auf der Methoden-Ebene überschreibt die Festlegung auf Klassen-Ebene. Und Festlegungen auf der Klasse-Ebene überschreiben Festlegungen auf der Namespace-Ebene.

Dabei können Sie festlegen ob:

- mit der Default-Lizenz verschlüsselt
- nicht verschlüsselt, oder
- mit separaten Lizenzlisten verschlüsselt wird.



Mit der letzten Option setzen Sie automatisch modularen Softwareschutz um.

7.4.2.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

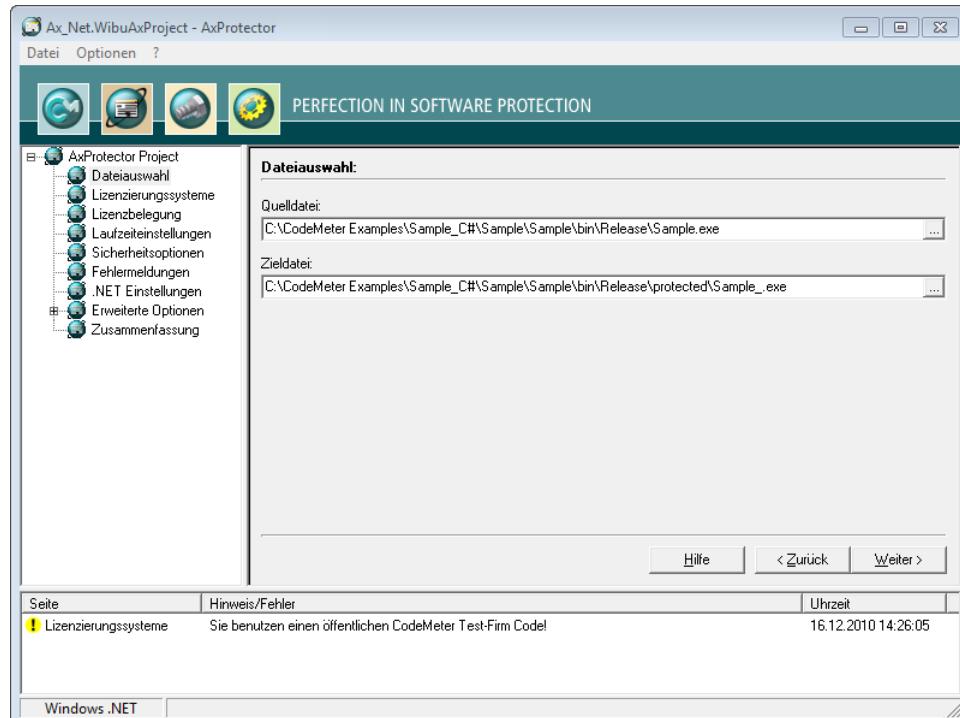


Abbildung 38: AxProtector - .NET "Dateiauswahl"

Dateiauswahl

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein. Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [.. \protected ..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.4.2.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Datei nehmen Sie hier Einstellungen zum verwendeten Lizenzierungssystem *CmDongle* und/oder *CmActLicense* vor.

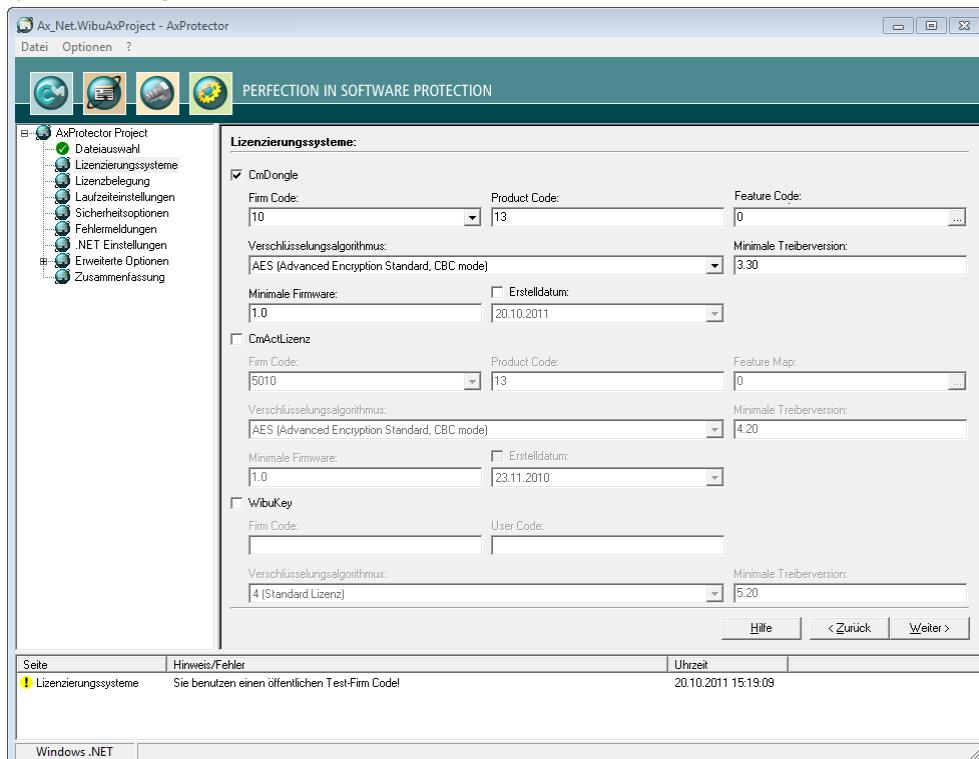


Abbildung 39: AxProtector - .NET "Lizenzierungssysteme"

Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenzierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenzierungssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die Wibu-Systems Interne Seiten.

Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich (siehe Kommandozeilen-Option [hier](#)²⁰³):

Element	Beschreibung
Firm Code	<p>Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird.</p> <p> Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle Evaluation-Firm Code</i> des <i>CodeMeter® Software Development Kits (SDK)</i> und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010. Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier ²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier ²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bewirkt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eintragen.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. <i>CodeMeter®</i> unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier ²⁹⁴.</p>
Minimale Treiberversion	<p>Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i> an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt AxProtector automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizzenzen.</p> <p> Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen beim Starten Ihrer geschützten Software.</p>

Element	Beschreibung
Erstelldatum	<p>Kommandozeilen-Option siehe hier²⁹⁴.</p> <p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden.</p> <p>Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <p> Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können.</p> <p>Beachten Sie auch, dass Sie hier das Kontrollkästchen aktivieren müssen, um Überprüfungsoptionen des Wartungszeitraumes (Maintenance Period) im Dialog zu den erweiterten Laufzeiteinstellungen¹²¹ vornehmen zu können.</p>
Minimale Firmware	<p>Kommandozeilen-Option siehe hier²⁹⁴.</p> <p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem *WibuKey* informiert separat das *WibuKey* Entwicklerhandbuch.

7.4.2.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Anwendung vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

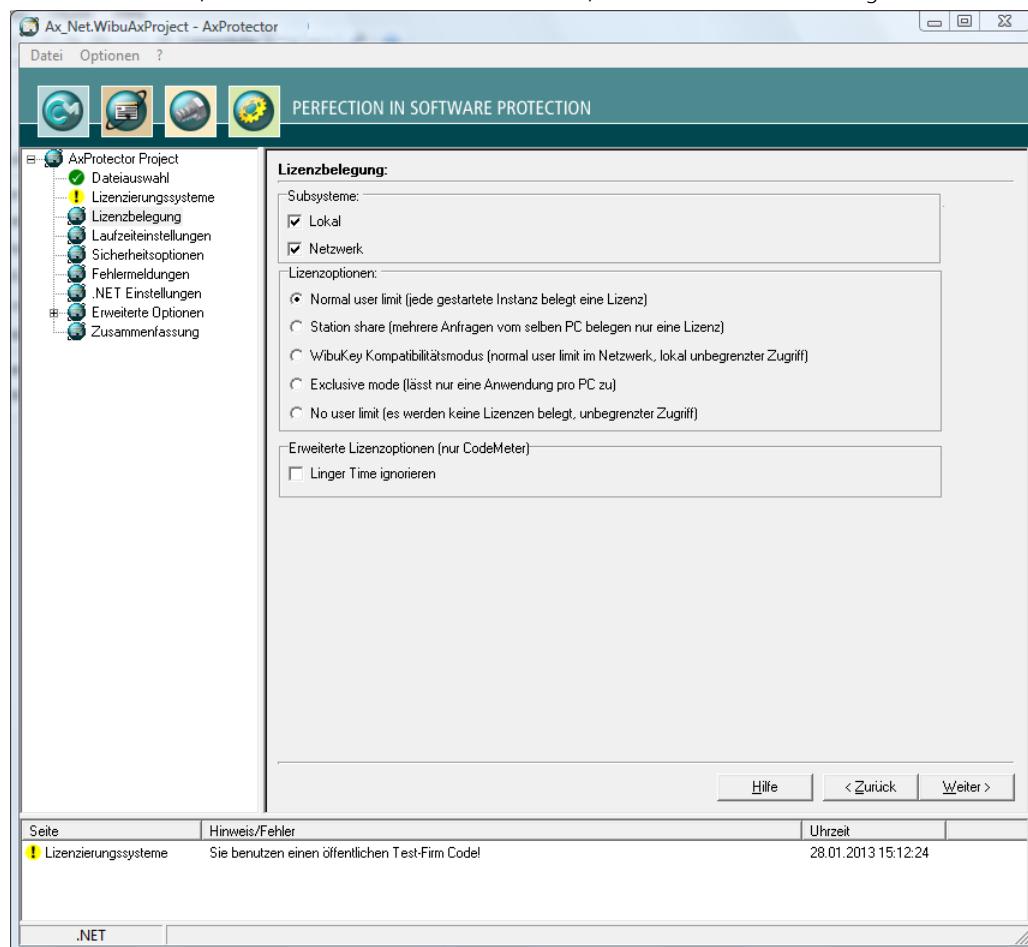


Abbildung 41: AxProtector - .NET "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#)²⁹⁴).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die

Element	Beschreibung
	sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der <i>CodeMeter Lizenzserver</i> mit einem aktivierte Netzwerkzugriff läuft. i Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizenzoptionen

Im Bereich Lizenzoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier](#)²⁰⁵).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz. i Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.
WibuKey Kompatibilitäts-Modus	Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit). i Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu <i>WibuKey</i> . <i>Wibu-Systems</i> empfiehlt die Einstellungen 'Normal user limit' und 'Station Share'.
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizenzen belegt werden. Belegte Lizenzen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenz Eigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden.

7.4.2.4 Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie das Verhalten der Anwendung zur Laufzeit fest, z.B. Abfrage der Lizenz in *CmContainer*, Ausgabe von Warnmeldung, etc..

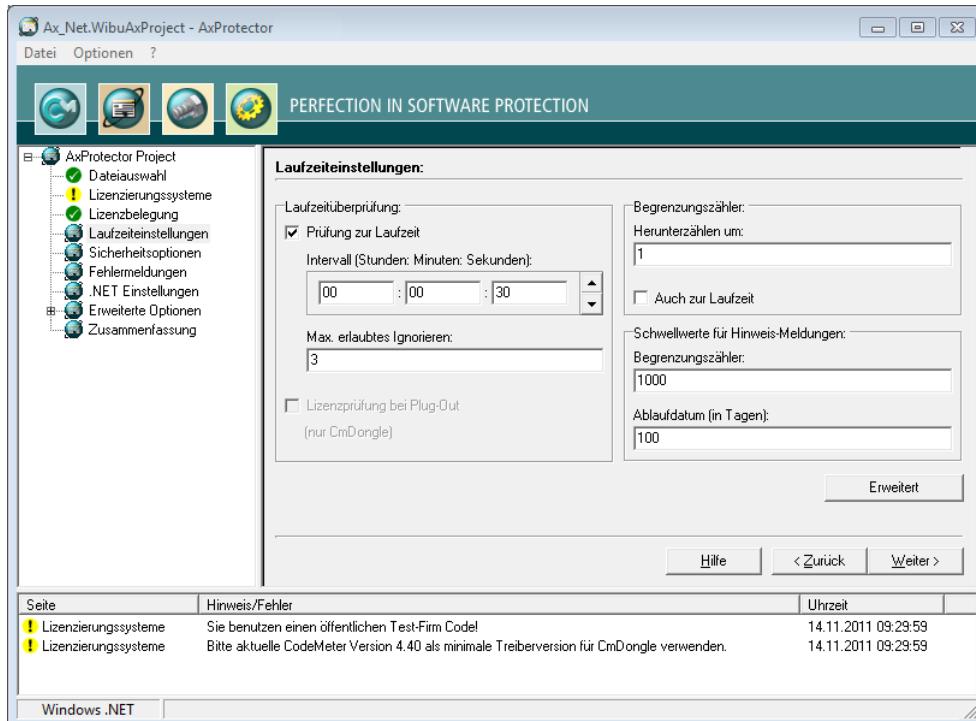


Abbildung 42: AxProtector - .NET "Laufzeiteinstellungen"

Laufzeitüberprüfung

In diesem Bereich können Sie definieren, ob und wie oft die geschützte Anwendung die Lizenz während der Laufzeit überprüft.

Elemente	Beschreibung
Prüfung zur Laufzeit	Aktiviert oder deaktiviert die Überprüfung während der Laufzeit der geschützten Anwendung. Kommandozeilen-Option siehe hier .
Intervall	Legt das Intervall zwischen zwei Überprüfungen fest. Angabe im Format Stunden: Minuten: Sekunden.
Max. erlaubtes Ignorieren	Gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann. Schlägt die Verbindung zum <i>CmContainer</i> fehl, d.h. kann nicht mehr auf die Lizenz zugriffen werden, geben Sie dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit, auch ohne die Lizenz noch weiterzuarbeiten.
Lizenzprüfung bei	Beendet die geschützte Anwendung, wenn der <i>CmDongle</i> während der Ausführung abgezogen

Elemente	Beschreibung
Plug-Out (nur CmDongle)	wird und eine sofortige Fehlermeldung wird ausgegeben. Kommandozeilen-Option siehe hier ²⁹⁷ .

Begrenzungszähler

Begrenzungszähler (Unit Counter) können u.a. dazu dienen, die Gültigkeit von Lizzenzen in einem *CmContainer* festzustellen. In diesem Bereich können Sie dieses Verhalten definieren (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Herunterzählen um	Gibt den Wert an, um den der Begrenzungszähler (Unit Counter) heruntergezählt wird. Diese Option bewirkt das Herunterzählen des Zählers beim Start der geschützten Anwendung. Ist die "Auch zur Laufzeit" Option aktiviert und sind die Einträge wie in der obigen Abbildung dargestellt gesetzt, wird alle 30 Sekunden (siehe das festgelegt Intervall) ein gesetzter Begrenzungszähler (Unit Counter) um den Wert 1 heruntergezählt.
Auch zur Laufzeit	Zählt den Begrenzungszähler (Unit Counter) auch während der Laufzeit der geschützten Anwendung herunter.  Diese Option greift nur, wenn die "Prüfung zu Laufzeit" Option im Bereich "Laufzeit-überprüfung" aktiviert ist.

Schwellenwerte für Hinweismeldungen

In diesem Bereich können Sie definieren, wann eine Hinweismeldung zur Gültigkeit der Lizenz ausgegeben wird.

	Zur individuellen Gestaltung des Textes der Hinweismeldungen siehe hier ¹²⁷ .
---	--

Element	Beschreibung
Begrenzungszähler	Wird der angegebene Schwellenwert unterschritten, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .
Ablaufdatum (in Tagen)	Wird das angegebene Ablaufdatum in Tagen innerhalb der vorgegebenen Schwelle erreicht, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .

7.4.2.4.1 Erweiterte Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie zusätzliche Einstellungen zur Laufzeit der verschlüsselten Anwendung fest.

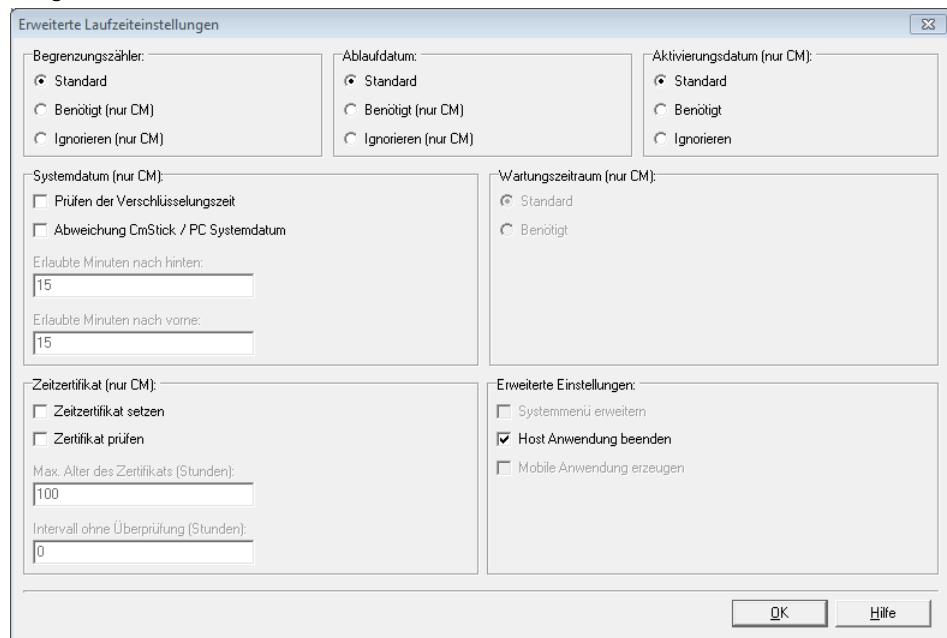


Abbildung 43: AxProtector - .NET "Erweiterte Laufzeiteinstellungen"

Für die Abfrage der in die Lizenz eingetragenen Optionen Begrenzungszähler (Unit Counter), Ablaufdatum (Expiration Time) und Aktivierungsdatum (Activation Time) gilt die folgende Handhabung.

Status	Standard	Benötigt	Ignorieren
= 0	X	X	✓
< > 0	✓	✓	✓
nicht angegeben	✓	✓	✓

Begrenzungszähler (Unit Counter)

Definiert die Handhabung eines Unit Counter (Begrenzungszählers), der in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Standard	Zählt einen vorhandenen Unit Counter-Eintrag in der Lizenz beim Start und/oder zur Laufzeit um den auf der vorherigen Seite definierten Wert herunter. Wenn der Unit Counter Null erreicht startet die verschlüsselte Anwendung nicht.
Benötigt	Ein Unit Counter-Eintrag < > 0 in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag star-

Element	Beschreibung
	tet die verschlüsselte Anwendung nicht.
Ignorieren	Ein vorhandener Unit Counter-Eintrag in der Lizenz wird ignoriert. Die Anwendung setzt den Unit Counter nicht herunter. Die Anwendung startet auch bei einem Unit Counter-Eintrag = 0.

Ablaufdatum (Expiration Time)

Definiert die Handhabung einer Expiration Time (Ablaufdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Expiration Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn keine Expiration Time vorhanden ist, oder das aktuelle Datum vor der Expiration Time liegt.
Benötigt	Ein Expiration Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten.
Ignorieren	Ein vorhandener Expiration Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum nach der Expiration Time liegt.

Aktivierungsdatum (Activation Time)

Definiert die Handhabung einer Activation Time (Aktivierungsdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Activation Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn kein solcher Eintrag vorhanden ist, oder die zertifizierte Zeit ⁴¹⁷ nach der Activation Time liegt.
Benötigt	Ein Activation Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten. Beachten Sie, dass dann eine Internet-Verbindung zum Abholen der zertifizierten Zeit erforderlich ist.
Ignorieren	Ein vorhandener Activation Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum vor der Activation Time liegt.

Wartungszeitraum (Maintenance Period)

Definiert die Handhabung eines Wartungszeitraumes (Maintenance Period), der in der Lizenz eingetragen ist. Eine Lizenz berechtigt dann zur Verwendung aller Softwareversionen, die innerhalb des definierten Wartungszeitraumes (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der Applikation hinterlegt und zur Laufzeit der geschützten Anwendung geprüft, ob das Erstelldatum (Release Date) innerhalb des Wartungszeitraumes (Maintenance Period) liegt (Kommandozeilen-Option siehe [hier](#)³⁰⁷).



Die Optionen sind nur auswählbar, wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) [aktiviert](#)¹¹⁵ worden ist.

Es bestehen zwei Überprüfungsoptionen:

Element	Beschreibung
Standard	Während der Laufzeit der geschützten Anwendung wird gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist. Dies entspricht der Standardeinstellung

Element	Beschreibung
	auch wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) nicht aktiviert ¹¹⁵ worden ist.
Benötigt	Während der Laufzeit der geschützten Anwendung ist das Prüfen des Wartungszeitraumes (Maintenance Period) gegen das Erstelldatum (Release Date) zwingend erforderlich. Die PIO Wartungszeitraum (Maintenance Period) muss vorhanden sein.

Zeitzertifikat

In jedem *CmContainer* ist eine laufende Uhr integriert, die läuft, wenn der *CmContainer* mit dem Rechner verbunden ist. Die Uhrzeit synchronisiert sich dabei beim Aktivieren des *CmContainers* nach vorne und nutzt ansonsten die letzte gespeicherte Zeit.

Wenn gewünscht, kann die zertifizierte Uhrzeit durch die Synchronisation mit dem *CodeMeter®* Zeitserver aktualisiert werden. Die Zeitserver sind von Wibu-Systems bereitgestellte Rechner, die über die Welt verteilt sind und eine zertifizierte Zeit zur Verfügung stellen. Bei einer Aktualisierung der zertifizierten Uhrzeit wird die interne *CmContainer*-Zeit synchronisiert (Kommandozeilen-Option siehe [hier](#)³⁰⁰).

 Für Informationen zur Manipulationssicherheit von Aktivierungs- und Ablaufdatum siehe hier ⁴¹⁷
--

Element	Beschreibung
Zeitzertifikat setzen	Mit dieser Option wird versucht die zertifizierte Zeit im <i>CmContainer</i> zu aktualisieren. Die zertifizierte Zeit wird beim Zeitserver angefordert.  Diese Option erfordert eine Internet-Verbindung.
Zertifikat prüfen	Diese Option überprüft, ob die zertifizierte Zeit älter ist, als das hier festlegbare maximale Alter. Ist das maximale Alter des Zeitzertifikats überschritten, so lässt sich die Anwendung nicht starten.
Max. Alter des Zertifikats (in Stunden)	Bei ausgewählter "Prüfung" des Zeitzertifikats können Sie hier das maximale Alter des Zertifikats in Stunden angeben. Das Alter des Zertifikates berechnet sich aus der Differenz der laufenden System-Zeit und der zertifizierten Zeit.
Intervall ohne Überprüfung (Stunden)	Gibt an, innerhalb welchen Intervalls <u>keine</u> Überprüfung des Zeitzertifikats stattfindet. Ist dieses Intervall noch nicht erreicht, findet keine Überprüfung statt. Befindet sich das Zeitzertifikat zwischen diesem Intervall und dem max. Alter des Zertifikats, wird versucht, das Zeitzertifikat zu aktualisieren. Gelingt dies nicht, läuft die Anwendung jedoch bis zum Erreichen des max. Alters des Zeitzertifikats weiter. Erst danach ist zwingend ein aktualisiertes Zeitzertifikat notwendig.

System Datum

In diesem Bereich nehmen Sie Einstellungen vor, die dem zusätzlichen Schutz dienen, eine Lizenz über ein bewusstes Falschstellen der PC-Zeit zu manipulieren (Kommandozeilen-Option siehe [hier](#)²⁹⁷).

Element	Beschreibung
Prüfen der Verschlüsselungszeit	Diese Option speichert die Verschlüsselungszeit (PC Time) in der geschützten Anwendung. Die Anwendung läuft auf dem Kunden-PC dann nur, wenn die <i>CmContainer</i> Systemzeit neuer ist als die Verschlüsselungszeit.

Element	Beschreibung
	 Erfordert mindestens <i>CodeMeter®</i> 4.10.
Abweichung CmContainer / PC Systemzeit	Wird diese Option aktiviert, ist die Festlegung eines Zeitkorridors möglich, innerhalb dessen sich die Abweichung zwischen <i>CmContainer</i> Systemzeit und der PC-Zeit bewegen darf. Wird dieser unter- bzw. überschritten läuft die geschützte Anwendung auf dem Kunden-PC nicht.
Erlaubte Minuten nach hinten	Gibt in Minuten an, um wieviele Minuten die PC Zeit älter als die <i>CmContainer</i> Systemzeit sein darf.
Erlaubte Minuten nach vorne	Gibt in Minuten an, um wieviele Minuten die PC Zeit vor der <i>CmContainer</i> Systemzeit liegen darf.

Erweiterte Optionen

Dieser Bereich lässt die Auswahl weiterer Optionen zu.

Element	Beschreibung
Host Anwendung beenden	Wenn keine gültige Lizenz gefunden wird, wird im Falle geschützter Dll-Anwendungsdateien die aufrufende *.exe beendet Kommandozeilen-Option siehe hier ³¹⁰).
Mobile Anwendung erzeugen	[noch nicht implementiert]

7.4.2.5 Sicherheitsoptionen

Über diese Seite treffen Sie eine Auswahl aus verschiedenen Schutzmethoden für Ihre Anwendung. Dabei können Sie auf Debugger prüfen oder den ganzen *CmContainer* sperren.

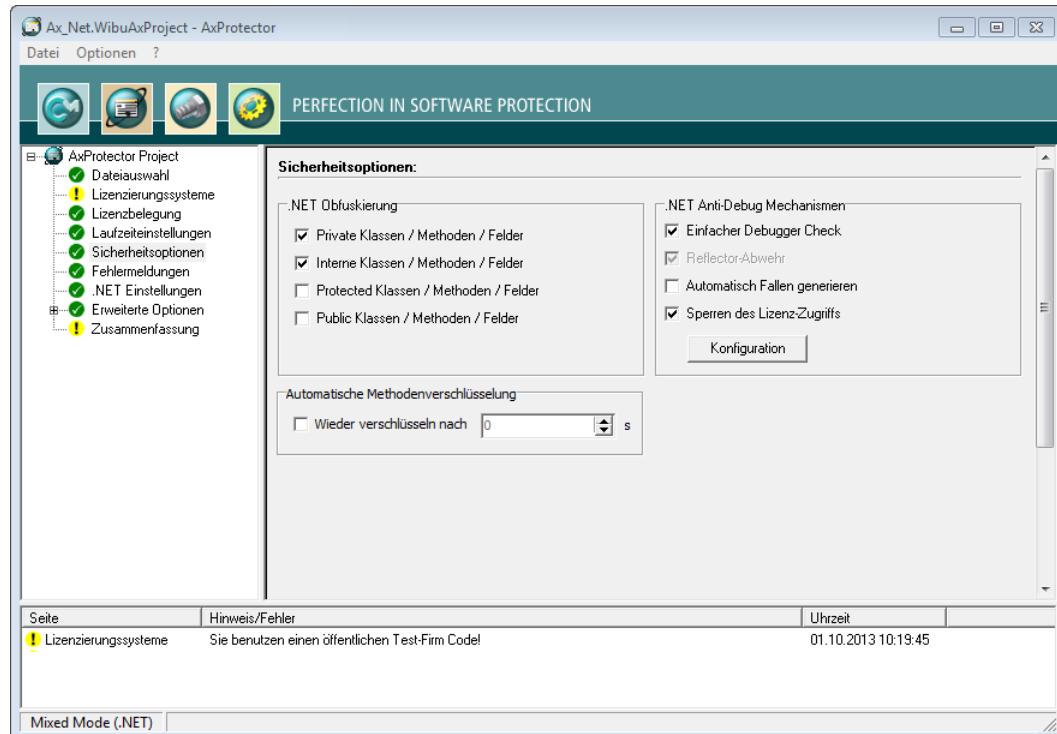


Abbildung 44: AxProtector - .NET "Sicherheitsoptionen"

.NET Obfuscierung

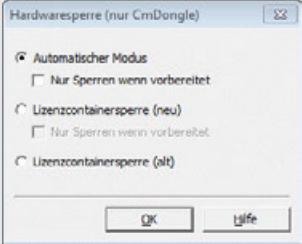
Die Obfuscierung ersetzt les- und nachvollziehbare Element-Namen durch maschinengenerierte Benennungen, verschleiert Programm-Informationen und schützt so vor einem Reverse Engineering (Kommandozeilen-Option siehe [hier](#)³⁰⁵). Elemente umfassen Klassen, Methoden und Felder.

Element	Beschreibung
Private Klassen / Methoden / Felder	Obfusciert Private-Elemente.
Interne Klassen / Methoden / Felder	Obfusciert Internal-Elemente.
Protected Klassen / Methoden / Felder	Obfusciert Protected-Elemente.
Public Klassen / Methoden / Felder	Obfusciert Public-Elemente.

Anti-Debugging Mechanismen

Debugger-Programme dienen der Fehlersuche und Fehlerbeseitigung, können aber auch von Hackern zur Analyse der Software verwendet werden. In diesem Bereich legen Sie die Optionen fest, wie auf Debugger-Programme reagiert werden soll (Kommandozeilen-Option siehe [hier](#)²⁹⁸).

Element	Beschreibung						
Einfacher Debugger Check	Überprüft ob ein Debugger an Ihre Anwendung angehängt (attached) ist. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.						
Reflector-Abwehr	Geschützte .NET Assemblies enthalten automatisch eine Reflektor-Abwehr gegen ein mögliches Decompiling.						
Automatisch Fallen generieren	Fügt automatisch Hacker-Fallen in die verschlüsselte Assembly ein (Kommandozeilen-Option siehe hier ³¹³).						
Sperren des Lizenz-Zugriffs	Mit dieser Option kann das genutzte Firm Item im <i>CmContainer</i> gesperrt werden sobald ein Debugger-Programm entdeckt wird. Wird die Option aktiviert, werden die Einstellungen übernommen, die Sie in einem Dialog setzen, der sich über die "Erweitert"-Schaltfläche öffnet.						
<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">  Diese Schaltfläche ist nur aktiviert für CodeMeter. </div>							
Konfiguration	<p>Wenn die Option "Sperren des Lizenz-Zugriffs" aktiviert wird, können Sie über die "Konfiguration"-Schaltfläche im folgenden Dialog weitere Einstellungen vornehmen: Der Dialog erlaubt in Abhängigkeit von der Firmware, die bei der Verschlüsselung verwendet wird, die Auswahl verschiedener Sperrszenarien.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Sperrszenario</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>sofortiges Sperren</td><td>erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.</td></tr> <tr> <td>vorbereitetes Sperren</td><td>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</td></tr> </tbody> </table>	Sperrszenario	Beschreibung	sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.	vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.
Sperrszenario	Beschreibung						
sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.						
vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.						

Element	Beschreibung	
	<p>Sperrszenario</p> <p>Beschreibung</p> 	
	<p>Abbildung 45: AxProtector - .NET "Sicherheitsoptionen - Hardware-Sperre"</p> <p>Die folgenden Einstellungen sind verfügbar</p>	
	Einstellung	
	<p>"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)</p> <p>"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert</p> <p>"Lizenzcontainersperre (neu)" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert</p> <p>"Lizenzcontainersperre (neu)" markiert und das Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert</p> <p>Option "Lizenzcontainersperre (alt)" markiert</p>	<p>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt.</p> <p>Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items.</p> <p>Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</p> <p>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.</p> <p>Ist die Firmware 1.14 und höher, dann wird gleichzeitig geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.</p> <p>Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt.</p> <p>Dies ist sicherheitstechnisch gesehen die empfohlene Einstellung. Voraussetzung ist jedoch, dass alle CmContainer im Feld mit einer Firmware Version 1.14 und höher ausgestattet sind.</p> <p>Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt.</p> <p>Gleichzeitig wird geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.</p> <p>Gilt für alle Firmware-Versionen. Ist ein vorbereitetes Sperren programmiert, wird der FAC um den Wert 1 heruntergezählt.</p>

Automatische Methodenverschlüsselung

Dieser Bereich ermöglicht, die Methoden nach n Sekunden Nichtbenutzung wieder zu verschlüsseln (Kommandozeilen-Option siehe [hier](#)^[306]).

Element	Beschreibung
Wieder verschlüsseln nach	Aktiviert das Feature, die Methoden nach Nichtbenutzung wieder zu verschlüsseln.
s	Zahlenfeld, das die Angabe von Sekunden ermöglicht, nach dem die nichtbenutzten Methoden wieder verschlüsselt werden sollen.

7.4.2.6 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

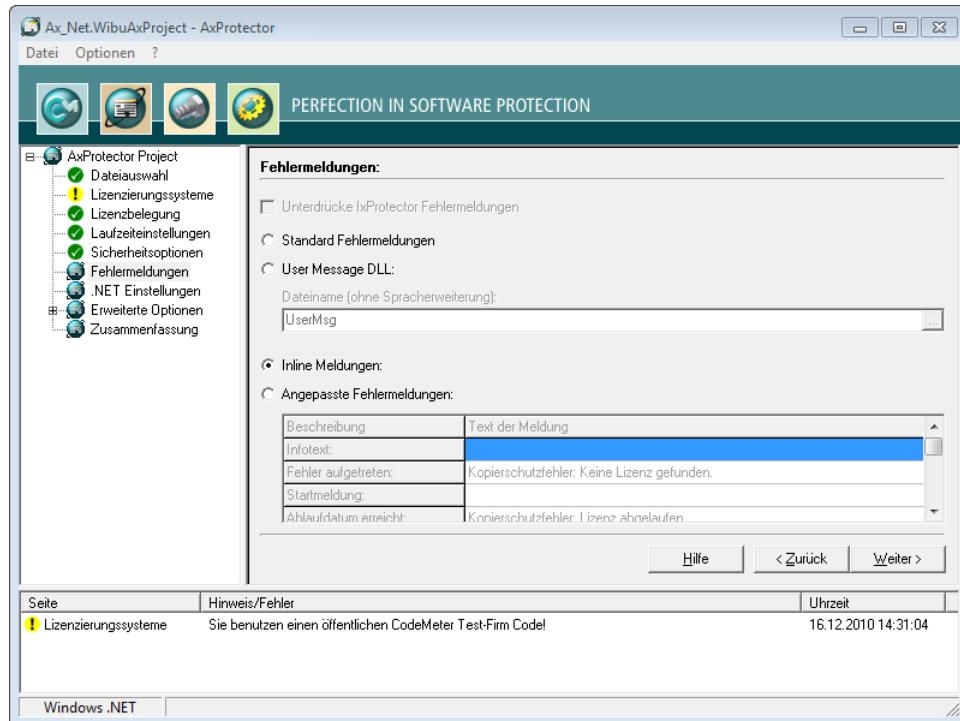
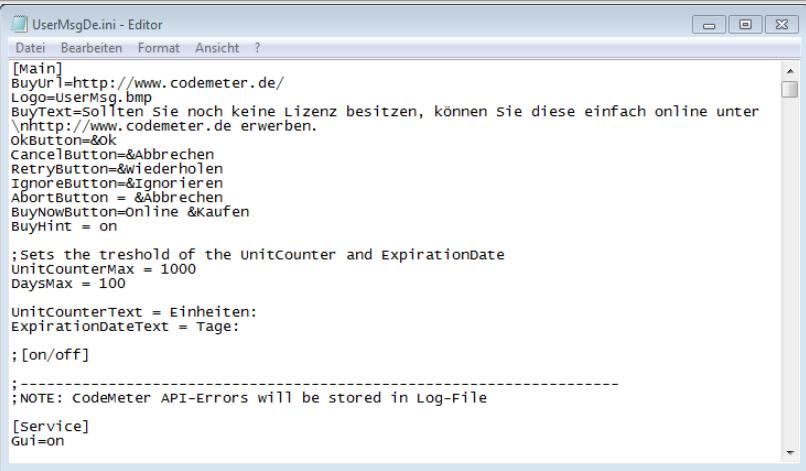


Abbildung 46: AxProtector - .NET "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Unterdrücke IxProtector Fehlermeldungen	Unterdrückt die Ausgabe von IxProtector Fehlermeldungen (Kommandozeilen-Option siehe hier ³⁰⁴).

Element	Beschreibung
dungen	<p> Setzen Sie diese Option nicht, so werden bei der Verwendung von <i>lProtect</i> im Fehlerfalle zusätzliche Meldungsfenster angezeigt, und zwar zusätzlich zu den im Projekt selbst ausprogrammierten Meldungen.</p>
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
User Message DLL	<p>Aktiviert die Benutzung der User Message DLL. Die Fehlertexte können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen (Kommandozeilen-Option siehe hier³¹²).</p> <p> Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die <i>AxProtector</i> geschützte Anwendung befindet.</p>  <pre>[Main] Buyurl=http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten sie noch keine Lizenz besitzen, können sie diese einfach online unter \nhttp://www.codemeter.de erwerben. OkButton=&k CancelButton=&Abbrechen RetryButton=&wiederholen IgnoreButton=&Ignorieren AbortButton = &abbrechen BuyNowButton=Online &Kaufen BuyHint = on ;sets the threshold of the unitCounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 unitCounterText = Einheiten: ExpirationDateText = Tage: ; [on/off] ; ----- ;NOTE: CodeMeter API-Errors will be stored in Log-File [service] Gu1=on</pre>
	Abbildung 47: <i>AxProtector – UserMsgDe.ini</i>
Inline Meldungen	<p>Dateiname (ohne Spracherweiterung)</p> <p>Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an. Die UserMsgDll wird aus dem Verzeichnis <code>%Program Files%\WI BU-SYSTEMS\AxProtector\DevKit\bin\UserMessage</code> kopiert. Die jeweiligen Initialisierungsdateien sind ebenfalls in diesem Verzeichnis abgelegt.</p> <p> Linkt für .NET Projekte eine inline assembly und kann ebenfalls über *.ini-Dateien konfiguriert werden (Kommandozeilen-Option siehe hier³¹²).</p> <p>Bei Verwendung der Inline UserMessages erfolgt das Logging in das Verzeichnis "%CommonApplicationData%". Ein anderer Pfad kann in der *.INI-Datei mit <code>LogPath=<Pfad></code> angegeben werden.</p>
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.4.2.7 .NET Einstellungen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für .NET-Verschlüsselungen vorzunehmen.

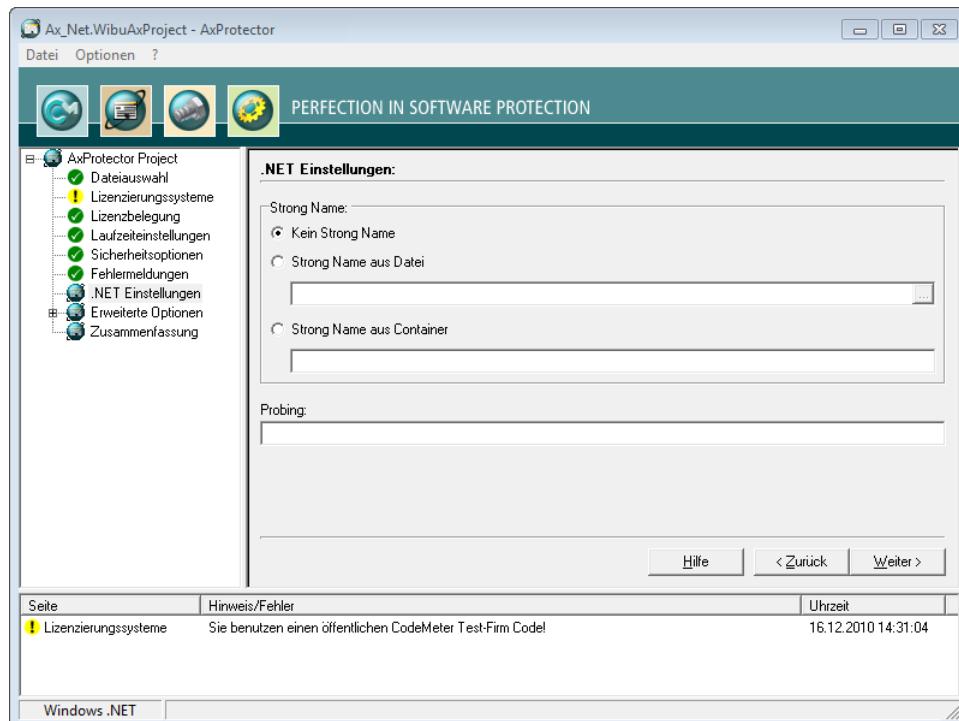


Abbildung 48: AxProtector - .NET ".NET Einstellungen"

.NET Einstellungen

Dieser Bereich erlaubt Ihnen Einstellungen zur Signierung Ihrer Assembly durch AxProtector.

Element	Beschreibung
Kein Strong Name	Die Assembly wird nicht signiert.
Strong Name aus Datei	Zur Signierung der Programmklasse, können Sie hier eine Datei als Quelle für das Schlüsselpaar zur Generierung eines Strong Names angeben (Kommandozeilen-Option siehe hier).
Strong Name aus Container	Geben Sie hier den Namen des Containers zur Signierung Ihrer Programmklasse an (Kommandozeilen-Option siehe hier).
Probing	Der Bereich erlaubt Angaben zum Lageort für signierte Programmklassen in einer app.config Datei.  Geben Sie den Pfad an, auf dem der Zugriff auf die Programmklasse erfolgt, getrennt durch ",". Oder geben Sie die jeweilige app.config Datei an.

Element	Beschreibung
	Kommandozeilen-Option siehe hier ³¹² .

7.4.2.8 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

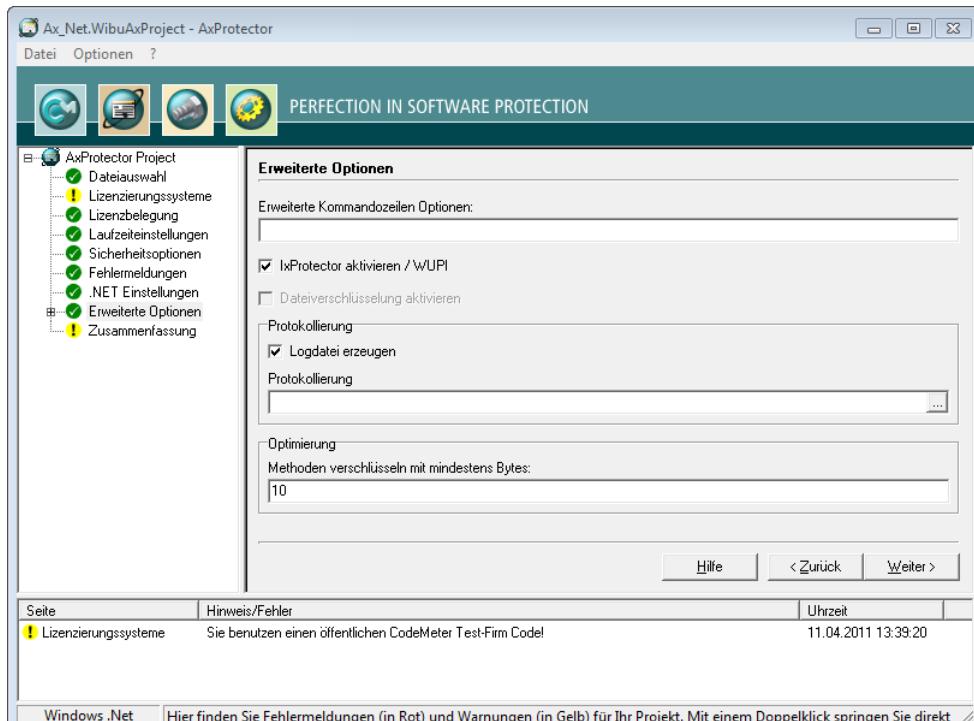


Abbildung 49: AxProtector - .NET "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
IxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das Softwareschutz-API (WUPI) ³²⁰ verwenden (Kommandozeilen-Option siehe hier ³⁰⁵).

Element	Beschreibung
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.  Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.
Optimierung	Geben Sie hier zu Optimierungszwecken ein, welche Mindestgröße ein Assembly besitzen muss, damit es verschlüsselt wird. Die Standard-Einstellung beträgt 10 Bytes (Kommandozeilen-Option siehe hier ³⁰⁶).

7.4.2.8.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *IxProtector* über das [Softwareschutz-API \(WUPI\)](#)³²⁰.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

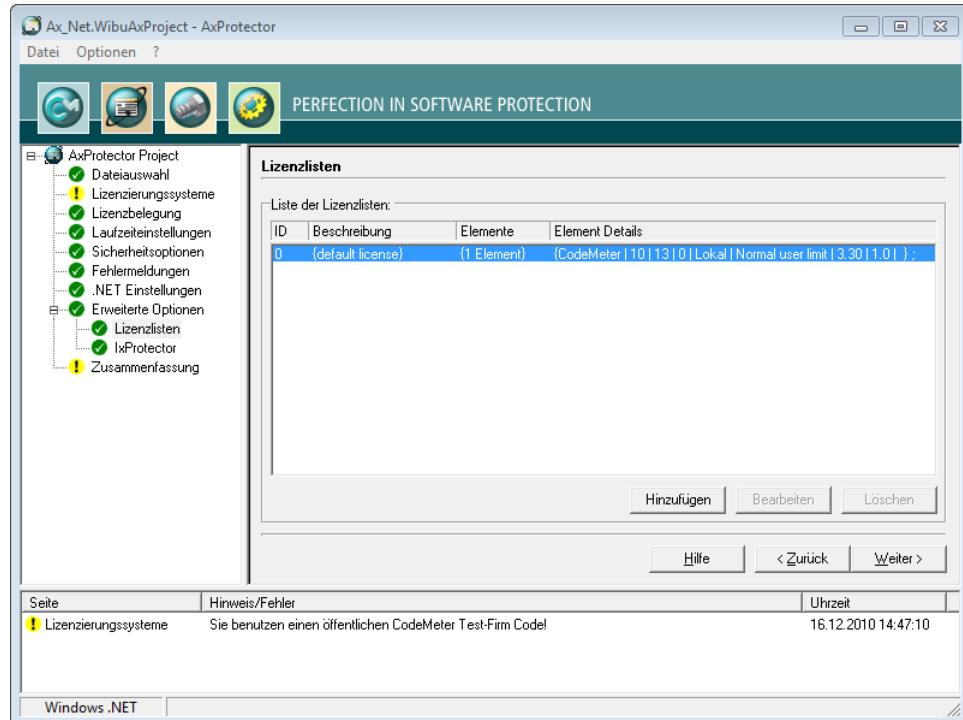
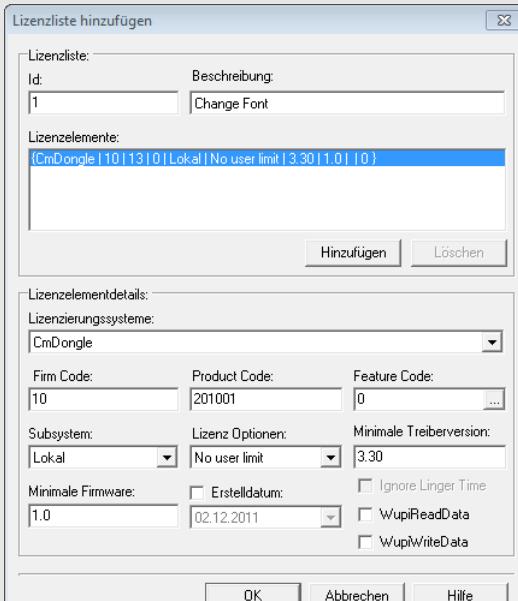


Abbildung 50: AxProtector - .NET - "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die **"Hinzufügen"** Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	<p>Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.</p> <p>i Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.</p>
Beschreibung	Beschreibt die Lizenzliste über einen Texteintrag.

Element	Beschreibung
	<p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
	<p>Abbildung 51: AxProtector - .NET Lizenzlisten hinzufügen</p>
Lizenzierungs systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (CmDongle, CmActLicense oder WibuKey).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	<p>Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt.</p> <p>Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal).</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenzen:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem CmContainer, wenn diese Da-

Element	Beschreibung
	ten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

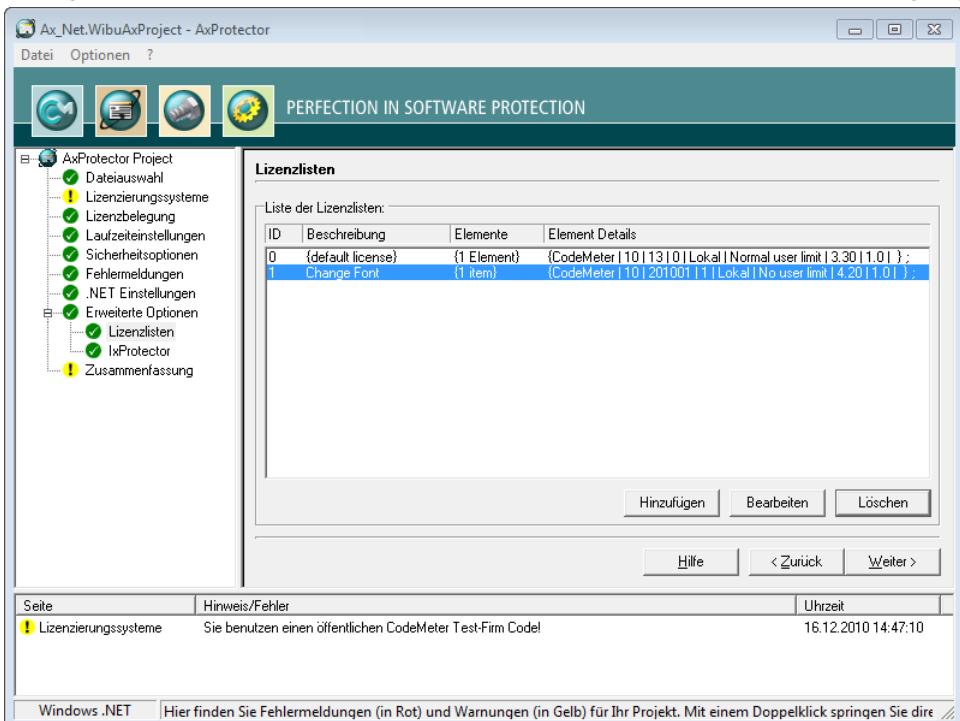


Abbildung 52: AxProtector - .NET - "ausgefüllte Lizenzliste"

7.4.2.8.2 IxProtector

Über diesen Menü-Eintrag definieren Sie Verschlüsselungstypen für einzelne Assembly-Elemente. Haben Sie im Menü-Eintrag "**Erweiterte Optionen**" den Eintrag "**IxProtector**" aktiviert, so wird die Quell-Assembly geladen und die gesamte Baumstruktur aus Namespaces, Klassen und Modulen ist verfügbar und wird angezeigt.

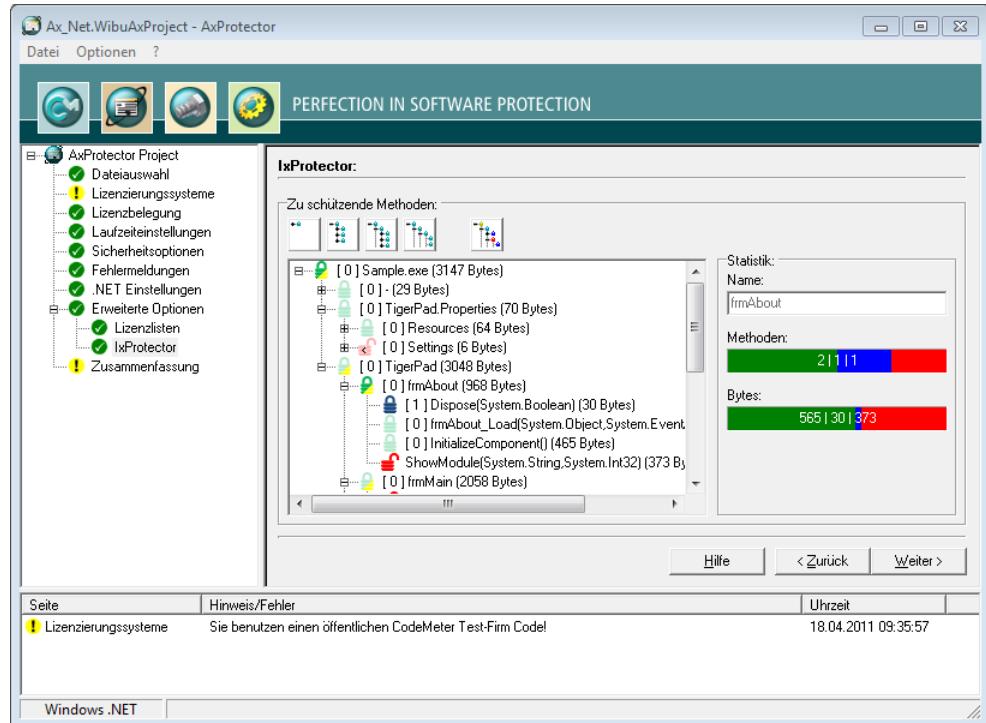


Abbildung 53: AxProtector - .NET - "IxProtector"

Sie können aus verschiedenen Assembly-Ansichten auswählen, indem Sie die oberen Schaltflächen im IxProtector-Bereich benutzen.

Ansichten

Schaltfläche	Beschreibung
	klappt alle Ebenen der Assembly in der Baumstruktur zu.
	klappt die Namespace-Ebene der Assembly auf.
	klappt die Klassen-Ebene der Assembly auf.
	klappt die Methoden-Ebene der Assembly auf.
	klappt alle Ebenen der Assembly auf, für die Änderungen vorgenommen worden sind.

Der Statistik-Bereich auf der rechten Seite zeigt Ihnen weitere Verschlüsselungsdetails in Abhängigkeit der Auswahl, die Sie in der Baumansicht getroffen haben.

Element	Beschreibung
Name	Dieses Feld verweist auf den Namen des Elementes, das Sie in der Baumansicht ausgewählt haben.

Element	Beschreibung	
Methoden	Im Methods-Balken zeigen die unterschiedlichen Farben an, mit welcher Schutztechnologie verschlüsselt bzw. ob nicht verschlüsselt wird. Die Zahlen-Angaben informieren gleichzeitig über die Anzahl der verschlüsselten bzw. unverschlüsselten Methoden für jede Schutztechnologie.	
Farbe	Beschreibung	
Grün	zeigt an, dass die Methode mit AxProtector verschlüsselt wird und die Lizenzlisten ID einen Wert von 0 hat (die Default-Lizenz).	
Blau	zeigt an, dass die Methode mit IxProtector verschlüsselt wird und die Lizenzlisten ID einen Wert ungleich 0 hat.	
Rot	zeigt an, dass die Methode unverschlüsselt ist.	
Bytes	Im Bytes-Balken zeigen die unterschiedlichen Farben ebenfalls an, mit welcher Schutztechnologie verschlüsselt bzw. ob nicht verschlüsselt wird. Die Zahlen-Angaben informieren gleichzeitig über die Anzahl der verschlüsselten bzw. unverschlüsselten Bytes der jeweiligen Schutztechnologie.	
Farbe	Beschreibung	
Grün	zeigt an, dass die Methode mit AxProtector verschlüsselt wird und die Lizenzlisten ID einen Wert von 0 hat (die Default-Lizenz).	
Blau	zeigt an, dass die Methode mit IxProtector verschlüsselt wird und die Lizenzlisten ID einen Wert ungleich 0 hat.	
Rot	zeigt an, dass die Methode unverschlüsselt ist.	

Sie besitzen auch die Möglichkeit, die Schutztechnologien *AxProtector* und *IxProtector* separat einzelnen Assembly-Elementen zuzuweisen bzw. Elemente von der Verschlüsselung auszuschließen. Diese Zuweisung nehmen Sie über ein Kontextmenü wie folgt vor:

1. Wählen Sie das gewünschte Assembly-Element (Namespace, Klasse oder Methode) aus der links stehenden Baumstruktur.
2. Klicken Sie die rechte Maustaste.
Das Kontextmenü öffnet sich.
3. Weisen Sie über die Symbole den gewünschten Verschlüsselungstypen zu.

Die zur Auswahl angebotenen Lizenzlisten IDs richten sich automatisch nach den Einträgen, die Sie der Lizenzliste hinzugefügt haben.

Symbol	Beschreibung
	schließt das gewählte Element aus der Verschlüsselung aus.
	verschlüsselt das gewählte Element mit AxProtector (Lizenzlisten ID mit einem Wert von 0, d.h. der Default-Lizenz).
	verschlüsselt das gewählte Element mit IxProtector (Lizenzlisten ID mit einem Wert von ungleich 0, d.h. nach vorhandenen Lizenzlisten-Einträgen).
	zeigt an, dass diese Methode aufgrund der Größe von der Verschlüsselung ausgenommen wird. Die Schwelle für die Methodengröße können Sie auf der Seite 'Erweiterte Einstellungen' unter Optimierung setzen.

Die Änderungen werden sofort im linken Bereich angezeigt.

7.4.2.9 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.



Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc. Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`.

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "Datei – wbc-Datei exportieren" Menü-Eintrag erstellen.

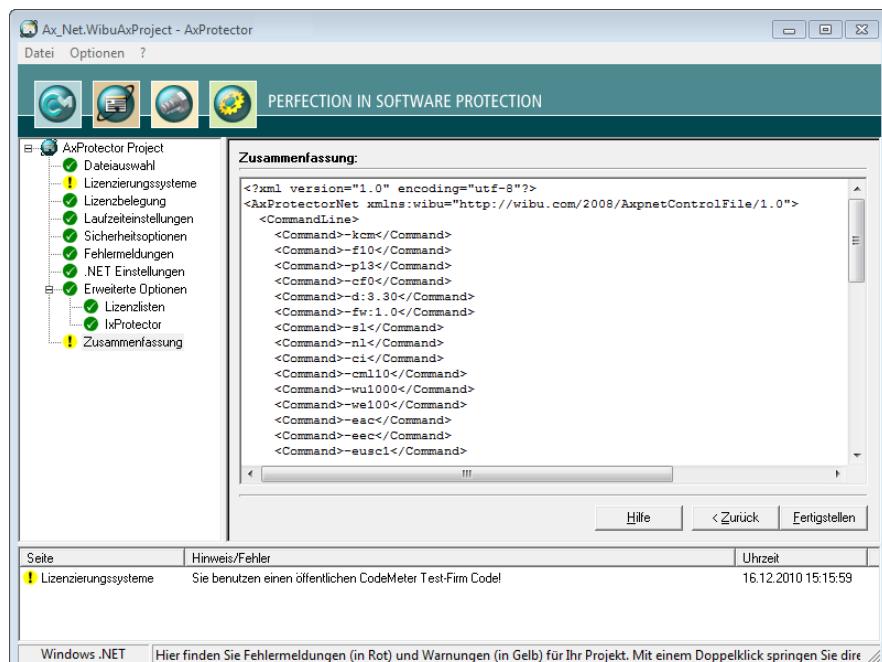


Abbildung 54: AxProtector - .NET "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

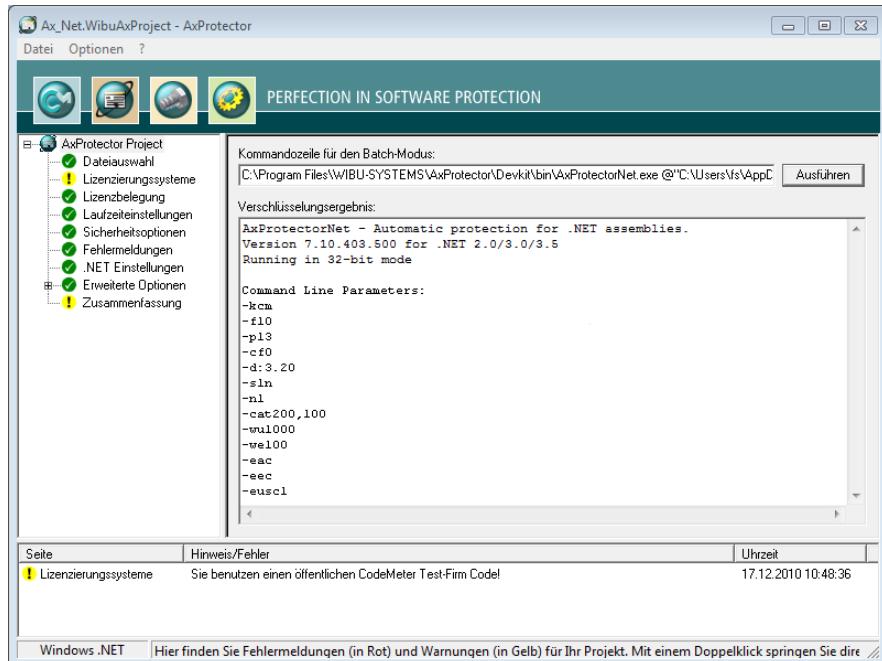


Abbildung 55: AxProtector - .NET "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die " Ausführen " Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.  Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen.

7.4.3 Mac OS X Anwendung oder Dylib

Zu verschlüsselnde Anwendungen für diesen Projekttyp umfassen Mac OS X Anwendungen ab der Version 10.4. Ab der Version 8.20 werden nur noch Applikationen geschützt, die für Mac OS X 10.5 und neuer erstellt wurden.

Sie können die Anwendungen entweder über die [AxProtector GUI](#)¹⁴¹ oder die [Kommandozeile](#)¹²⁹² verschlüsseln. In der Kommandozeilen-Variante für Mac OS X können Sie die [Verschlüsselungsparameter](#)¹⁶⁷ auch in die Kommandozeile einfügen.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Mac OS X mit AxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Mac OS X Anwendung oder Dylib	 AxProtector Mac ¹⁴¹	✓	Windows Kommandozeile ¹²⁹² In einer separaten Kommandozeile für Mac, die auf Mac OS X-Betriebssystemen läuft, können Sie ebenfalls Verschlüsselungsparameter ¹⁶⁷ eingeben.

7.4.3.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

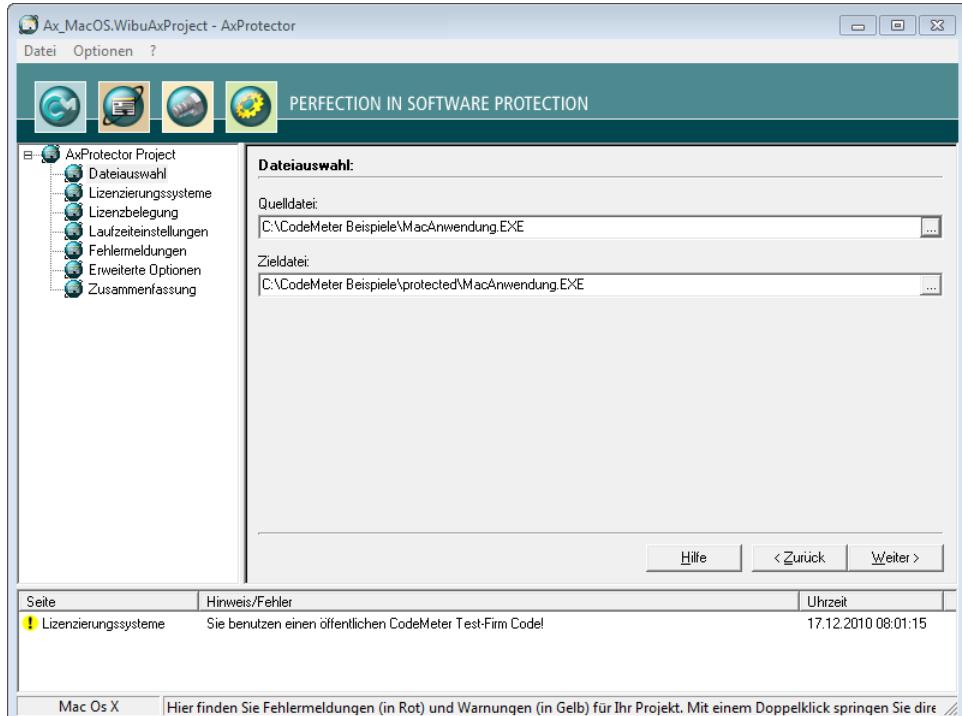


Abbildung 56: AxProtector - Mac OS X "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein.  Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..\protected\..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.4.3.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Datei nehmen Sie hier Einstellungen zum verwendeten Lizenzierungssystem *CmDongle* und / oder *CmActLicense* vor.

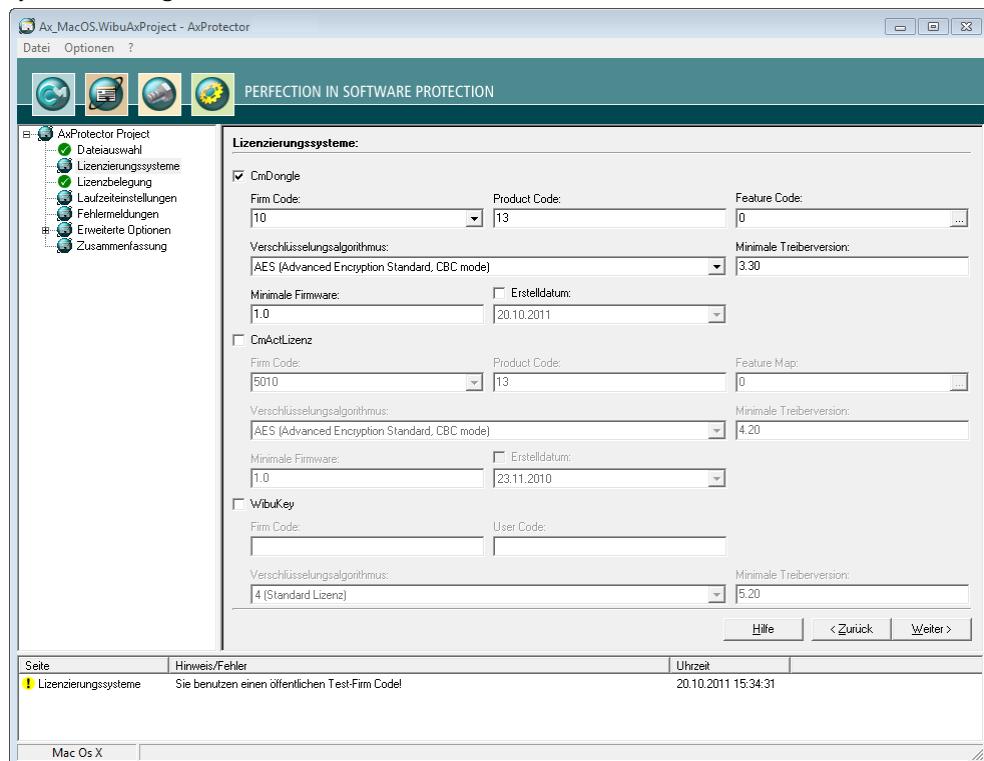


Abbildung 57: AxProtector - Mac OS X "Lizenzierungssysteme"

Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenzierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenzierungssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die Wibu-Systems Internetseiten.

Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich (siehe Kommandozeilen-Option [hier](#)²⁹³):

Element	Beschreibung
Firm Code	<p>Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird.</p> <p> Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle</i> Evaluation-Firm Code des <i>CodeMeter®</i> Software Development Kits (SDK) und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010. Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bewirkt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eingeben.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. <i>CodeMeter®</i> unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Minimale Treiberversion	<p>Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i> an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt <i>AxProtector</i> automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizenzen.</p> <p> Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen beim Starten Ihrer geschützten Software.</p>

Element	Beschreibung
Erstelldatum	<p>Kommandozeilen-Option siehe hier²⁹⁴.</p> <p>Ab der Firmware-Version 1.18 unterstützt <i>CodeMeter</i>[®] die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden.</p> <p>Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;">  Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können. <p>Beachten Sie auch, dass Sie hier das Kontrollkästchen aktivieren müssen, um Überprüfungsoptionen des Wartungszeitraumes (Maintenance Period) im Dialog zu den erweiterten Laufzeiteinstellungen¹⁵⁰ vornehmen zu können.</p> </div>
Minimale Firmware	<p>Kommandozeilen-Option siehe hier²⁹⁴.</p> <p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem *WibuKey* informiert separat das *WibuKey* Entwicklerhandbuch.

7.4.3.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Anwendung vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

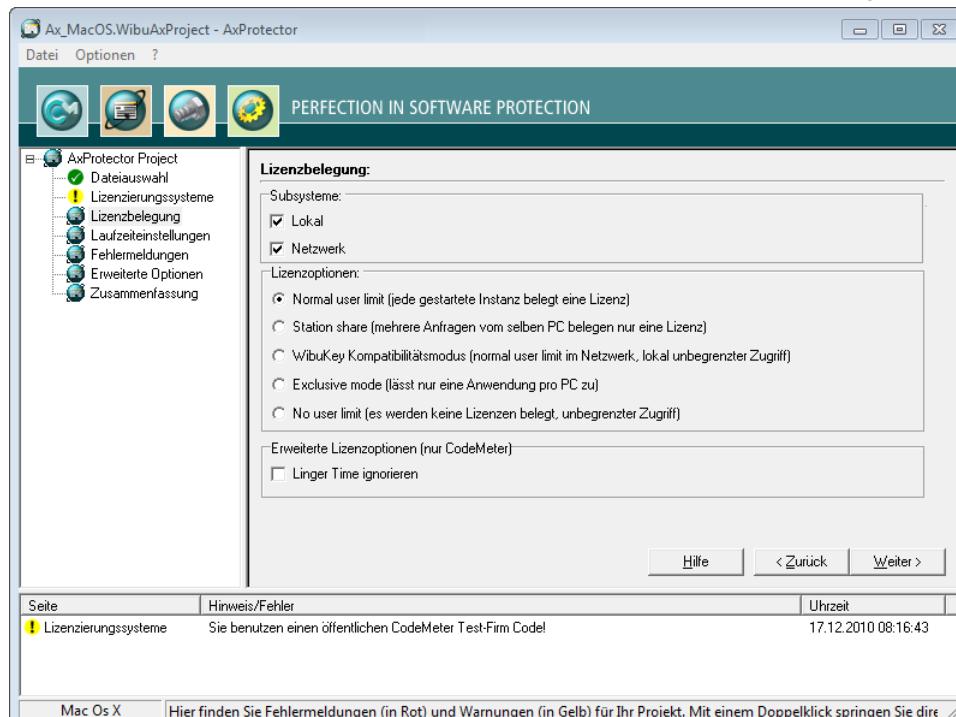


Abbildung 59: AxProtector - Mac OS X "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#)²⁹⁴).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der CodeMeter Lizenzserver mit einem aktivierte Netwerkzugriff läuft. i Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizenzoptionen

Im Bereich Lizenzoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und

die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier²⁰⁵](#)).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	<p>Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz.</p> <p> Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.</p>
WibuKey Kompatibilitäts-Modus	<p>Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).</p> <p> Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu WibuKey. WibuSystems empfiehlt die Einstellungen 'Normal user limit' und 'Station Share'.</p>
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizenzen belegt werden. Belegte Lizenzen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	<p>Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren.</p> <p>Mit dieser Lizenzenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).</p>

7.4.3.4 Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie das Verhalten der Anwendung zur Laufzeit fest, z.B. Abfrage der Lizenz in *CmContainern*, Ausgabe von Warnmeldung, etc..

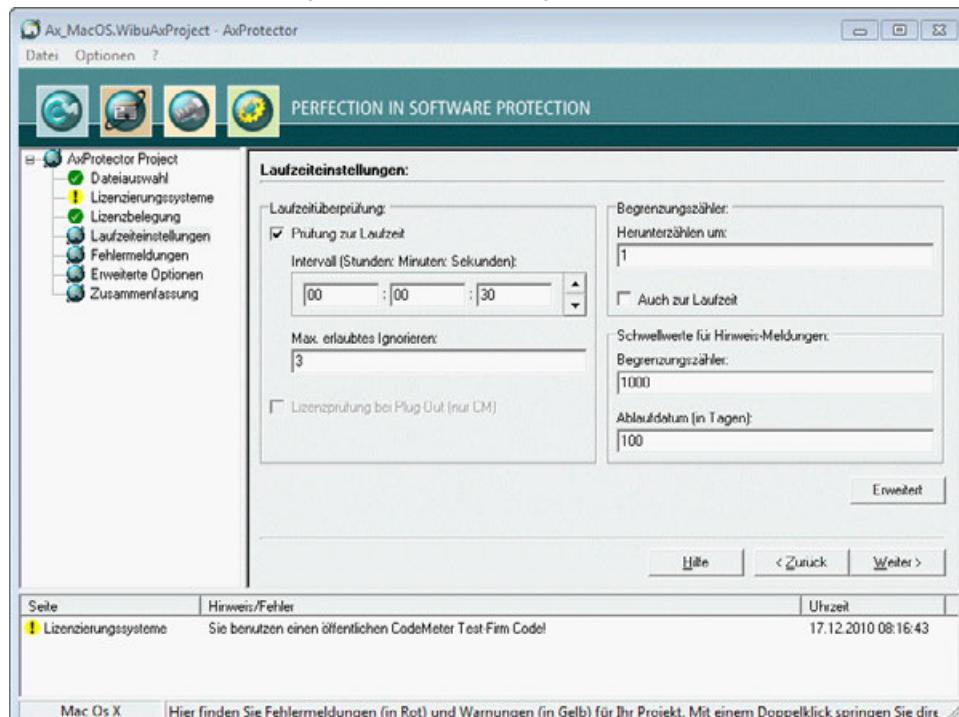


Abbildung 60: AxProtector - Mac OS X "Laufzeiteinstellungen"

Laufzeitüberprüfung

In diesem Bereich können Sie definieren, ob und wie oft die geschützte Anwendung die Lizenz während der Laufzeit überprüft.

Elemente	Beschreibung
Prüfung zur Laufzeit	Aktiviert oder deaktiviert die Überprüfung während der Laufzeit der geschützten Anwendung. Kommandozeilen-Option siehe hier .
Intervall	Legt das Intervall zwischen zwei Überprüfungen fest. Angabe im Format Stunden: Minuten: Sekunden.
Max. erlaubtes Ignorieren	Gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann. Schlägt die Verbindung zum <i>CmContainer</i> fehl, d.h. kann nicht mehr auf die Lizenz zugriffen werden, geben Sie dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit, auch ohne die Lizenz noch weiterzuarbeiten.
Lizenzprüfung bei	Beendet die geschützte Anwendung, wenn der <i>CmDongle</i> während der Ausführung abgezogen

Elemente	Beschreibung
Plug-Out (nur CmDongle)	wird und eine sofortige Fehlermeldung wird ausgegeben. Kommandozeilen-Option siehe hier ²⁹⁷ .

Begrenzungszähler

Begrenzungszähler (Unit Counter) können u.a. dazu dienen, die Gültigkeit von Lizzenzen in einem *CmContainer* festzustellen. In diesem Bereich können Sie dieses Verhalten definieren (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Herunterzählen um	Gibt den Wert an, um den der Begrenzungszähler (Unit Counter) heruntergezählt wird. Diese Option bewirkt das Herunterzählen des Zählers beim Start der geschützten Anwendung. Ist die "Auch zur Laufzeit" Option aktiviert und sind die Einträge wie in der obigen Abbildung dargestellt gesetzt, wird alle 30 Sekunden (siehe das festgelegt Intervall) ein gesetzter Begrenzungszähler (Unit Counter) um den Wert 1 heruntergezählt.
Auch zur Laufzeit	Zählt den Begrenzungszähler (Unit Counter) auch während der Laufzeit der geschützten Anwendung herunter.  Diese Option greift nur, wenn die "Prüfung zu Laufzeit" Option im Bereich "Laufzeitüberprüfung" aktiviert ist.

Schwellenwerte für Hinweismeldungen

In diesem Bereich können Sie definieren, wann eine Hinweismeldung zur Gültigkeit der Lizenz ausgegeben wird.

	Zur individuellen Gestaltung des Textes der Hinweismeldungen siehe hier ¹⁵² .
--	--

Element	Beschreibung
Begrenzungszähler	Wird der angegebene Schwellenwert unterschritten, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .
Ablaufdatum (in Tagen)	Wird das angegebene Ablaufdatum in Tagen innerhalb der vorgegebenen Schwelle erreicht, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .

7.4.3.4.1 Erweiterte Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie zusätzliche Einstellungen zur Laufzeit der verschlüsselten Anwendung fest.

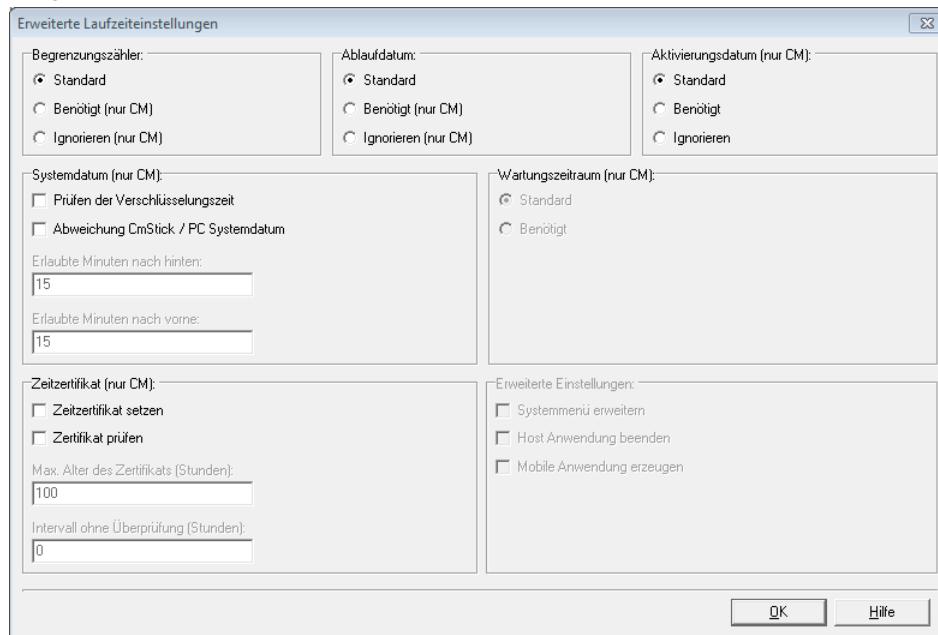


Abbildung 61: AxProtector - Mac OS X "Erweiterte Laufzeiteinstellungen"

Für die Abfrage der in die Lizenz eingetragenen Optionen Begrenzungszähler (Unit Counter), Ablaufdatum (Expiration Time) und Aktivierungsdatum (Activation Time) gilt die folgende Handhabung.

Status	Standard	Benötigt	Ignorieren
= 0	X	X	✓
< > 0	✓	✓	✓
nicht angegeben	✓	✓	✓

Begrenzungszähler (Unit Counter)

Definiert die Handhabung eines Unit Counter (Begrenzungszählers), der in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Standard	Zählt einen vorhandenen Unit Counter-Eintrag in der Lizenz beim Start und/oder zur Laufzeit um den auf der vorherigen Seite definierten Wert herunter. Wenn der Unit Counter Null erreicht startet die verschlüsselte Anwendung nicht.
Benötigt	Ein Unit Counter-Eintrag < > 0 in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag star-

Element	Beschreibung
	tet die verschlüsselte Anwendung nicht.
Ignorieren	Ein vorhandener Unit Counter-Eintrag in der Lizenz wird ignoriert. Die Anwendung setzt den Unit Counter nicht herunter. Die Anwendung startet auch bei einem Unit Counter-Eintrag = 0.

Ablaufdatum (Expiration Time)

Definiert die Handhabung einer Expiration Time (Ablaufdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Expiration Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn keine Expiration Time vorhanden ist, oder das aktuelle Datum vor der Expiration Time liegt.
Benötigt	Ein Expiration Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten.
Ignorieren	Ein vorhandener Expiration Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum nach der Expiration Time liegt.

Aktivierungsdatum (Activation Time)

Definiert die Handhabung einer Activation Time (Aktivierungsdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Activation Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn kein solcher Eintrag vorhanden ist, oder die zertifizierte Zeit ⁴¹⁷ nach der Activation Time liegt.
Benötigt	Ein Activation Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten. Beachten Sie, dass dann eine Internet-Verbindung zum Abholen der zertifizierten Zeit erforderlich ist.
Ignorieren	Ein vorhandener Activation Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum vor der Activation Time liegt.

Wartungszeitraum (Maintenance Period)

Definiert die Handhabung eines Wartungszeitraumes (Maintenance Period), der in der Lizenz eingetragen ist. Eine Lizenz berechtigt dann zur Verwendung aller Softwareversionen, die innerhalb des definierten Wartungszeitraumes (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der Applikation hinterlegt und zur Laufzeit der geschützten Anwendung geprüft, ob das Erstelldatum (Release Date) innerhalb des Wartungszeitraumes (Maintenance Period) liegt (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

 Die Optionen sind nur auswählbar, wenn auf der Seite "Lizenzerierungssysteme" das Erstelldatum (Release Date) [aktiviert](#)¹¹⁵ worden ist.

Es bestehen zwei Überprüfungsoptionen:

Element	Beschreibung
Standard	Während der Laufzeit der geschützten Anwendung wird gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist. Dies entspricht der Standardeinstellung

Element	Beschreibung
	auch wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) nicht aktiviert ¹⁴⁴ worden ist.
Benötigt	Während der Laufzeit der geschützten Anwendung ist das Prüfen des Wartungszeitraumes (Maintenance Period) gegen das Erstelldatum (Release Date) zwingend erforderlich. Die PIO Wartungszeitraum (Maintenance Period) muss vorhanden sein.

Zeitzertifikat

In jedem *CmContainer* ist eine laufende Uhr integriert, die läuft, wenn der *CmContainer* mit dem Rechner verbunden ist. Die Uhrzeit synchronisiert sich dabei beim Aktivieren des *CmContainers* nach vorne und nutzt ansonsten die letzte gespeicherte Zeit.

Wenn gewünscht, kann die zertifizierte Uhrzeit durch die Synchronisation mit dem *CodeMeter®* Zeitserver aktualisiert werden. Die Zeitserver sind von Wibu-Systems bereitgestellte Rechner, die über die Welt verteilt sind und eine zertifizierte Zeit zur Verfügung stellen. Bei einer Aktualisierung der zertifizierten Uhrzeit wird die interne *CmContainer*-Zeit synchronisiert (Kommandozeilen-Option siehe [hier](#)¹⁴⁵).

	Für Informationen zur Manipulationssicherheit von Aktivierungs- und Ablaufdatum siehe hier ¹⁴⁷
---	---

Element	Beschreibung
Zeitzertifikat setzen	Mit dieser Option wird versucht die zertifizierte Zeit im <i>CmContainer</i> zu aktualisieren. Die zertifizierte Zeit wird beim Zeitserver angefordert.  Diese Option erfordert eine Internet-Verbindung.
Zertifikat prüfen	Diese Option überprüft, ob die zertifizierte Zeit älter ist, als das hier festlegbare maximale Alter. Ist das maximale Alter des Zeitzertifikats überschritten, so lässt sich die Anwendung nicht starten.
Max. Alter des Zertifikats (in Stunden)	Bei ausgewählter "Prüfung" des Zeitzertifikats können Sie hier das maximale Alter des Zertifikats in Stunden angeben. Das Alter des Zertifikates berechnet sich aus der Differenz der laufenden System-Zeit und der zertifizierten Zeit.
Intervall ohne Überprüfung (Stunden)	Gibt an, innerhalb welchen Intervalls keine Überprüfung des Zeitzertifikats stattfindet. Ist dieses Intervall noch nicht erreicht, findet keine Überprüfung statt. Befindet sich das Zeitzertifikat zwischen diesem Intervall und dem max. Alter des Zertifikats, wird versucht, das Zeitzertifikat zu aktualisieren. Gelingt dies nicht, läuft die Anwendung jedoch bis zum Erreichen des max. Alters des Zeitzertifikats weiter. Erst danach ist zwingend ein aktualisiertes Zeitzertifikat notwendig.

System Datum

In diesem Bereich nehmen Sie Einstellungen vor, die dem zusätzlichen Schutz dienen, eine Lizenz über ein bewusstes Falschstellen der PC-Zeit zu manipulieren (Kommandozeilen-Option siehe [hier](#)¹⁴⁹).

Element	Beschreibung
Prüfen der Verschlüsselungszeit	Diese Option speichert die Verschlüsselungszeit (PC Time) in der geschützten Anwendung. Die Anwendung läuft auf dem Kunden-PC dann nur, wenn die <i>CmContainer</i> Systemzeit neuer ist als die Verschlüsselungszeit.

Element	Beschreibung
	 Erfordert mindestens <i>CodeMeter® 4.10</i> .
Abweichung CmContainer / PC Systemzeit	Wird diese Option aktiviert, ist die Festlegung eines Zeitkorridors möglich, innerhalb dessen sich die Abweichung zwischen <i>CmContainer</i> Systemzeit und der PC-Zeit bewegen darf. Wird dieser unter- bzw. überschritten läuft die geschützte Anwendung auf dem Kunden-PC nicht.
Erlaubte Minuten nach hinten	Gibt in Minuten an, um wieviele Minuten die PC Zeit älter als die <i>CmContainer</i> Systemzeit sein darf.
Erlaubte Minuten nach vorne	Gibt in Minuten an, um wieviele Minuten die PC Zeit vor der <i>CmContainer</i> Systemzeit liegen darf.

7.4.3.5 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob Sie entweder eine eigene angepasste Fehlerausgabe verwenden, oder ob Standard-Hinweisfenster angezeigt werden sollen.

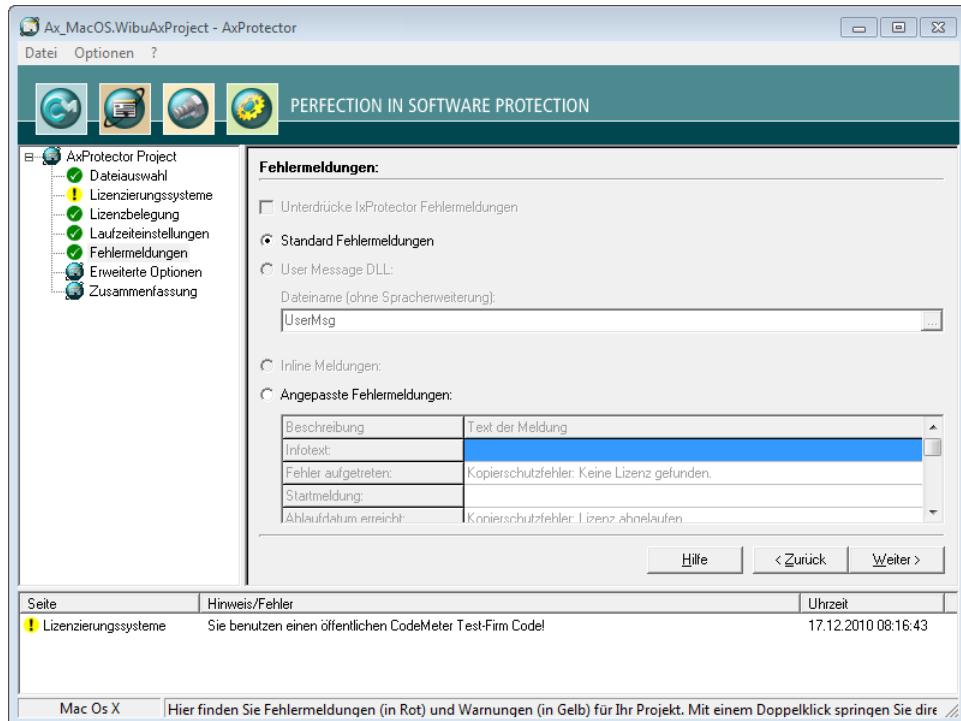


Abbildung 62: AxProtector - Mac OS X "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.4.3.6 Sicherheitsoptionen

Über diese Seite treffen Sie eine Auswahl aus verschiedenen Schutzmechanismen und -methoden für Ihre Anwendung. Sie können hier den Grad der Sicherheit selbst skalieren. Dabei legen Sie z.B. selbst fest, wie intensiv nach Debuggern gesucht werden soll, bis hin zum Sperren des *CmContainers*.

 Sollten sich die gesetzten Optionen inkompatibel zu Ihrer geschützten Anwendung verhalten, so können die einzelnen Sicherheitsoptionen auch einzeln deaktiviert werden.

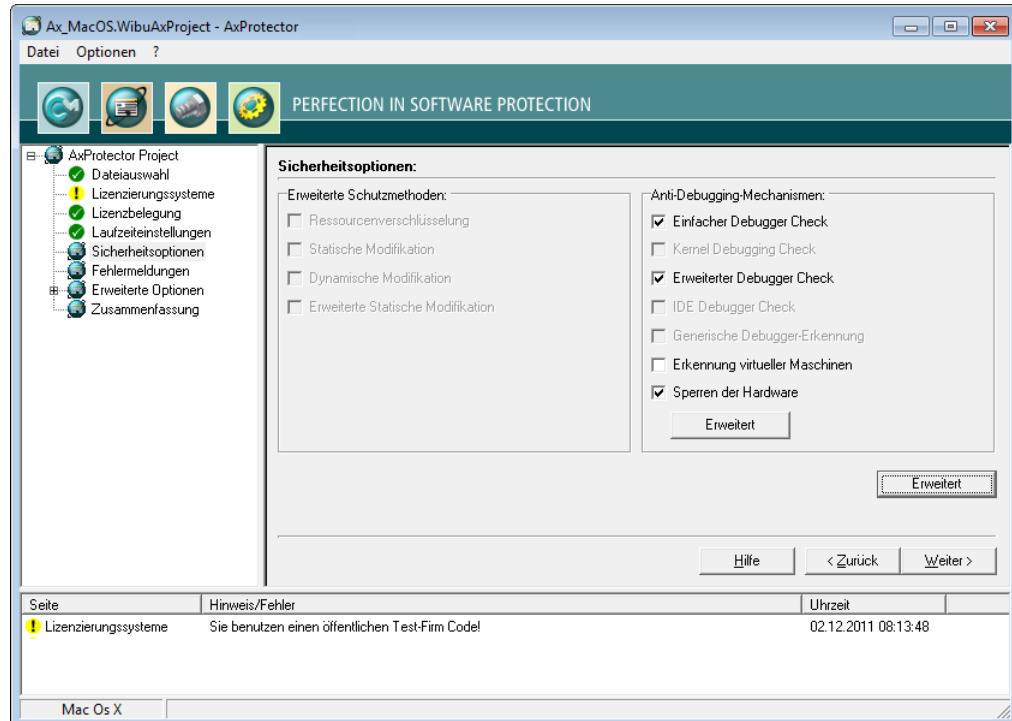
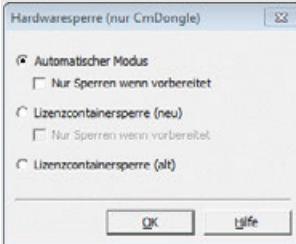


Abbildung 63: AxProtector - Mac OS "Sicherheitsoptionen"

Anti-Debugging Mechanismen

Debugger-Programme dienen der Fehlersuche und Fehlerbeseitigung, können aber auch von Hackern zur Analyse der Software verwendet werden. In diesem Bereich legen Sie die Optionen fest, wie auf Debugger-Programme reagiert werden soll (Kommandozeilen-Option siehe [hier](#)²⁹⁷).

Element	Beschreibung
Einfacher Debugger Check	Überprüft ob ein Debugger an Ihre Anwendung angehängt (attached) ist. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.
Erweiterter Debugger Check	Überprüft in einer erweiterten Suche auf Debugger-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.
Virtuelle Maschinen	Erkennt, ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies.
Sperren des Lizenz-Zugriffs	Mit dieser Option kann das genutzte Firm Item im CmContainer gesperrt werden sobald ein Debugger-Programm entdeckt wird. Wird die Option aktiviert, werden die Einstellungen übernommen, die Sie in einem Dialog setzen, der sich über die "Konfiguration"-Schaltfläche öffnet.

Element	Beschreibung												
	<p> Diese Schaltfläche ist nur aktiviert für CodeMeter.</p>												
Konfiguration	<p>Wenn die Option "Sperren des Lizenz-Zugriffs" aktiviert wird, können Sie über die "Konfiguration"-Schaltfläche im folgenden Dialog weitere Einstellungen vornehmen: Der Dialog erlaubt in Abhängigkeit von der Firmware, die bei der Verschlüsselung verwendet wird, die Auswahl verschiedener Sperrszenarien.</p> <table border="1"> <thead> <tr> <th>Sperrszenario</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>sofortiges Sperren</td><td>erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.</td></tr> <tr> <td>vorbereitetes Sperren</td><td>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperrre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</td></tr> </tbody> </table>  <p>Abbildung 64: AxProtector - Mac OS "Sicherheitsoptionen - Hardware-Sperre" Die folgenden Einstellungen sind verfügbar</p> <table border="1"> <thead> <tr> <th>Einstellung</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)</td><td>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt. Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items. Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</td></tr> <tr> <td>"Automatischer Modus" markiert und Kontrollkästchen</td><td>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.</td></tr> </tbody> </table>	Sperrszenario	Beschreibung	sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.	vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperrre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.	Einstellung	Beschreibung	"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)	Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt. Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items. Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.	"Automatischer Modus" markiert und Kontrollkästchen	Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.
Sperrszenario	Beschreibung												
sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.												
vorbereitetes Sperren	erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt. Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperrre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.												
Einstellung	Beschreibung												
"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)	Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt. Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items. Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.												
"Automatischer Modus" markiert und Kontrollkästchen	Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.												

Element	Beschreibung	
	Einstellung	Beschreibung
	"Nur Sperren wenn vorbereitet" aktiviert	Ist die Firmware 1.14 und höher, dann wird gleichzeitig geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.
	"Lizenzcontainersperre (neu)" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert	Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Dies ist sicherheitstechnisch gesehen die empfohlene Einstellung. Voraussetzung ist jedoch, dass alle <i>CmContainer</i> im Feld mit einer Firmware Version 1.14 und höher ausgestattet sind.
	"Lizenzcontainersperre (neu)" markiert und das Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert	Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Gleichzeitig wird geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.
	Option "Lizenzcontainersperre (alt)" markiert	Gilt für alle Firmware-Versionen. Ist ein vorbereitetes Sperren programmiert, wird der FAC um den Wert 1 heruntergezählt.

7.4.3.6.1 Erweiterte Sicherheitsoptionen

Ermöglicht die Auswahl zusätzlicher Sicherheitseinstellungen.



Abbildung 65: AxProtector - Mac OS "Erweiterte Sicherheitsoptionen"

Erweiterte Einstellungen

Dieser Bereich lässt die Auswahl weitere Optionen zu.

Element	Beschreibung
Virusprüfung hinzufügen	Der geschützten Anwendung wird eine Virenprüfung über eine Prüfsumme hinzugefügt (Kommandozeilen-Option siehe hier ³⁰¹).
API statisch linken	Das <i>CodeMeter Kern-API</i> wird statisch zur geschützten Anwendung hinzugelinkt. Diese Option erhöht die Sicherheit, sie vergrößert jedoch auch die ausführbare Datei (Kommandozeilen-Option siehe hier ³⁰²).
Zu verschlüsselnder Code	Hier kann die Menge des zu verschlüsselnden Codes (in %) angegeben werden (Kommandozeilen-Option siehe hier ³⁰⁰).

Element	Beschreibung
(in %)	

7.4.3.7 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

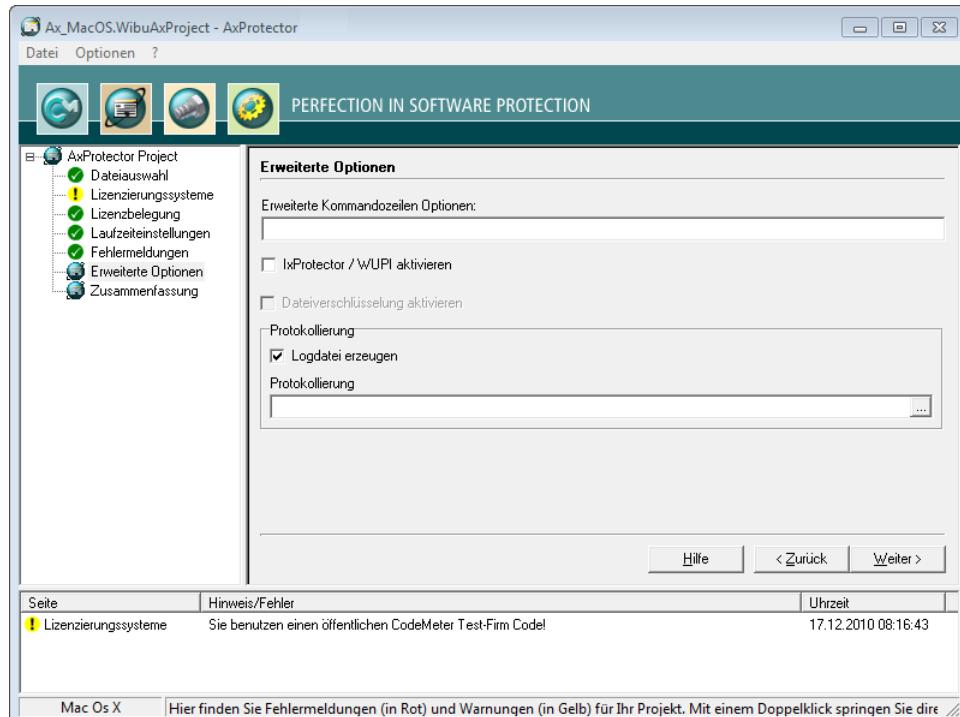


Abbildung 66: AxProtector - Mac OS X "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung. </div>
lxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit lxProtector über das Softwareschutz-API (WUP) verwenden (Kommandozeilen-Optionen siehe hier).

Element	Beschreibung
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.  Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin abgelegt.

7.4.3.7.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *lXProtector* über das [Softwareschutz-API \(WUPI\)](#)  ³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.



Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)  ³²² zur Identifizierung der Lizenz benötigen.

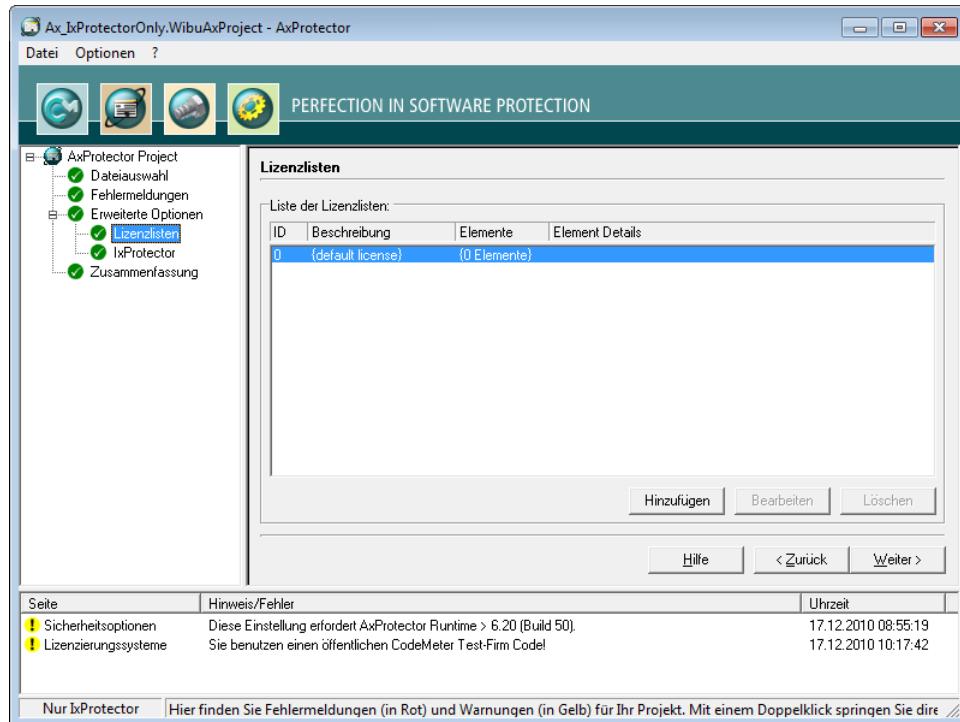
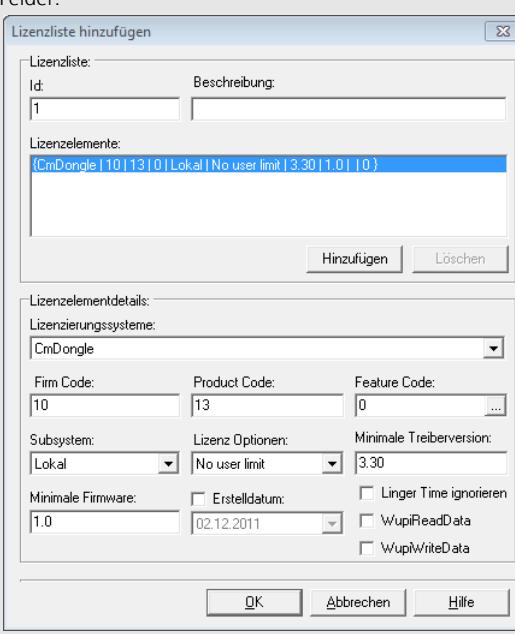


Abbildung 67: AxProtector - Mac OS "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **ID** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.  Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.

Element	Beschreibung
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag.</p> <p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p>  <p>The dialog box has two main sections:</p> <ul style="list-style-type: none"> Lizenzziste: Shows a table with one row: Id: 1 and Beschreibung: (CmDongle 10 13 0 Lokal No user limit 3.30 1.0 0). Lizenzelementdetails: Contains fields for: <ul style="list-style-type: none"> Lizenzierungsysteme: CmDongle Firm Code: 10, Product Code: 13, Feature Code: 0 Subsystem: Lokal, Lizenz Optionen: No user limit, Minimale Treiberversion: 3.30 Minimale Firmware: 1.0, Erstelltdatum: 02.12.2011 Checkboxes: Linger Time ignorieren, WupiReadData, WupiWriteData
	Abbildung 68: AxProtector -Mac OS "Lizenzen hinzufügen"
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (CmDongle, CmActLicense oder WibuKey).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	<p>Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt.</p> <p>Über die "... " Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal) .</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenzen:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt <i>CodeMeter®</i> die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzeigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

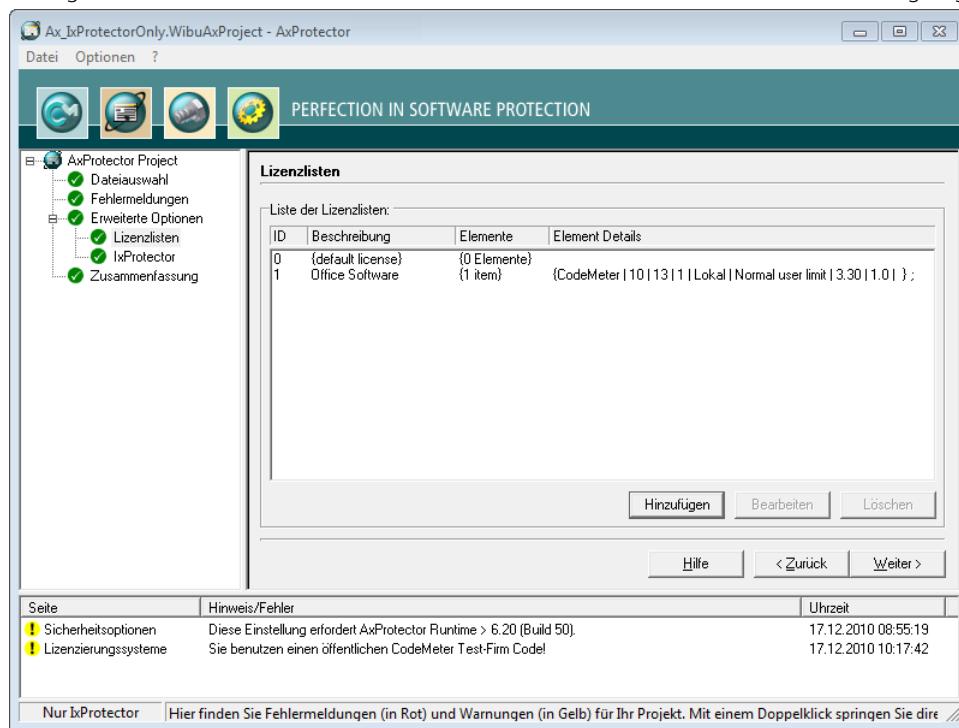


Abbildung 69: AxProtector - Mac OS "ausgefüllte Lizenzliste"

7.4.3.7.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

 In diesem Fall werden *CodeMeter®* und *WibuKey API*-Aufrufe über die dynamische Bibliothek (*.d11) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

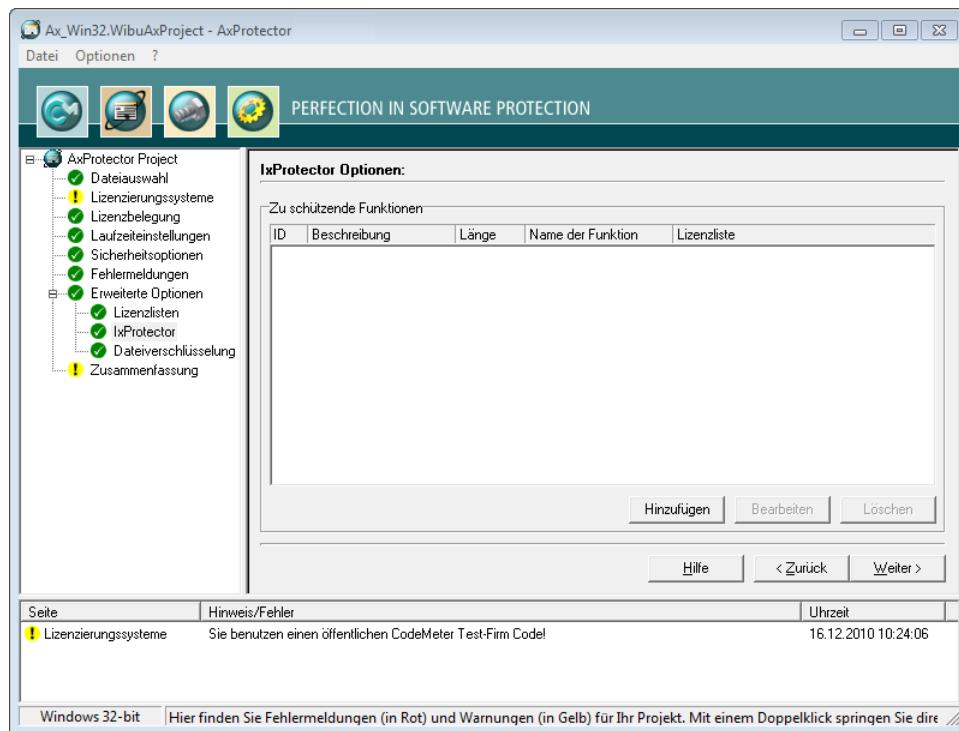
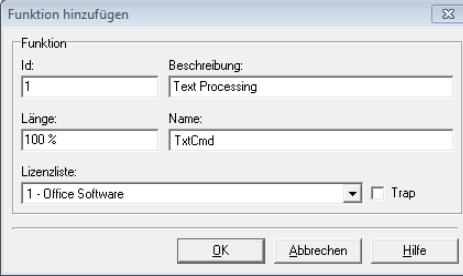


Abbildung 70: AxProtector - Mac OS - Funktionsliste

Element	Beschreibung
Zu schützende Funktionen	<p>Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Betätigen Sie im Bereich IxProtector Optionen die "Hinzufügen" Schaltfläche.

Element	Beschreibung
<p>2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.</p> 	
<p>Abbildung 71: AxProtector - Mac OS – Funktion hinzufügen</p>	
Element	Beschreibung
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode³²² und WupiEncryptCode³²² verwenden.</p>
Beschreibung	Beschreibt die Funktion durch einen Texteintrag.
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an.</p> <p>Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>i Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p>
Lizenzliste	Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.
Trap	Aktiviert die Trap-Funktion für die Funktion. Kommandozeilen-Option siehe hier ³⁰⁸ .
<p>3. Betätigen Sie die "OK" Schaltfläche. Die neuen Funktionen werden der Funktionsliste hinzugefügt.</p>	

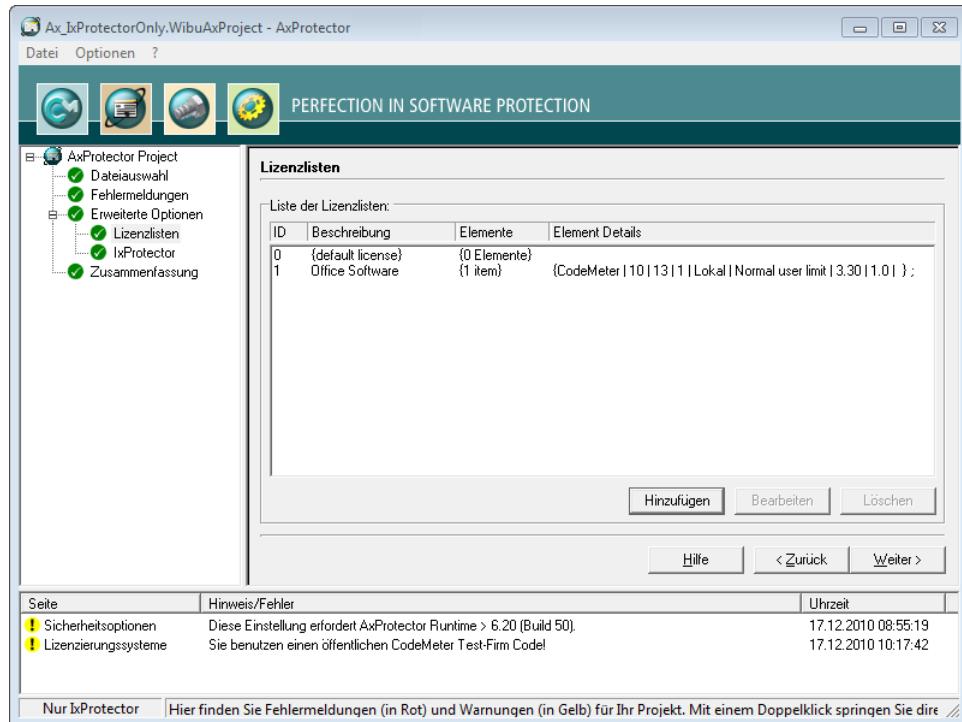


Abbildung 72: AxProtector - Mac OS "gefüllte Funktionsliste"

7.4.3.8 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf` ³¹⁹.

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

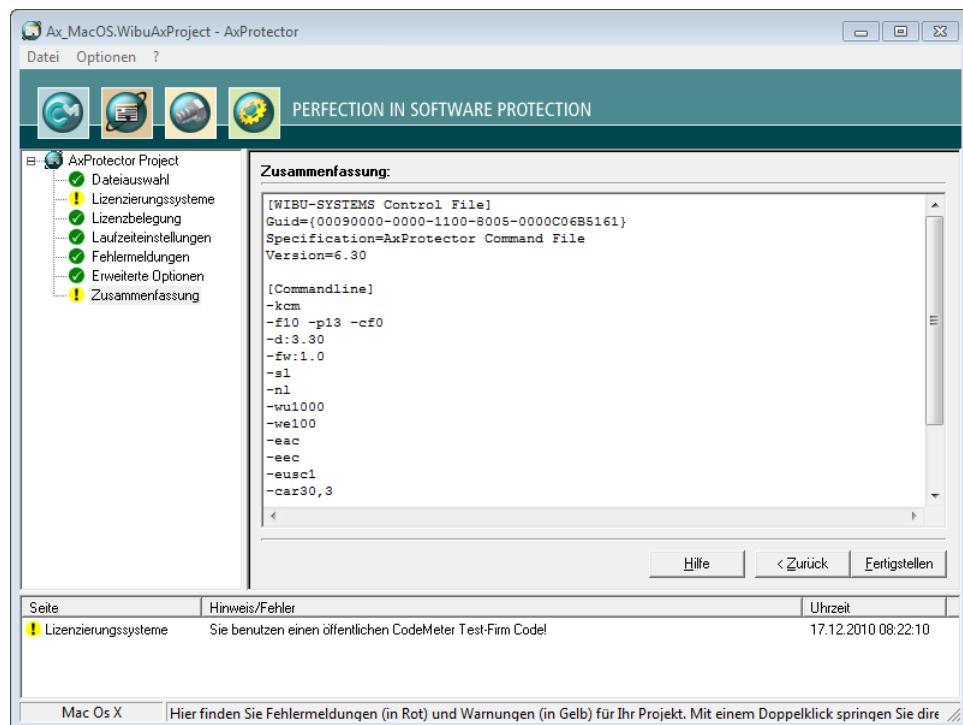


Abbildung 73: AxProtector - Mac OS X "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

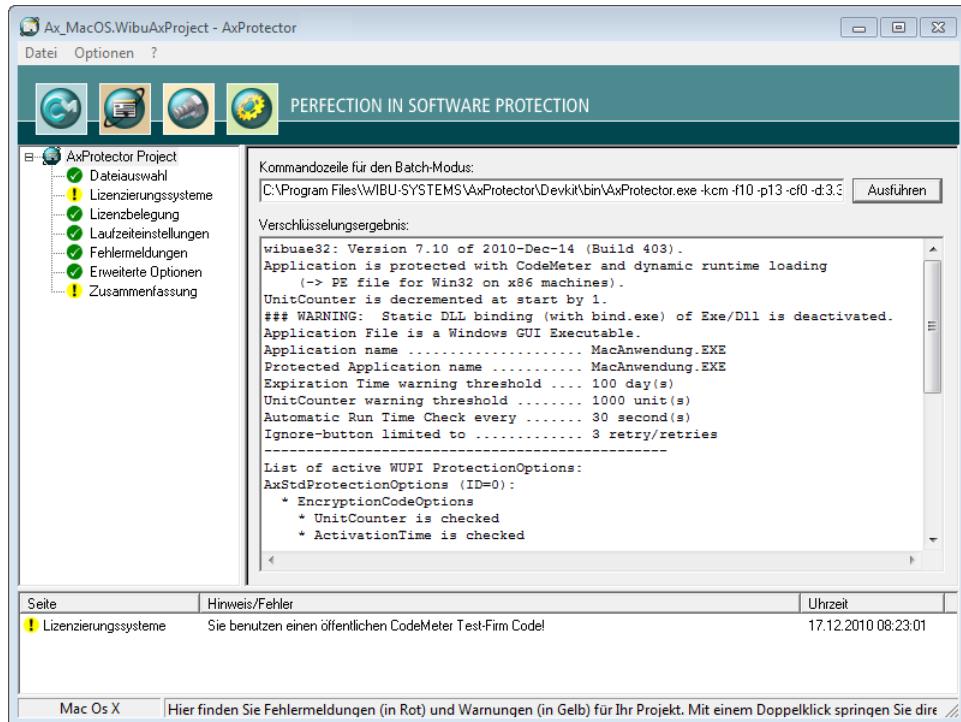


Abbildung 74: AxProtector - Mac OS X "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die " Ausführen " Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.  Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen.

7.4.4 Java Anwendung (jar-Datei)

Kompilierter Java-Code lässt sich, ähnlich wie .NET-Code, sehr einfach und ohne besondere Programmierkenntnisse zurückübersetzen in unkompliierten Quelltext. Somit ist nahezu alles, was in der Applikation passiert, prinzipiell öffentlich verfügbar und kann z.B. von der Konkurrenz einfach analysiert werden. Ihr geistiges Eigentum ist nahezu ungeschützt. Ebenso lässt sich ein eingebautes Lizenzmanagement in der Regel einfach wieder aus der Software entfernen. Daher stellt sich jedem Java-Entwickler über kurz oder lang die Frage, wie er sein geistiges Eigentum schützen und sich gegen Missbrauch absichern kann.

Der *AxProtector Java* löst diese Aufgabenstellung. Ein Java-Kompilat besteht im Wesentlichen nur aus einer Sammlung von kompliierten Klassen, von `class`-Dateien. Diese werden in der Regel zusammengebündelt als `*.jar`-Archiv ausgeliefert und abgelegt. Das Grundprinzip des *AxProtector Java* ist nun, jede Klasse einzeln zu verschlüsseln. Dazu wird automatisch das `*.jar`-Archiv entpackt, jede `class`-Datei gemäß den gemachten Einstellungen verschlüsselt, und alles hinterher, zusammen mit noch notwendigen `class`-Dateien von Wibu-Systems wieder in das Archiv eingepackt.

ClassLoader Modifikation

Beim Start der Applikation wird zunächst `SystemClassLoader` geladen, der nach und nach bei Bedarf die in der Applikation benötigten Klassen nachlädt. Und genau dort setzt der *AxProtector* an. Beim Verschlüsseln hat der *AxProtector* ebenfalls noch die Manifest-Datei modifiziert, so dass beim Start der Applikation nicht mehr wie bisher gestartet wird. Stattdessen wird als erstes vom `SystemClassLoader` über zwei Helfer-Klassen (Wrapper / Starter) ein von Wibu-Systems gelieferter `ClassLoader` geladen. Dieser `WibuClassLoader` kümmert sich um das Laden der verschlüsselten Klassen, unverschlüsselte Klassen werden weiterhin von dem original verwendeten, Java-eigenen `ClassLoader` geladen.

Entschlüsselung in nativem Code

Java selbst bietet eine Menge verschiedener Möglichkeiten, um sich unter anderem in die Ladeprozesse einzuklinken. Eine Entschlüsselung innerhalb des Java-Codes zu machen, ist daher nicht sinnvoll und schnell ausgehebelt. Im *AxProtector Java* reicht der `WibuClassLoader` daher die verschlüsselten Klassen weiter an eine native Bibliothek. In dieser `wibuXPM4J`-Bibliothek wird nun die Klasse unter Verwendung des gewählten Lizenzierungssystems (`CmDongle` / `CmActLicense` / `WibuKey`) entschlüsselt und an die native Java-Bibliothek weitergereicht. Die entschlüsselte Klasse steht nun in Java uneingeschränkt zur Verfügung.

Zusätzliche Sicherheitsmechanismen

Zusätzlich zu diesem Ladeprinzip erweitert der *AxProtector Java* die Applikation um weitere Sicherheitsmechanismen. Um sicherzustellen, dass die verwendete Lizenz auch weiterhin verfügbar ist, und nicht zum Beispiel der Dongle abgezogen wurde, kann eine regelmäßige Überprüfung zur Laufzeit durchgeführt werden. Dabei wird die verwendete Lizenz in einem konfigurierbaren Intervall erneut durch Verschlüsselungsoperationen überprüft und im Fehlerfall die Applikation angehalten.

Signatur-Überprüfung der Laufzeitumgebung

Seit Version 6 sind die Quellen von Java offen gelegt und daher könnte sich nun jeder selbst eine leicht abgewandelte Version von Java zusammenstellen und wäre damit auch in der Lage, eigenen Code in die native Java-Bibliothek einzuschleusen um das Laden der entschlüsselten Klassen zu protokollieren. Daher gibt es für Java 6 die Möglichkeit, die Authentizität der verwendeten Java-Version zu überprüfen. Dazu sind die Signaturen der nativen Java-Bibliotheken der Applikation beigelegt und werden beim Start überprüft. Wird eine neuere Version der Java-Bibliothek verwendet, merkt dies der *AxProtector* und bietet an, automatisch neue Signaturen von der Wibu-Systems Webseite herunterzuladen. Somit kann die Applikation auch mit zum Zeitpunkt der Verschlüsselung noch nicht veröffentlichten Versionen umgehen.

Voraussetzungen

Der *AxProtector Java* arbeitet ausschließlich mit dem originalen Sun Java zusammen, der üblicherweise verwendeten Variante von Java. Beim Anwender wird neben den im *.jar-Archiv beigelegten Dateien auch die oben schon erwähnte native wibuXPM4J-Bibliothek benötigt.

Diese ist für Windows und Mac im Runtime-Kit von *CodeMeter®* und *WibuKey* enthalten, für Linux gibt es kleine, separate Installer.

Bei der Verschlüsselung gibt es zusätzlich die Möglichkeit, nur bestimmte Klassen zu verschlüsseln (Whitelist) oder bestimmte Klassen von der Verschlüsselung auszunehmen (Blacklist). Somit kann man zum Beispiel Klassen anderer Hersteller von der Verschlüsselung ausnehmen. Ebenso kann eine Minimum-Version angegeben werden.

Die bisherigen Ausführungen beziehen sich auf Java-Applikationen, also eigenständige Programme, die auf der Festplatte des Anwenders liegen. Gerade bei Java sind aber die Einsatzfälle mittlerweile sehr vielfältig und man möchte beispielsweise auch Serverapplikationen schützen. Wie sieht es also nun bei der Integration von Softwareschutz zum Beispiel in den Applikationsserver Tomcat aus?

Angepasste Verwendung

Der *AxProtector Java* eignet sich genauso für den Schutz von zum Beispiel Java Servlets, Eclipse Rich Client Applikationen oder auch Java Web Start Applikationen. Bei Verwendung in diesem Umfeld müssen einige Dinge beachtet oder angepasst werden. Es gibt mittlerweile mehrere von Wibu-Systems bereitgestellte *ClassLoader*, die speziell auf die in dem jeweiligen Anwendungsfall notwendigen angepasst sind, zum Beispiel den *ServletClassLoader* oder den *EclipseClassLoader*. Fragen Sie beim Support von Wibu-Systems nach passenden Beispielen oder Hilfestellung bei der Integration.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Java mit *AxProtector* verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Java Anwendung	 AxProtector Java <small>170</small>		Windows Kommandozeile <small>292</small>

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
			In einer separaten Kommandozeile für Java, die auf Windows, Mac OS X-, Linux- und Solaris -Betriebssystemen läuft, können Sie ebenfalls Verschlüsselungsparameter ¹⁸⁹ eingeben.

7.4.4.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

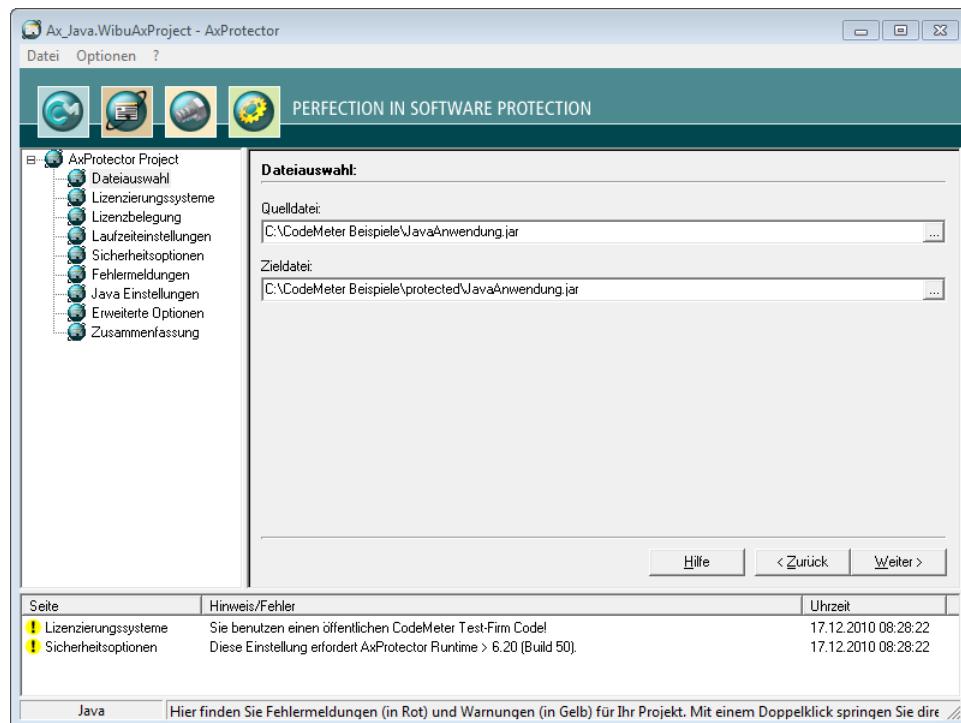


Abbildung 75: AxProtector - Java "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsseln-

Element	Beschreibung
	de Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein.  Als Alternative zur "... Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [...] Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier <small>313</small> .

7.4.4.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Datei nehmen Sie hier Einstellungen zum verwendeten Lizenzierungssystem *CodeMeter* vor.

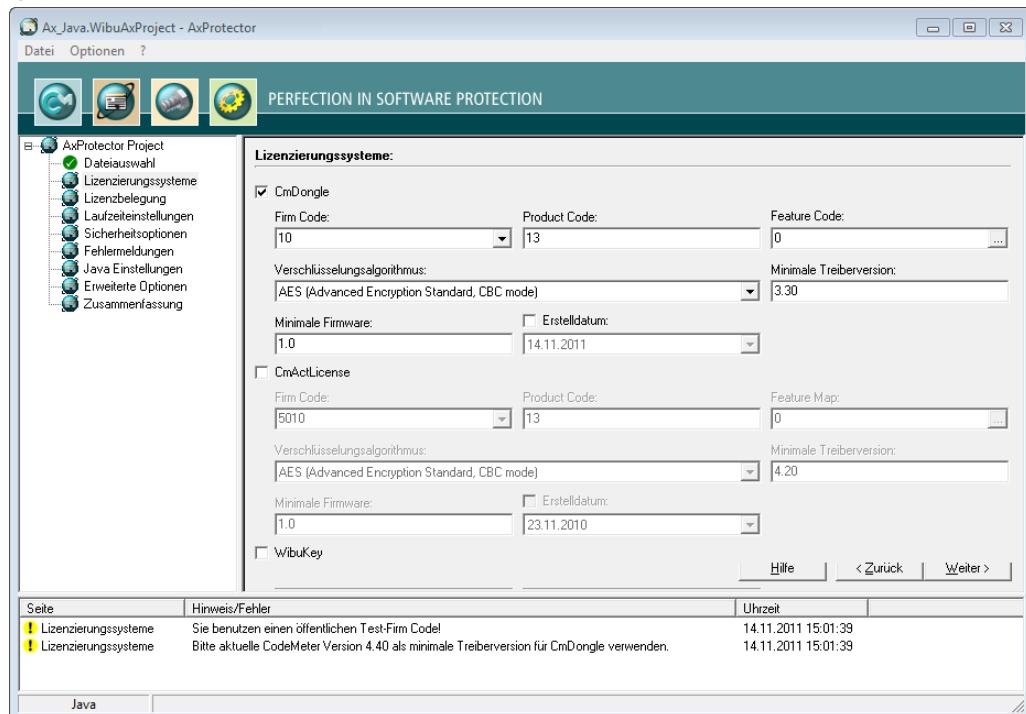


Abbildung 76: AxProtector - Java "Lizenzierungssysteme"

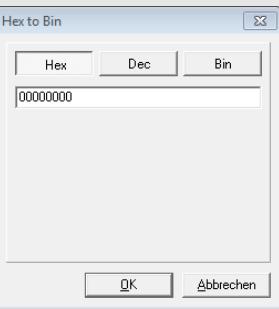
 Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenzierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit

Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenzierungssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die Wibu-Systems Internetseiten.

Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich (siehe Kommandozeilen-Option [hier](#)²⁹³):

Element	Beschreibung
Firm Code	<p>Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird.</p> <p> Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle</i> Evaluation-Firm Code des <i>CodeMeter®</i> Software Development Kits (SDK) und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010. Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bewirkt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eintragen.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. <i>CodeMeter®</i> unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Minimale Treiberversi-	Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i>

Element	Beschreibung
on	<p>an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt AxProtector automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizenzen.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen beim Starten Ihrer geschützten Software. </div>
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datums-werte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden . Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können. Beachten Sie auch, dass Sie hier das Kontrollkästchen aktivieren müssen, um Überprüfungsoptionen des Wartungszeitraums (Maintenance Period) im Dialog zu den erweiterten Laufzeiteinstellungen¹⁷⁹ vornehmen zu können. </div> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Minimale Firmware	<p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem WibuKey informiert separat das WibuKey Entwicklerhandbuch.

7.4.4.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Anwendung vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

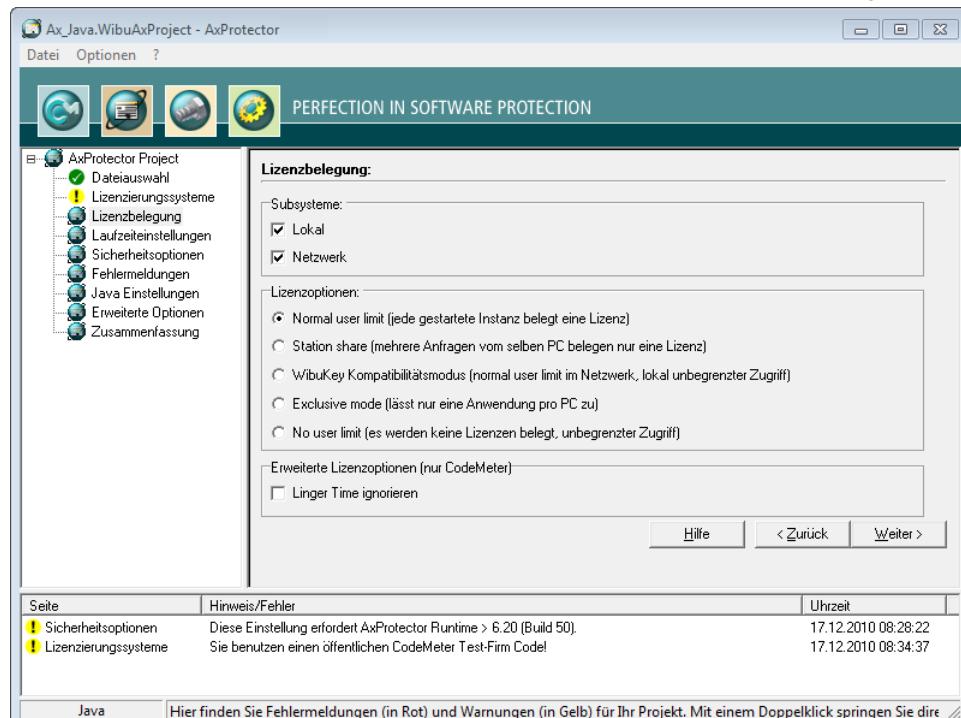


Abbildung 78: AxProtector - Java "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#) [294]).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der CodeMeter Lizenzserver mit einem aktivierte Netzwerkzugriff läuft. Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizensoptionen

Im Bereich Lizensoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und

die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier](#)²⁹⁵).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz.  Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.
WibuKey Kompatibilitäts-Modus	Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).  Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu WibuKey. Wibu-Systems empfiehlt die Einstellungen 'Normal user limit' und 'Station Share'.
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizenzen belegt werden. Belegte Lizenzen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).

7.4.4.4 Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie das Verhalten der Anwendung zur Laufzeit fest, z.B. Abfrage der Lizenz in *CmContainern*, Ausgabe von Warnmeldung, etc..

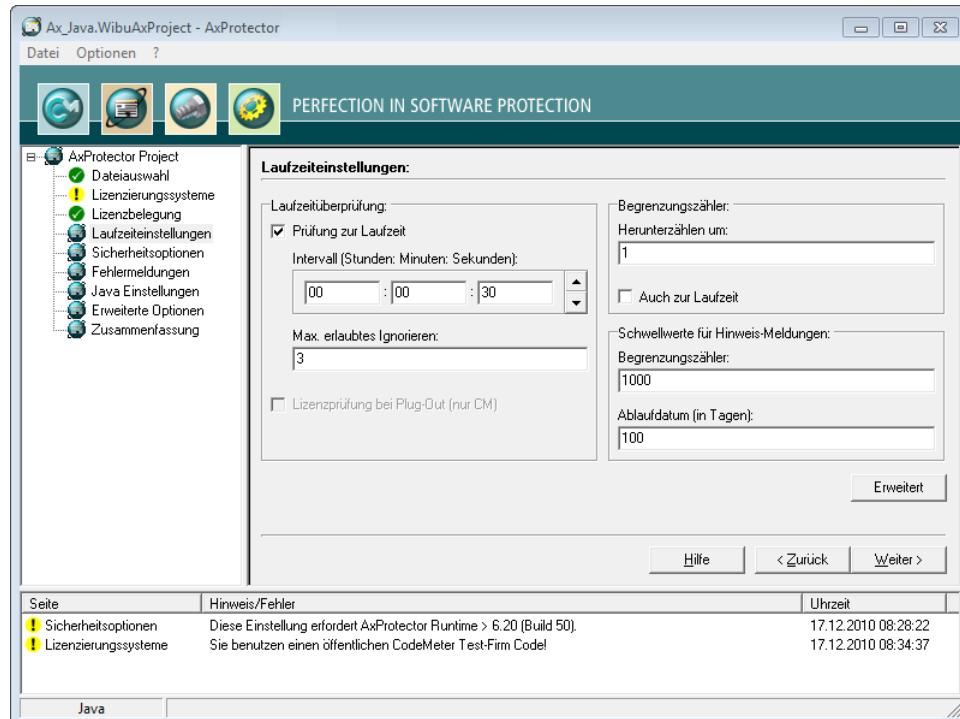


Abbildung 79: AxProtector - Java "Laufzeiteinstellungen"

Laufzeitüberprüfung

In diesem Bereich können Sie definieren, ob und wie oft die geschützte Anwendung die Lizenz während der Laufzeit überprüft.

Elemente	Beschreibung
Prüfung zur Laufzeit	Aktiviert oder deaktiviert die Überprüfung während der Laufzeit der geschützten Anwendung.
Intervall	Legt das Intervall zwischen zwei Überprüfungen fest. Angabe im Format Stunden: Minuten: Sekunden.
Max. erlaubtes Ignorieren	Gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann. Schlägt die Verbindung zum <i>CmContainer</i> fehl, d.h. kann nicht mehr auf die Lizenz zugriffen werden, geben Sie dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit, auch ohne die Lizenz noch weiterzuarbeiten.

Begrenzungszähler

Begrenzungszähler (Unit Counter) können u.a. dazu dienen, die Gültigkeit von Lizenzen in einem *CmContainer* festzustellen. In diesem Bereich können Sie dieses Verhalten definieren (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Herunterzählen um	Gibt den Wert an, um den der Begrenzungszähler (Unit Counter) heruntergezählt wird. Diese Option bewirkt das Herunterzählen des Zählers beim Start der geschützten Anwendung. Ist die "Auch zur Laufzeit" Option aktiviert und sind die Einträge wie in der obigen Abbildung dargestellt gesetzt, wird alle 30 Sekunden (siehe das festgelegt Intervall) ein gesetzter Begrenzungszähler (Unit Counter) um den Wert 1 heruntergezählt.
Auch zur Laufzeit	Zählt den Begrenzungszähler (Unit Counter) auch während der Laufzeit der geschützten Anwendung herunter.  Diese Option greift nur, wenn die "Prüfung zu Laufzeit" Option im Bereich "Laufzeit-überprüfung" aktiviert ist.

Schwellenwerte für Hinweismeldungen

In diesem Bereich können Sie definieren, wann eine Hinweismeldung zur Gültigkeit der Lizenz ausgegeben wird.

	Zur individuellen Gestaltung des Textes der Hinweismeldungen siehe hier ³⁰³ .
Element	Beschreibung
Begrenzungszähler	Wird der angegebene Schwellenwert unterschritten, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .
Ablaufdatum (in Tagen)	Wird das angegebene Ablaufdatum in Tagen innerhalb der vorgegebenen Schwelle erreicht, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .

7.4.4.4.1 Erweiterte Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie zusätzliche Einstellungen zur Laufzeit der verschlüsselten Anwendung fest.

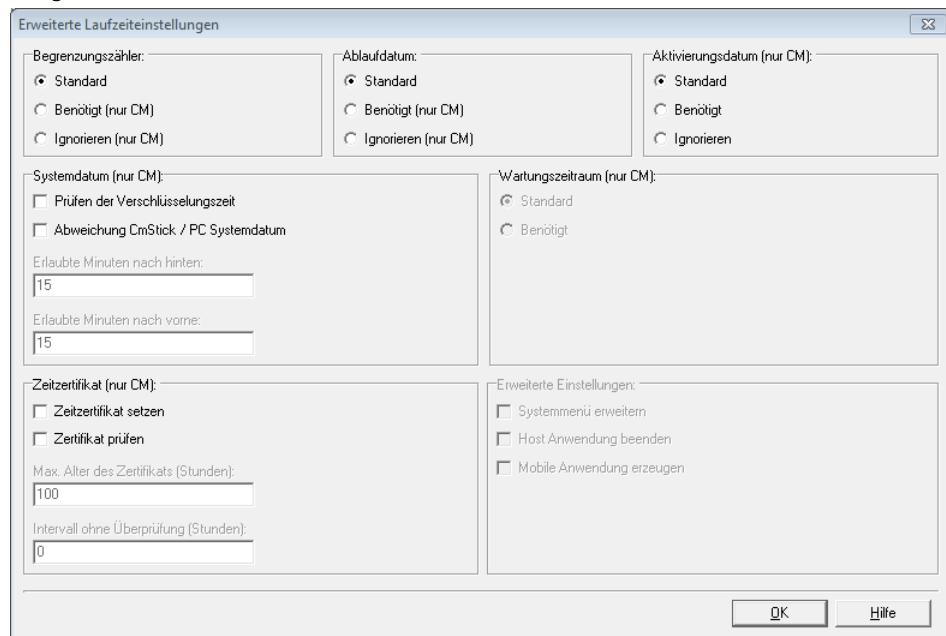


Abbildung 64: AxProtector - Java "Erweiterte Laufzeiteinstellungen"

Für die Abfrage der in die Lizenz eingetragenen Optionen Begrenzungszähler (Unit Counter), Ablaufdatum (Expiration Time) und Aktivierungsdatum (Activation Time) gilt die folgende Handhabung.

Status	Standard	Benötigt	Ignorieren
= 0	X	X	✓
< > 0	✓	✓	✓
nicht angegeben	✓	✓	✓

Begrenzungszähler (Unit Counter)

Definiert die Handhabung eines Unit Counter (Begrenzungszählers), der in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁸).

Element	Beschreibung
Standard	Zählt einen vorhandenen Unit Counter-Eintrag in der Lizenz beim Start und/oder zur Laufzeit um den auf der vorherigen Seite definierten Wert herunter. Wenn der Unit Counter Null erreicht startet die verschlüsselte Anwendung nicht.
Benötigt	Ein Unit Counter-Eintrag < > 0 in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag star-

Element	Beschreibung
	tet die verschlüsselte Anwendung nicht.
Ignorieren	Ein vorhandener Unit Counter-Eintrag in der Lizenz wird ignoriert. Die Anwendung setzt den Unit Counter nicht herunter. Die Anwendung startet auch bei einem Unit Counter-Eintrag = 0.

Ablaufdatum (Expiration Time)

Definiert die Handhabung einer Expiration Time (Ablaufdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Expiration Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn keine Expiration Time vorhanden ist, oder das aktuelle Datum vor der Expiration Time liegt.
Benötigt	Ein Expiration Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten.
Ignorieren	Ein vorhandener Expiration Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum nach der Expiration Time liegt.

Aktivierungsdatum (Activation Time)

Definiert die Handhabung einer Activation Time (Aktivierungsdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Activation Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn kein solcher Eintrag vorhanden ist, oder die zertifizierte Zeit ⁴¹⁷ nach der Activation Time liegt.
Benötigt	Ein Activation Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten. Beachten Sie, dass dann eine Internet-Verbindung zum Abholen der zertifizierten Zeit erforderlich ist.
Ignorieren	Ein vorhandener Activation Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum vor der Activation Time liegt.

Wartungszeitraum (Maintenance Period)

Definiert die Handhabung eines Wartungszeitraumes (Maintenance Period), der in der Lizenz eingetragen ist. Eine Lizenz berechtigt dann zur Verwendung aller Softwareversionen, die innerhalb des definierten Wartungszeitraumes (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der Applikation hinterlegt und zur Laufzeit der geschützten Anwendung geprüft, ob das Erstelldatum (Release Date) innerhalb des Wartungszeitraumes (Maintenance Period) liegt (Kommandozeilen-Option siehe [hier](#)³⁰⁷).



Die Optionen sind nur auswählbar, wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) [aktiviert](#)¹¹⁵ worden ist.

Es bestehen zwei Überprüfungsoptionen:

Element	Beschreibung
Standard	Während der Laufzeit der geschützten Anwendung wird gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist. Dies entspricht der Standardeinstellung

Element	Beschreibung
	auch wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) nicht aktiviert ¹¹⁵ worden ist.
Benötigt	Während der Laufzeit der geschützten Anwendung ist das Prüfen des Wartungszeitraumes (Maintenance Period) gegen das Erstelldatum (Release Date) zwingend erforderlich. Die PIO Wartungszeitraum (Maintenance Period) muss vorhanden sein.

Zeitzertifikat

In jedem *CmContainer* ist eine laufende Uhr integriert, die läuft, wenn der *CmContainer* mit dem Rechner verbunden ist. Die Uhrzeit synchronisiert sich dabei beim Aktivieren des *CmContainers* nach vorne und nutzt ansonsten die letzte gespeicherte Zeit.

Wenn gewünscht, kann die zertifizierte Uhrzeit durch die Synchronisation mit dem *CodeMeter®* Zeitserver aktualisiert werden. Die Zeitserver sind von Wibu-Systems bereitgestellte Rechner, die über die Welt verteilt sind und eine zertifizierte Zeit zur Verfügung stellen. Bei einer Aktualisierung der zertifizierten Uhrzeit wird die interne *CmContainer*-Zeit synchronisiert (Kommandozeilen-Option siehe [hier](#) ³⁰⁰).

 Für Informationen zur Manipulationssicherheit von Aktivierungs- und Ablaufdatum siehe hier ⁴¹⁷ .
--

Element	Beschreibung
Zeitzertifikat setzen	Mit dieser Option wird versucht die zertifizierte Zeit im <i>CmContainer</i> zu aktualisieren. Die zertifizierte Zeit wird beim Zeitserver angefordert.  Diese Option erfordert eine Internet-Verbindung.
Zertifikat prüfen	Diese Option überprüft, ob die zertifizierte Zeit älter ist, als das hier festlegbare maximale Alter. Ist das maximale Alter des Zeitzertifikats überschritten, so lässt sich die Anwendung nicht starten.
Max. Alter des Zertifikats (in Stunden)	Bei ausgewählter "Prüfung" des Zeitzertifikats können Sie hier das maximale Alter des Zertifikats in Stunden angeben. Das Alter des Zertifikates berechnet sich aus der Differenz der laufenden System-Zeit und der zertifizierten Zeit.
Intervall ohne Überprüfung (Stunden)	Gibt an, innerhalb welchen Intervalls keine Überprüfung des Zeitzertifikats stattfindet. Ist dieses Intervall noch nicht erreicht, findet keine Überprüfung statt. Befindet sich das Zeitzertifikat zwischen diesem Intervall und dem max. Alter des Zertifikats, wird versucht, das Zeitzertifikat zu aktualisieren. Gelingt dies nicht, läuft die Anwendung jedoch bis zum Erreichen des max. Alters des Zeitzertifikats weiter. Erst danach ist zwingend ein aktualisiertes Zeitzertifikat notwendig.

System Datum

In diesem Bereich nehmen Sie Einstellungen vor, die dem zusätzlichen Schutz dienen, eine Lizenz über ein bewusstes Falschstellen der PC-Zeit zu manipulieren (Kommandozeilen-Option siehe [hier](#) ²⁹⁷).

Element	Beschreibung
Prüfen der Verschlüsselungszeit	Diese Option speichert die Verschlüsselungszeit (PC Time) in der geschützten Anwendung. Die Anwendung läuft auf dem Kunden-PC dann nur, wenn die <i>CmContainer</i> Systemzeit neuer ist als die Verschlüsselungszeit.

Element	Beschreibung
	 Erfordert mindestens <i>CodeMeter®</i> 4.10.
Abweichung CmContainer / PC Systemzeit	Wird diese Option aktiviert, ist die Festlegung eines Zeitkorridors möglich, innerhalb dessen sich die Abweichung zwischen <i>CmContainer</i> Systemzeit und der PC-Zeit bewegen darf. Wird dieser unter- bzw. überschritten, läuft die geschützte Anwendung auf dem Kunden-PC nicht.
Erlaubte Minuten nach hinten	Gibt in Minuten an, um wieviele Minuten die PC Zeit älter als die <i>CmContainer</i> Systemzeit sein darf.
Erlaubte Minuten nach vorne	Gibt in Minuten an, um wieviele Minuten die PC Zeit vor der <i>CmContainer</i> Systemzeit liegen darf.

7.4.4.5 Sicherheitsoptionen

Über diese Seite treffen Sie eine Auswahl aus verschiedenen Schutzmethoden für Ihre Anwendung. Sie können hier den Grad der Sicherheit selbst skalieren (Kommandozeilen-Option siehe [hier](#)¹⁹⁹).

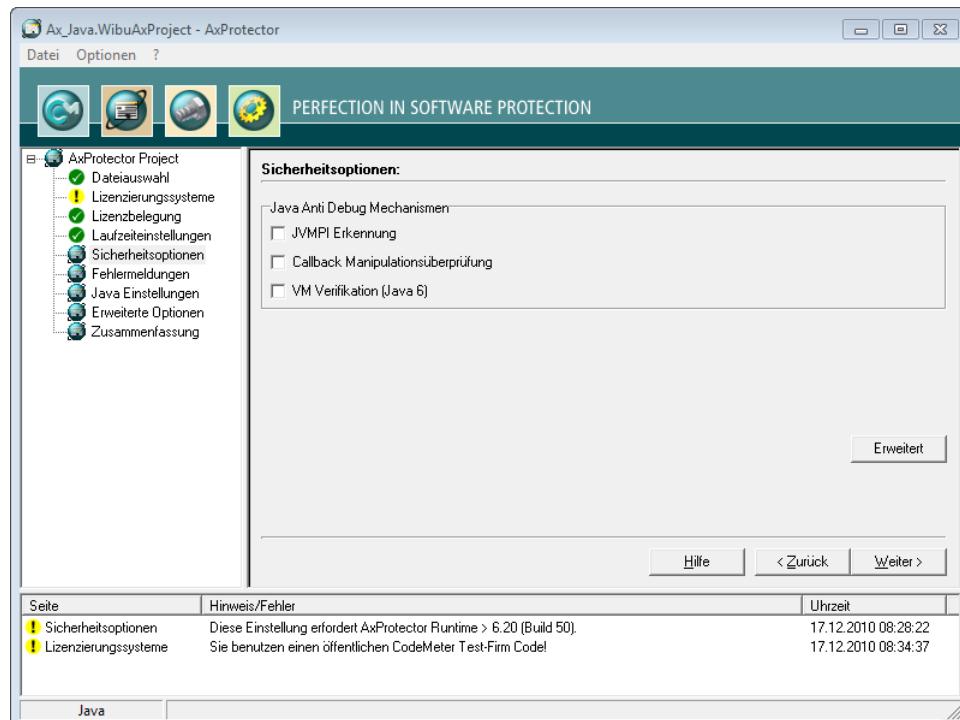


Abbildung 80: AxProtector - Java "Sicherheitsoptionen"

Element	Beschreibung
JVMPI Erkennung	Aktiviert die Erkennung des Java Virtual Machine Profiler Interface (JVMPI). Über JVMPI ist die Java Virtual Machine manipulierbar, so dass diese Nachrichten an nativen Code schickt. Insbesondere das Ereignis <code>JVMPI_EVENT_CLASS_LOAD_HOOK</code> kann dazu verwendet werden, den unverfälschten Byte Code der aktuell geladenen Klasse abzufangen. Das Aktivieren dieser Option verhindert dieses Vorgehen.
Callback Manipulations-Überprüfung	Aktiviert den Schutz gegen Manipulation von Callback-Funktionen. D.h. es findet eine Überprüfung von Funktionen statt, die anderen Funktionen als Parameter übergeben werden.
VM Verifikation (Java 6 + 7)	Aktiviert die Überprüfung nach der korrekten Java Virtual Machine-Laufzeitumgebung für Java 6 und 7.

7.4.4.6 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine User Message Class mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

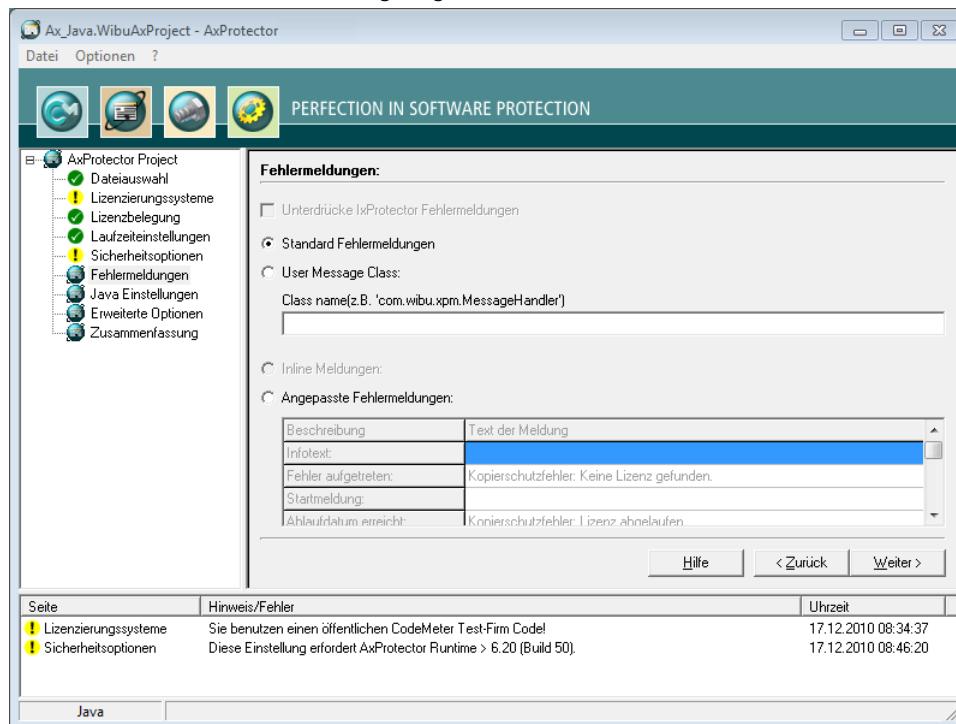


Abbildung 81: AxProtector - Java "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier □ ³¹⁰).
User Message Class	Aktiviert die Benutzung der User Message Class.
Class name	Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an.
Angepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlerertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.4.4.7 Java Einstellungen

In diesem Eingabefenster legen Sie einige Parameter zur Konfiguration der Java Umgebung fest.

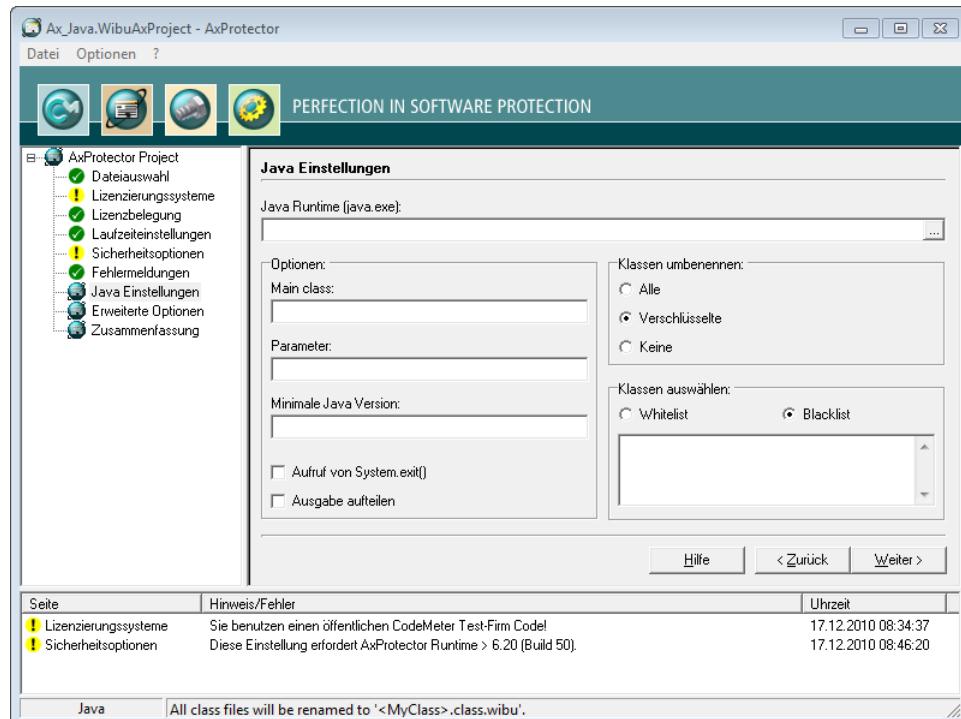


Abbildung 82: AxProtector - Java "Java Einstellungen" Java Runtime (java.exe)

Element	Beschreibung
Java Runtime (ja-va.exe)	Hier wählen Sie über die Schaltfläche "..." die <code>java.exe</code> -Datei der installierten Laufzeitumgebung aus.
Main class	Geben Sie hier den Name der Java main class ein (Kommandozeilen-Option siehe hier ³¹⁴).
Parameter	Definieren Sie hier die Parameter zum Aufrufen der Java main class (Kommandozeilen-Option siehe hier ³¹³).
Minimale Java Version	Geben Sie hier die minimal erforderliche Java Version an (Kommandozeilen-Option siehe hier ³¹⁴). <div style="border: 1px solid #ccc; padding: 5px; margin-left: 20px;"> i Geben Sie hier die minimal benötigte Java Version an. Schlägt die Überprüfung fehl, so wird eine entsprechende Fehlermeldung ausgegeben. So stellen Sie schon beim Start der geschützten Anwendung sicher, dass die von Ihrer Anwendung benötigte Funktionalität gegeben ist. </div>
Aufruf von Sys-tem.exit()	Das Aktivieren dieses Kontrollkästchens beendet die Anwendung durch Aufruf von <code>System.exit()</code> nach dem Rücksprung zur Java main class (Kommandozeilen-Option siehe hier ³¹⁵).

Element	Beschreibung								
	<p> Dies stellt sicher, dass im Fehlerfall die geschützte Anwendung korrekt und vollständig beendet wird, auch wenn der Fehler außerhalb der Java main class auftrat.</p>								
Ausgabe aufteilen	<p>Das Aktivieren dieses Kontrollkästchens bewirkt, dass Laufzeit-Klassen in die separate <code>wibuXpm4JRuntime.jar</code> gespeichert werden (Kommandozeilen-Option siehe hier³¹⁵).</p> <p> Das Auslagern des Wibu Classloader in eine separate Datei beschleunigt das Leistungsverhalten der geschützten Anwendung. Dadurch wird selbst bei mehreren verschlüsselten Klassen der Wibu Classloader nur einmal geladen.</p>								
Klassen umbenennen	<p>In diesem Bereich bestimmen Sie über Auswahlfelder mit gegenseitigem Ausschluss welche Klassen umbenannt und in den Wibu Classloader geladen werden (Kommandozeilen-Option siehe hier³¹⁴).</p> <p> Bei allen Einstellungen die Klassen betreffen, werden die betroffenen Klassen umbenannt zu <code><MyClass>.class.wibu</code>.</p> <table border="1" style="margin-top: 10px;"> <thead> <tr> <th style="background-color: #0070C0; color: white;">Element</th><th style="background-color: #0070C0; color: white;">Beschreibung</th></tr> </thead> <tbody> <tr> <td>Alle</td><td>Bei Auswahl werden alle vorhandenen Klassen umbenannt</td></tr> <tr> <td>verschlüsselte</td><td>Bei Auswahl werden nur die verschlüsselten Klassen umbenannt.</td></tr> <tr> <td>keine</td><td>Bei Auswahl werden keine Klassen umbenannt.</td></tr> </tbody> </table> <p> Werden nur die verschlüsselten Klassen umbenannt, werden nur diese vom Wibu Classloader geladen, wodurch das Leistungsverhalten der Anwendung verbessert wird. Werden alle Klassen umbenannt, erhöht dies zwar geringfügig die Sicherheit, jedoch verschlechtert sich gegebenenfalls das Leistungsverhalten der geschützten Anwendung.</p>	Element	Beschreibung	Alle	Bei Auswahl werden alle vorhandenen Klassen umbenannt	verschlüsselte	Bei Auswahl werden nur die verschlüsselten Klassen umbenannt.	keine	Bei Auswahl werden keine Klassen umbenannt.
Element	Beschreibung								
Alle	Bei Auswahl werden alle vorhandenen Klassen umbenannt								
verschlüsselte	Bei Auswahl werden nur die verschlüsselten Klassen umbenannt.								
keine	Bei Auswahl werden keine Klassen umbenannt.								
Klassen auswählen	<p>In diesem Bereich weisen Sie Klassen einer Positiv- bzw. Negativliste zu (Kommandozeilen-Option siehe hier³¹⁵).</p> <table border="1" style="margin-top: 10px;"> <thead> <tr> <th style="background-color: #0070C0; color: white;">Element</th><th style="background-color: #0070C0; color: white;">Beschreibung</th></tr> </thead> <tbody> <tr> <td>Whitelist</td><td>Alle Klassen, die in der Whitelist referenziert sind, werden verschlüsselt. Diese Whitelist wird in jedem Jar-Archiv als unverschlüsselte Textdatei <code>com/wibu/xpm/encrypted</code> abgelegt.</td></tr> <tr> <td>Blacklist</td><td>Alle Klassen auf dieser Ausschlussliste werden nicht verschlüsselt. AxProtector-Syntax: <code>-JL[W B]:<whitelist blacklist></code></td></tr> </tbody> </table> <p> Mit diesen Listen haben Sie direkten Einfluss auf die zu verschlüsselnden Klassen. So ist es eventuell nicht sinnvoll Klassen von Drittanbietern, die im Projekt Verwendung finden, zu schützen und dadurch das Leistungsverhalten der Anwendung zu belasten. Zur Ausgabe von Fehlermeldungen zur Laufzeit der verschlüsselten Java Anwendung kann die Fehlerklasse <code>com.wibu.xpm.MessageHandler</code> verwendet werden.</p>	Element	Beschreibung	Whitelist	Alle Klassen, die in der Whitelist referenziert sind, werden verschlüsselt. Diese Whitelist wird in jedem Jar-Archiv als unverschlüsselte Textdatei <code>com/wibu/xpm/encrypted</code> abgelegt.	Blacklist	Alle Klassen auf dieser Ausschlussliste werden nicht verschlüsselt. AxProtector-Syntax: <code>-JL[W B]:<whitelist blacklist></code>		
Element	Beschreibung								
Whitelist	Alle Klassen, die in der Whitelist referenziert sind, werden verschlüsselt. Diese Whitelist wird in jedem Jar-Archiv als unverschlüsselte Textdatei <code>com/wibu/xpm/encrypted</code> abgelegt.								
Blacklist	Alle Klassen auf dieser Ausschlussliste werden nicht verschlüsselt. AxProtector-Syntax: <code>-JL[W B]:<whitelist blacklist></code>								

7.4.4.8 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

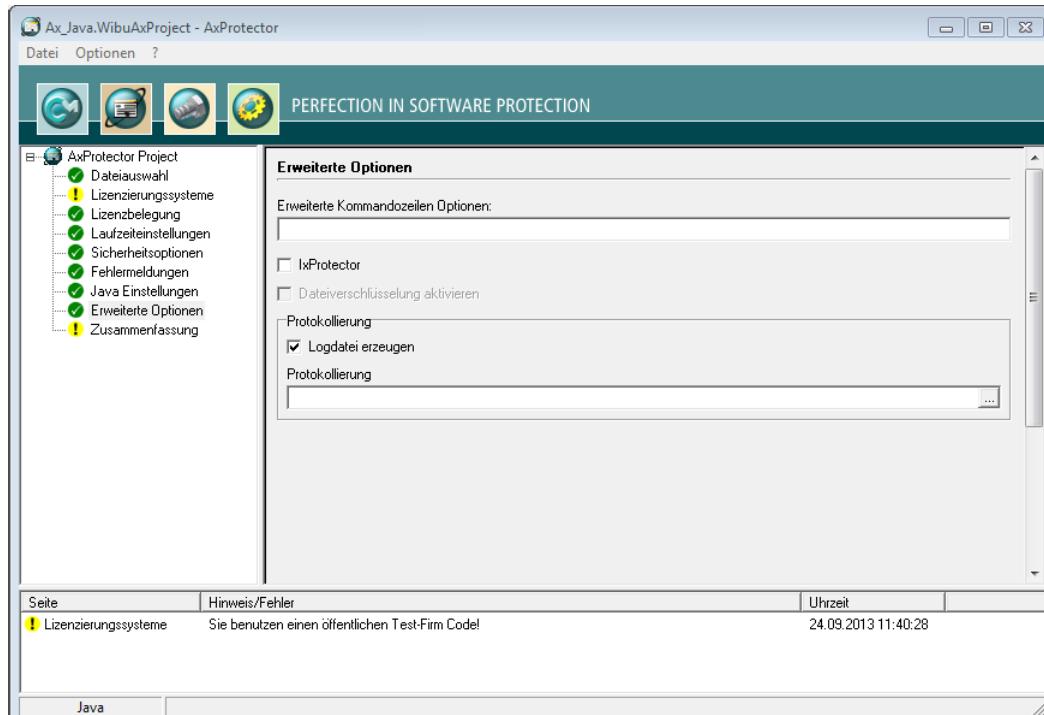


Abbildung 83: AxProtector - Java "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
IxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das Softwareschutz-API (WUPI) verwenden (Kommandozeilen-Optionen siehe hier).
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.

Element	Beschreibung
	Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.

7.4.4.9 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

 Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`

319

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

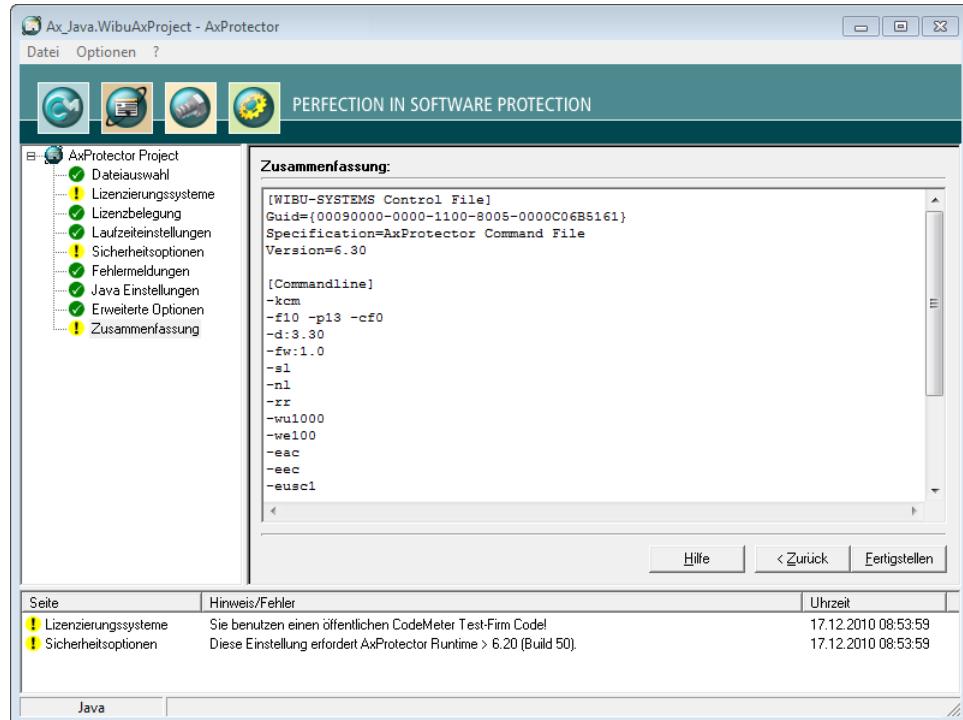


Abbildung 84: AxProtector - Java "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

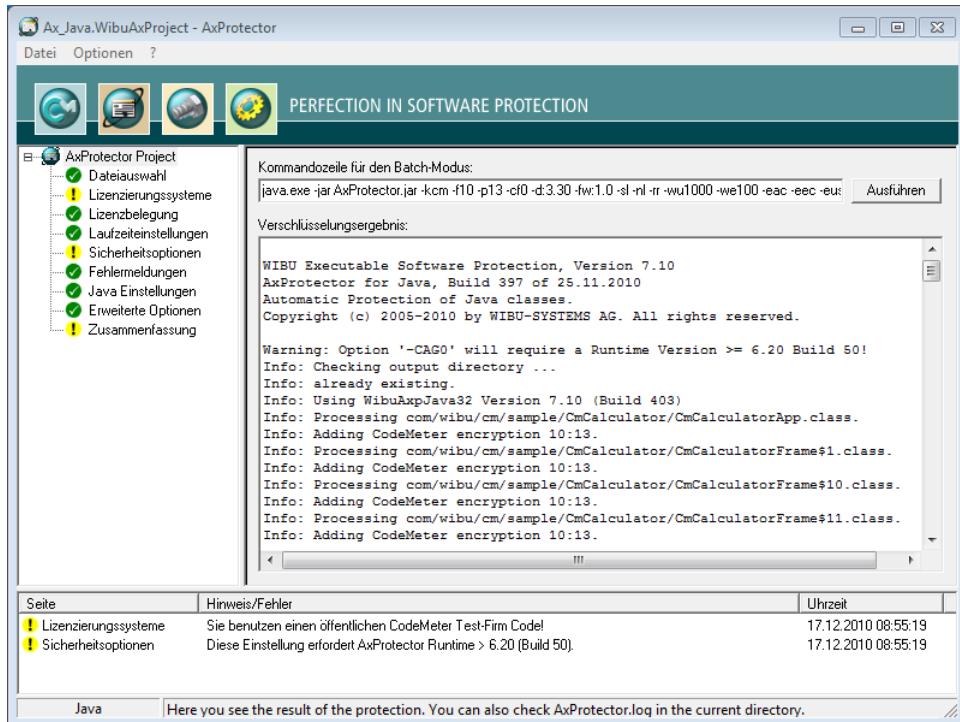


Abbildung 85: AxProtector - Java "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	<p>Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.4.5 Linux Anwendung oder Shared Object

Dieser Projekttyp gilt für die Verschlüsselung ausführbarer Anwendungen im Standard-Binärformat (ELF, Executable and Linking Format) und Programmblibliotheken (shared objects *.so).

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Linux mit AxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Linux Anwendung oder Shared Object	 AxProtector Linux ¹⁹¹		Windows Kommandozeile ²⁹²  In einer separaten Kommandozeile für Linux, die auf Linux-Betriebssystemen läuft, können Sie ebenfalls Verschlüsselungsparameter ²¹⁷ eingeben.

7.4.5.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

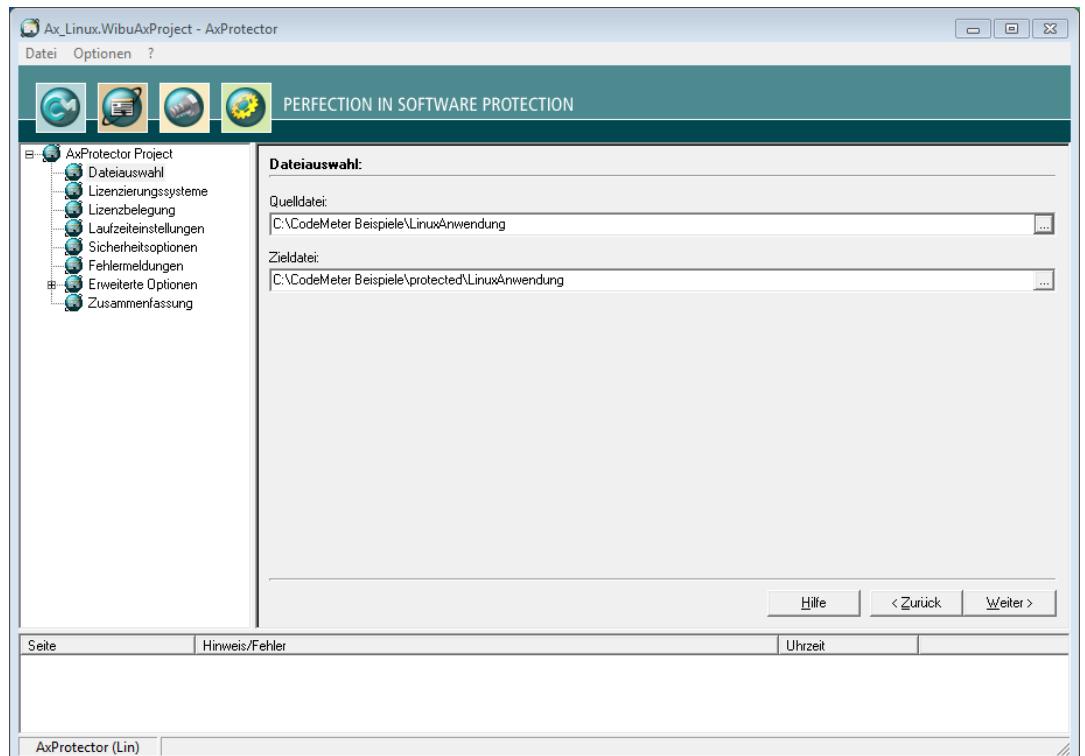


Abbildung 86: AxProtector - Linux "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein.  Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..\protected\..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.4.5.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Datei nehmen Sie hier Einstellungen zum verwendeten Lizenzierungssystem *CmDongle* und / oder *CmActLicense* vor.

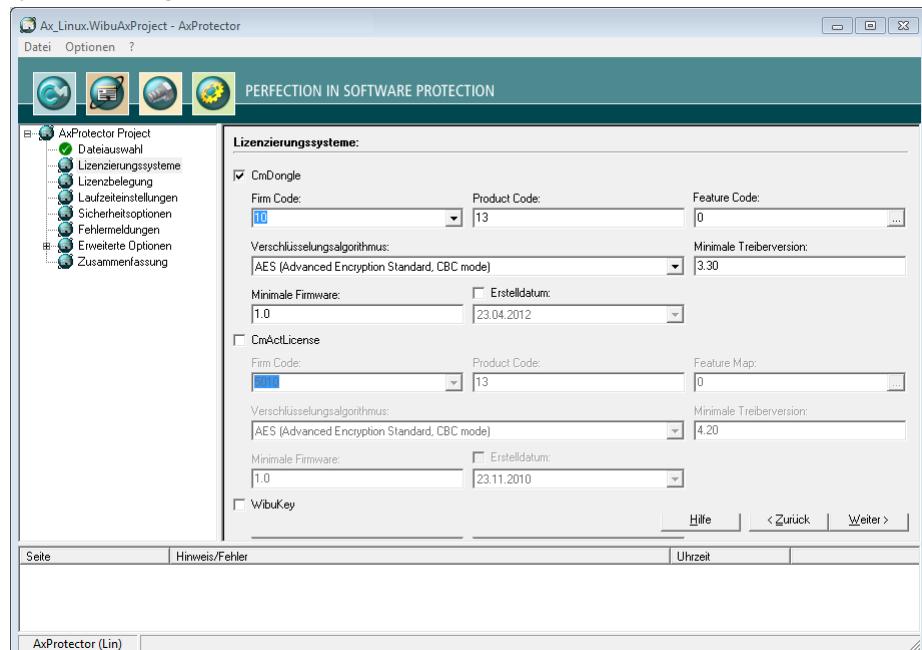


Abbildung 87: AxProtector - Linux "Lizenzierungssysteme"

Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenzierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenzierungssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die Wibu-Systems Internetseiten.

Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich (siehe Kommandozeilen-Option [hier](#)²⁹³):

Element	Beschreibung
Firm Code	Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird. Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle</i> Evaluation-Firm Code des

Element	Beschreibung
	<p>CodeMeter® Software Development Kits (SDK) und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010.</p> <p>Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bestimmt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eingeben.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
	<p>Abbildung 88: AxProtector - Mac OS Feature Map Eingabe</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. CodeMeter® unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Minimale Treiberversion	<p>Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i> an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt AxProtector automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizenzen.</p> <p> Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen beim Starten Ihrer geschützten Software.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

Element	Beschreibung
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt <i>CodeMeter</i>® die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden.</p> <p>Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können. Beachten Sie auch, dass Sie hier das Kontrollkästchen aktivieren müssen, um Überprüfungsoptionen des Wartungszeitraumes (Maintenance Period) im Dialog zu den erweiterten Laufzeiteinstellungen²⁰⁰ vornehmen zu können. </div>
Minimale Firmware	<p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem *WibuKey* informiert separat das *WibuKey* Entwicklerhandbuch.

7.4.5.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Anwendung vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

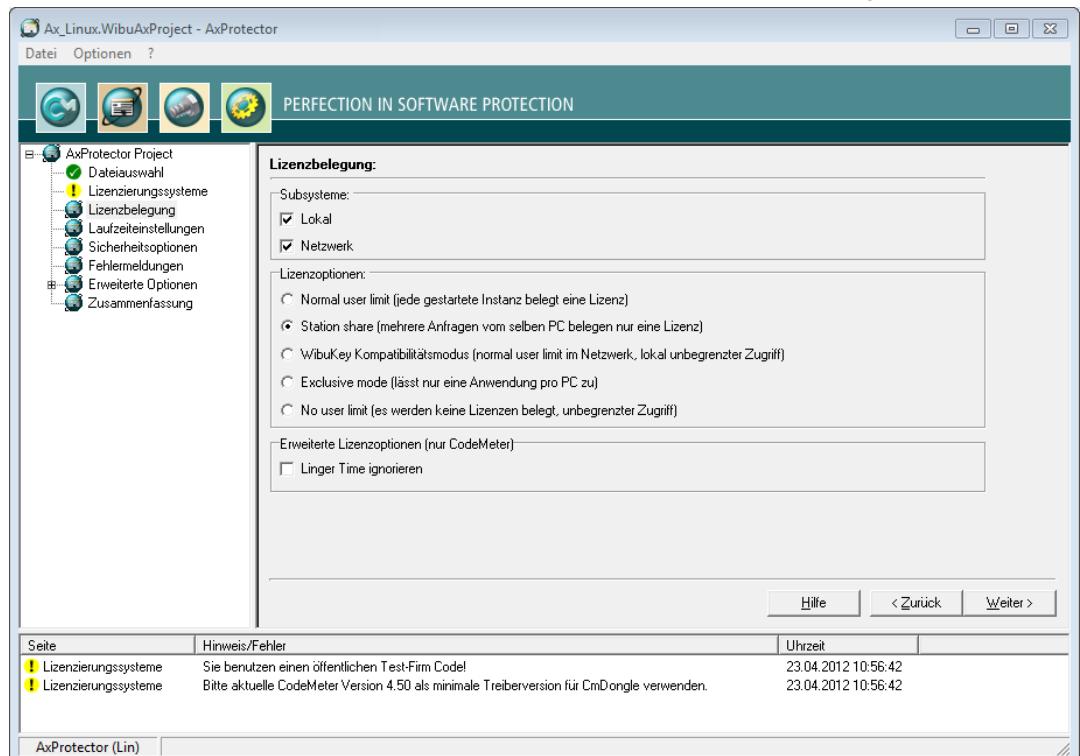


Abbildung 89: AxProtector - Linux "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#)²⁹⁴).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der <i>CodeMeter Lizenzserver</i> mit einem aktvierten Netzwerkgzugriff läuft.

 Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizenzoptionen

Im Bereich Lizenzoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier](#)²⁰⁶).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz.  Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.
WibuKey Kompatibilitäts-Modus	Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).  Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu WibuKey. Wibu-Systems <u>empfiehlt</u> die Einstellungen 'Normal user limit' und 'Station Share'.
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizenzen belegt werden. Belegte Lizenzen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).

7.4.5.4 Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie das Verhalten der Anwendung zur Laufzeit fest, z.B. Abfrage der Lizenz in *CmContainern*, Ausgabe von Warnmeldung, etc..

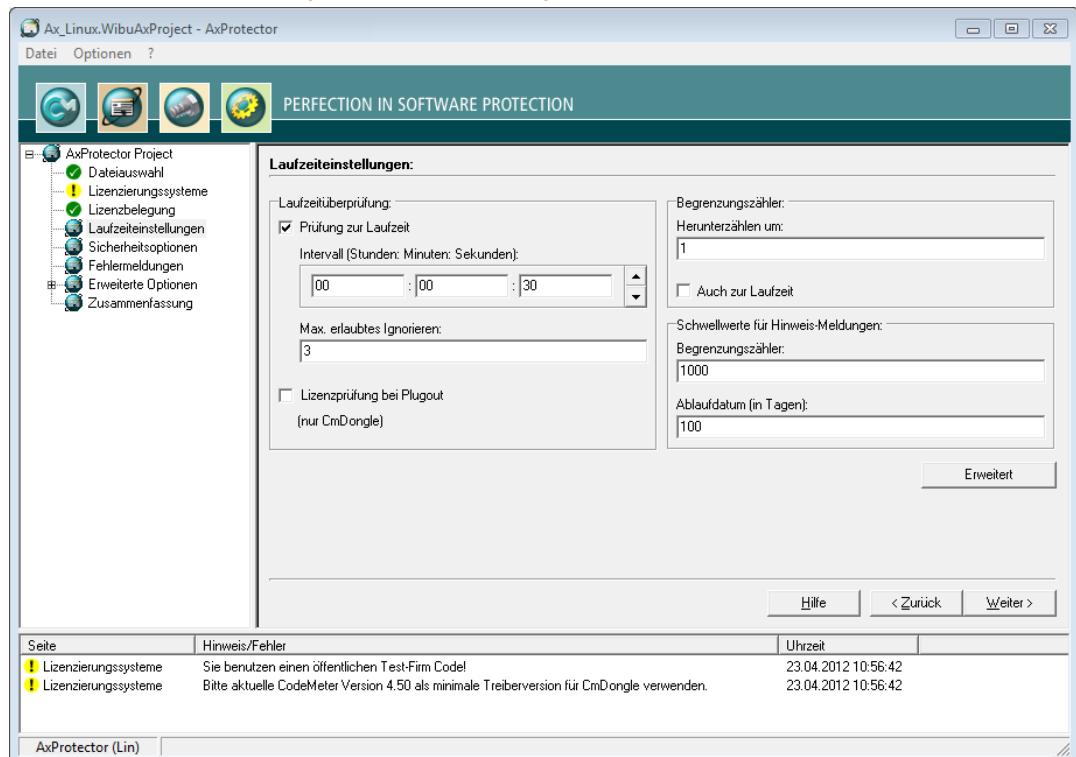


Abbildung 90: AxProtector - Linux "Laufzeiteinstellungen"

Laufzeitüberprüfung

In diesem Bereich können Sie definieren, ob und wie oft die geschützte Anwendung die Lizenz während der Laufzeit überprüft.

Elemente	Beschreibung
Prüfung zur Laufzeit	Aktiviert oder deaktiviert die Überprüfung während der Laufzeit der geschützten Anwendung. Kommandozeilen-Option siehe hier .
Intervall	Legt das Intervall zwischen zwei Überprüfungen fest. Angabe im Format Stunden: Minuten: Sekunden.
Max. erlaubtes Ignorieren	Gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann. i Schlägt die Verbindung zum <i>CmContainer</i> fehl, d.h. kann nicht mehr auf die Lizenz zugriffen werden, geben Sie dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit, auch ohne die Lizenz noch weiterzuarbeiten.

Elemente	Beschreibung
Lizenzprüfung bei Plug-Out (nur CmDongle)	Beendet die geschützte Anwendung, wenn der <i>CmDongle</i> während der Ausführung abgezogen wird und eine sofortige Fehlermeldung wird ausgegeben. Kommandozeilen-Option siehe hier ²⁹⁷ .

Begrenzungszähler

Begrenzungszähler (Unit Counter) können u.a. dazu dienen, die Gültigkeit von Lizzenzen in einem *CmContainer* festzustellen. In diesem Bereich können Sie dieses Verhalten definieren (Kommandozeilen-Option siehe [hier](#)²⁹⁸).

Element	Beschreibung
Herunterzählen um	Gibt den Wert an, um den der Begrenzungszähler (Unit Counter) heruntergezählt wird. Diese Option bewirkt das Herunterzählen des Zählers beim Start der geschützten Anwendung. Ist die "Auch zur Laufzeit" Option aktiviert und sind die Einträge wie in der obigen Abbildung dargestellt gesetzt, wird alle 30 Sekunden (siehe das festgelegt Intervall) ein gesetzter Begrenzungszähler (Unit Counter) um den Wert 1 heruntergezählt.
Auch zur Laufzeit	Zählt den Begrenzungszähler (Unit Counter) auch während der Laufzeit der geschützten Anwendung herunter.  Diese Option greift nur, wenn die "Prüfung zu Laufzeit" Option im Bereich "Laufzeitüberprüfung" aktiviert ist.

Schwellenwerte für Hinweismeldungen

In diesem Bereich können Sie definieren, wann eine Hinweismeldung zur Gültigkeit der Lizenz ausgegeben wird.

	Zur individuellen Gestaltung des Textes der Hinweismeldungen siehe hier ²⁰⁶ .
--	--

Element	Beschreibung
Begrenzungszähler	Wird der angegebene Schwellenwert unterschritten, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .
Ablaufdatum (in Tagen)	Wird das angegebene Ablaufdatum in Tagen innerhalb der vorgegebenen Schwelle erreicht, wird ein Warnhinweis ausgegeben. Kommandozeilen-Option siehe hier ³⁰⁹ .

7.4.5.4.1 Erweiterte Laufzeiteinstellungen

Über dieses Eingabefenster legen Sie zusätzliche Einstellungen zur Laufzeit der verschlüsselten Anwendung fest.

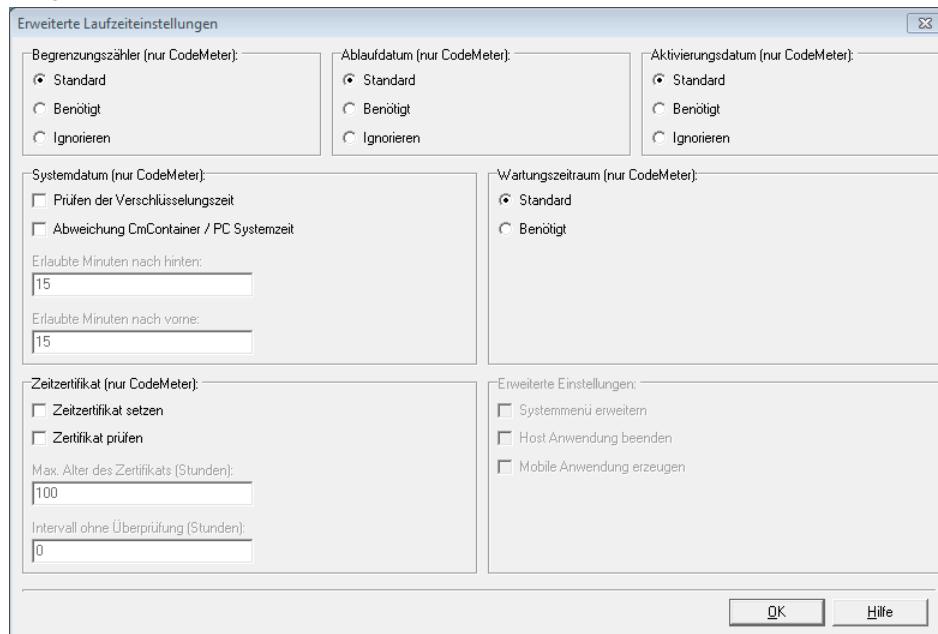


Abbildung 91: AxProtector - Linux "Erweiterte Laufzeiteinstellungen"

Für die Abfrage der in die Lizenz eingetragenen Optionen Begrenzungszähler (Unit Counter), Ablaufdatum (Expiration Time) und Aktivierungsdatum (Activation Time) gilt die folgende Handhabung.

Status	Standard	Benötigt	Ignorieren
= 0	X	X	✓
< > 0	✓	✓	✓
nicht angegeben	✓	✓	✓

Begrenzungszähler (Unit Counter)

Definiert die Handhabung eines Unit Counter (Begrenzungszählers), der in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)^{▷ 308}).

Element	Beschreibung
Standard	Zählt einen vorhandenen Unit Counter-Eintrag in der Lizenz beim Start und/oder zur Laufzeit um den auf der vorherigen Seite definierten Wert herunter. Wenn der Unit Counter Null erreicht startet die verschlüsselte Anwendung nicht.
Benötigt	Ein Unit Counter-Eintrag < > 0 in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag star-

Element	Beschreibung
	tet die verschlüsselte Anwendung nicht.
Ignorieren	Ein vorhandener Unit Counter-Eintrag in der Lizenz wird ignoriert. Die Anwendung setzt den Unit Counter nicht herunter. Die Anwendung startet auch bei einem Unit Counter-Eintrag = 0.

Ablaufdatum (Expiration Time)

Definiert die Handhabung einer Expiration Time (Ablaufdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Expiration Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn keine Expiration Time vorhanden ist, oder das aktuelle Datum vor der Expiration Time liegt.
Benötigt	Ein Expiration Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten.
Ignorieren	Ein vorhandener Expiration Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum nach der Expiration Time liegt.

Aktivierungsdatum (Activation Time)

Definiert die Handhabung einer Activation Time (Aktivierungsdatum), die in der Lizenz eingetragen ist (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

Element	Beschreibung
Standard	Überprüft, ob ein Activation Time-Eintrag in der Lizenz vorhanden ist. Die Anwendung lässt sich aber auch starten, wenn kein solcher Eintrag vorhanden ist, oder die zertifizierte Zeit ⁴¹⁷ nach der Activation Time liegt.
Benötigt	Ein Activation Time-Eintrag in der Lizenz ist zwingend notwendig, ohne einen solchen Eintrag lässt sich die Anwendung nicht starten. Beachten Sie, dass dann eine Internet-Verbindung zum Abholen der zertifizierten Zeit erforderlich ist.
Ignorieren	Ein vorhandener Activation Time-Eintrag in der Lizenz wird ignoriert, auch wenn das aktuelle Datum vor der Activation Time liegt.

Wartungszeitraum (Maintenance Period)

Definiert die Handhabung eines Wartungszeitraumes (Maintenance Period), der in der Lizenz eingetragen ist. Eine Lizenz berechtigt dann zur Verwendung aller Softwareversionen, die innerhalb des definierten Wartungszeitraumes (Maintenance Period) erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der Applikation hinterlegt und zur Laufzeit der geschützten Anwendung geprüft, ob das Erstelldatum (Release Date) innerhalb des Wartungszeitraumes (Maintenance Period) liegt (Kommandozeilen-Option siehe [hier](#)³⁰⁷).

 Die Optionen sind nur auswählbar, wenn auf der Seite "Lizenzerierungssysteme" das Erstelldatum (Release Date) [aktiviert](#)¹⁹⁴ worden ist.

Es bestehen zwei Überprüfungsoptionen:

Element	Beschreibung
Standard	Während der Laufzeit der geschützten Anwendung wird gegen das Erstelldatum (Release Date) nur geprüft wird, falls ein Wartungszeitraum (Maintenance Period) vorhanden ist. Dies entspricht der Standardeinstellung

Element	Beschreibung
	auch wenn auf der Seite "Lizenzierungssysteme" das Erstelldatum (Release Date) nicht aktiviert ¹⁹⁴ worden ist.
Benötigt	Während der Laufzeit der geschützten Anwendung ist das Prüfen des Wartungszeitraumes (Maintenance Period) gegen das Erstelldatum (Release Date) zwingend erforderlich. Die PIO Wartungszeitraum (Maintenance Period) muss vorhanden sein.

Zeitzertifikat

In jedem *CmContainer* ist eine laufende Uhr integriert, die läuft, wenn der *CmContainer* mit dem Rechner verbunden ist. Die Uhrzeit synchronisiert sich dabei beim Aktivieren des *CmContainers* nach vorne und nutzt ansonsten die letzte gespeicherte Zeit.

Wenn gewünscht, kann die zertifizierte Uhrzeit durch die Synchronisation mit dem *CodeMeter®* Zeitserver aktualisiert werden. Die Zeitserver sind von Wibu-Systems bereitgestellte Rechner, die über die Welt verteilt sind und eine zertifizierte Zeit zur Verfügung stellen. Bei einer Aktualisierung der zertifizierten Uhrzeit wird die interne *CmContainer*-Zeit synchronisiert (Kommandozeilen-Option siehe [hier](#)¹⁹⁹).

	Für Informationen zur Manipulationssicherheit von Aktivierungs- und Ablaufdatum siehe hier ⁴¹⁷
---	---

Element	Beschreibung
Zeitzertifikat setzen	Mit dieser Option wird versucht die zertifizierte Zeit im <i>CmContainer</i> zu aktualisieren. Die zertifizierte Zeit wird beim Zeitserver angefordert. <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;"> Diese Option erfordert eine Internet-Verbindung.</div>
Zertifikat prüfen	Diese Option überprüft, ob die zertifizierte Zeit älter ist, als das hier festlegbare maximale Alter. Ist das maximale Alter des Zeitzertifikats überschritten, so lässt sich die Anwendung nicht starten.
Max. Alter des Zertifikats (in Stunden)	Bei ausgewählter "Prüfung" des Zeitzertifikats können Sie hier das maximale Alter des Zertifikats in Stunden angeben. Das Alter des Zertifikates berechnet sich aus der Differenz der laufenden System-Zeit und der zertifizierten Zeit.
Intervall ohne Überprüfung (Stunden)	Gibt an, innerhalb welchen Intervalls keine Überprüfung des Zeitzertifikats stattfindet. Ist dieses Intervall noch nicht erreicht, findet keine Überprüfung statt. Befindet sich das Zeitzertifikat zwischen diesem Intervall und dem max. Alter des Zertifikats, wird versucht, das Zeitzertifikat zu aktualisieren. Gelingt dies nicht, läuft die Anwendung jedoch bis zum Erreichen des max. Alters des Zeitzertifikats weiter. Erst danach ist zwingend ein aktualisiertes Zeitzertifikat notwendig.

System Datum

In diesem Bereich nehmen Sie Einstellungen vor, die dem zusätzlichen Schutz dienen, eine Lizenz über ein bewusstes Falschstellen der PC-Zeit zu manipulieren (Kommandozeilen-Option siehe [hier](#)²⁹⁷).

Element	Beschreibung
Prüfen der Verschlüsselungszeit	Diese Option speichert die Verschlüsselungszeit (PC Time) in der geschützten Anwendung. Die Anwendung läuft auf dem Kunden-PC dann nur, wenn die <i>CmContainer</i> Systemzeit neuer ist als die Verschlüsselungszeit.

Element	Beschreibung
	 Erfordert mindestens <i>CodeMeter® 4.10</i> .
Abweichung CmContainer / PC Systemzeit	Wird diese Option aktiviert, ist die Festlegung eines Zeitkorridors möglich, innerhalb dessen sich die Abweichung zwischen <i>CmContainer</i> Systemzeit und der PC-Zeit bewegen darf. Wird dieser unter- bzw. überschritten läuft die geschützte Anwendung auf dem Kunden-PC nicht.
Erlaubte Minuten nach hinten	Gibt in Minuten an, um wieviele Minuten die PC Zeit älter als die <i>CmContainer</i> Systemzeit sein darf.
Erlaubte Minuten nach vorne	Gibt in Minuten an, um wieviele Minuten die PC Zeit vor der <i>CmContainer</i> Systemzeit liegen darf.

7.4.5.5 Sicherheitsoptionen

Über diese Seite treffen Sie eine Auswahl aus verschiedenen Schutzmechanismen und -Methoden für Ihre Anwendung. Sie können hier den Grad der Sicherheit selbst skalieren. Dabei legen Sie z.B. selbst fest, wie intensiv nach Debuggern gesucht werden soll, bis hin zum Sperren des *CmContainers*.



Sollten sich die gesetzten Optionen inkompatibel zu Ihrer geschützten Anwendung verhalten, so können die einzelnen Sicherheitsoptionen auch einzeln deaktiviert werden.

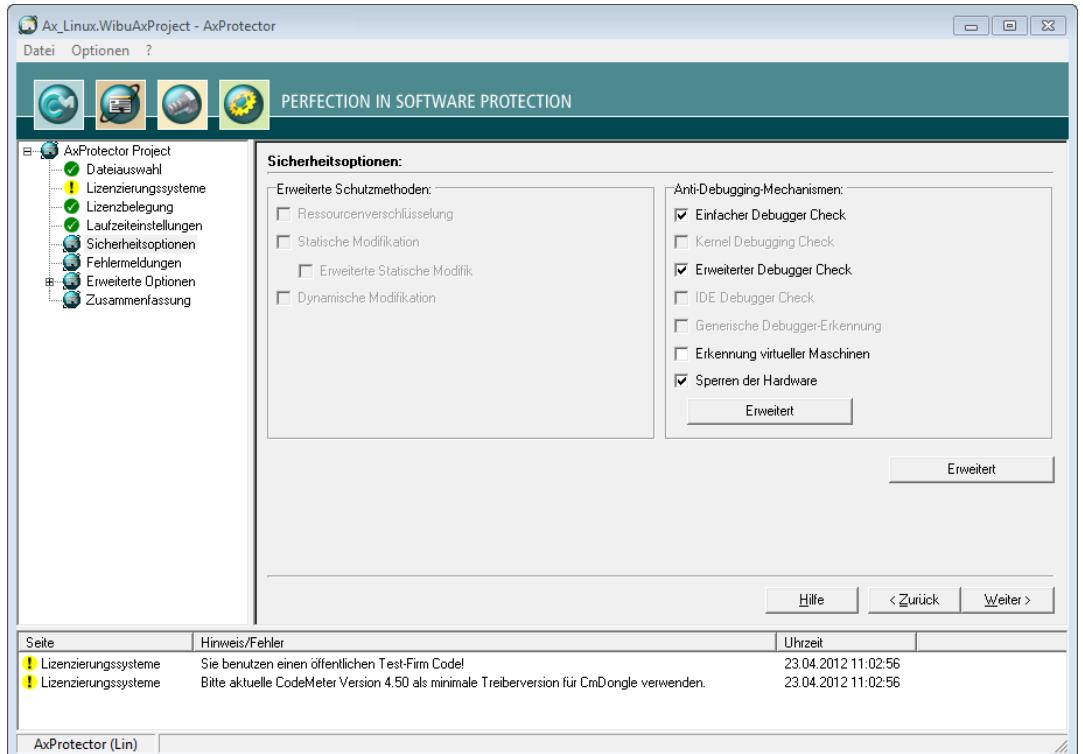
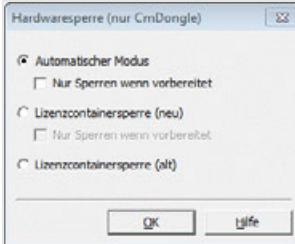
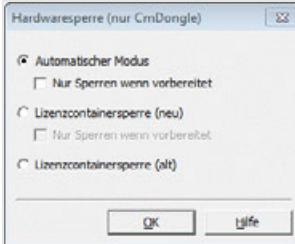
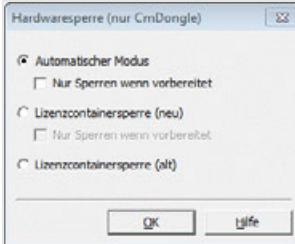


Abbildung 92: AxProtector - Linux "Sicherheitsoptionen"

Anti-Debugging Mechanismen

Debugger-Programme dienen der Fehlersuche und Fehlerbeseitigung, können aber auch von Hackern zur Analyse der Software verwendet werden. In diesem Bereich legen Sie die Optionen fest, wie auf Debugger-Programme reagiert werden soll (Kommandozeilen-Option siehe [hier](#)^[297]).

Element	Beschreibung
Einfacher Debugger Check	Überprüft ob ein Debugger an Ihre Anwendung angehängt (attached) ist. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.
Erweiterter Debugger Check	Überprüft in einer erweiterten Suche auf Debugger-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet bzw. beendet.
Erkennung virtueller Maschinen	Erkennt, ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies.
Sperren des Lizenz-Zugriffs	Mit dieser Option kann das genutzte Firm Item im CmContainer gesperrt werden sobald ein Debugger-Programm entdeckt wird. Wird die Option aktiviert, werden die Einstellungen übernommen, die Sie in einem Dialog setzen, der sich über die "Konfiguration"-Schaltfläche öffnet.

Element	Beschreibung						
	<p> Diese Schaltfläche ist nur aktiviert für CodeMeter.</p>						
Konfiguration	<p>Wenn die Option "Sperren des Lizenz-Zugriffs" aktiviert wird, können Sie über die "Konfiguration"-Schaltfläche im folgenden Dialog weitere Einstellungen vornehmen:</p> <p>Der Dialog erlaubt in Abhängigkeit von der Firmware, die bei der Verschlüsselung verwendet wird, die Auswahl verschiedener Sperrszenarien.</p> <table border="1"> <thead> <tr> <th>Sperrszenario</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>sofortiges Sperren</td><td>erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.</td></tr> <tr> <td>vorbereitetes Sperren</td><td> <p>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines CmContainers. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht.</p> <p>Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt.</p> <p>Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</p>  </td></tr> </tbody> </table>	Sperrszenario	Beschreibung	sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.	vorbereitetes Sperren	<p>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines CmContainers. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht.</p> <p>Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt.</p> <p>Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</p> 
Sperrszenario	Beschreibung						
sofortiges Sperren	erfolgt ab einer Firmware-Version 1.14 sobald ein Debugger erkannt wird.						
vorbereitetes Sperren	<p>erfolgt über eine Abfrage des Firm Access Counter (FAC). Der Firm Access Counter liegt auf der Firm Item-Ebene eines CmContainers. Über diesen Zähler ist es möglich zu kontrollieren, ob ein Firm Item für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht.</p> <p>Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535 (0xFFFF). Er kann jedoch vom Software-Hersteller auf andere Werte programmiert werden. Bei Erkennen eines Debuggers wird der FAC um den Wert 1 heruntergezählt.</p> <p>Erreicht der FAC einen Wert von 0, wird das Firm Item gesperrt. Der Besitzer / Anwender des gesperrten Firm Items muss zwecks Aufhebung der Sperre dann mit dem Software-Hersteller in Kontakt treten. Das Firm Item kann vom Softwarehersteller per Remote Programming wieder freigeschaltet werden.</p> 						
	<p>Abbildung 93: AxProtector - Mac OS "Sicherheitsoptionen - Hardware-sperre"</p> <p>Die folgenden Einstellungen sind verfügbar</p> <table border="1"> <thead> <tr> <th>Einstellung</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)</td><td> <p>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt.</p> <p>Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items.</p> <p>Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</p> </td></tr> <tr> <td>"Automatischer Modus" markiert und Kontrollkästchen</td><td> <p>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.</p> </td></tr> </tbody> </table>	Einstellung	Beschreibung	"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)	<p>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt.</p> <p>Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items.</p> <p>Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</p>	"Automatischer Modus" markiert und Kontrollkästchen	<p>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.</p>
Einstellung	Beschreibung						
"Automatischer Modus" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert (Standard)	<p>Wenn die Firmware kleiner als 1.14 ist, wird der FAC im Rahmen eines vorbereiteten Sperren um den Wert 1 heruntergezählt.</p> <p>Ist die Firmware 1.14 und höher erfolgt ein sofortiges Sperren des Firm Items.</p> <p>Dies entspricht aus Kompatibilitätsgründen der Standard-Einstellung.</p>						
"Automatischer Modus" markiert und Kontrollkästchen	<p>Wenn die Firmware kleiner als 1.14 ist, dann tritt ein Herunterzählen des FAC in Kraft.</p>						

Element	Beschreibung	
	Einstellung	Beschreibung
	"Nur Sperren wenn vorbereitet" aktiviert	Ist die Firmware 1.14 und höher, dann wird gleichzeitig geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.
	"Lizenzcontainersperre (neu)" markiert und Kontrollkästchen "Nur Sperren wenn vorbereitet" nicht aktiviert	Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Dies ist sicherheitstechnisch gesehen die empfohlene Einstellung. Voraussetzung ist jedoch, dass alle <i>CmContainer</i> im Feld mit einer Firmware Version 1.14 und höher ausgestattet sind.
	"Lizenzcontainersperre (neu)" markiert und das Kontrollkästchen "Nur Sperren wenn vorbereitet" aktiviert	Die Firmware ist 1.14 und höher und eine sofortige Sperrung des Firm Items erfolgt. Gleichzeitig wird geprüft, ob ein vorbereitetes Sperren programmiert ist. Ist die Sperrung vorbereitet, erfolgt die Sperrung des Firm Items.
	Option "Lizenzcontainersperre (alt)" markiert	Gilt für alle Firmware-Versionen. Ist ein vorbereitetes Sperren programmiert, wird der FAC um den Wert 1 heruntergezählt.

7.4.5.5.1 Erweiterte Sicherheitsoptionen

Ermöglicht die Auswahl zusätzlicher Sicherheitseinstellungen.



Abbildung 94: AxProtector - Linux "Erweiterte Sicherheitsoptionen"

Erweiterte Einstellungen

Dieser Bereich lässt die Auswahl weitere Optionen zu.

Element	Beschreibung
Virusprüfung hinzufügen	Der geschützten Anwendung wird eine Virenprüfung über eine Prüfsumme hinzugefügt (Kommandozeilen-Option siehe hier ³⁰¹).
API statisch linken	Das <i>CodeMeter Kern-API</i> wird statisch zur geschützten Anwendung hinzugelinkt. Diese Option erhöht die Sicherheit, sie vergrößert jedoch auch die ausführbare Datei (Kommandozeilen-Option siehe hier ³⁰²).
Zu verschlüsselnder Code	Hier kann die Menge des zu verschlüsselnden Codes (in %) angegeben werden (Kommandozeilen-Option siehe hier ³⁰⁰).

Element	Beschreibung
(in %)	

7.4.5.6 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob Sie entweder eine eigene angepasste Fehlerausgabe verwenden, oder ob Standard-Hinweisfenster angezeigt werden sollen.

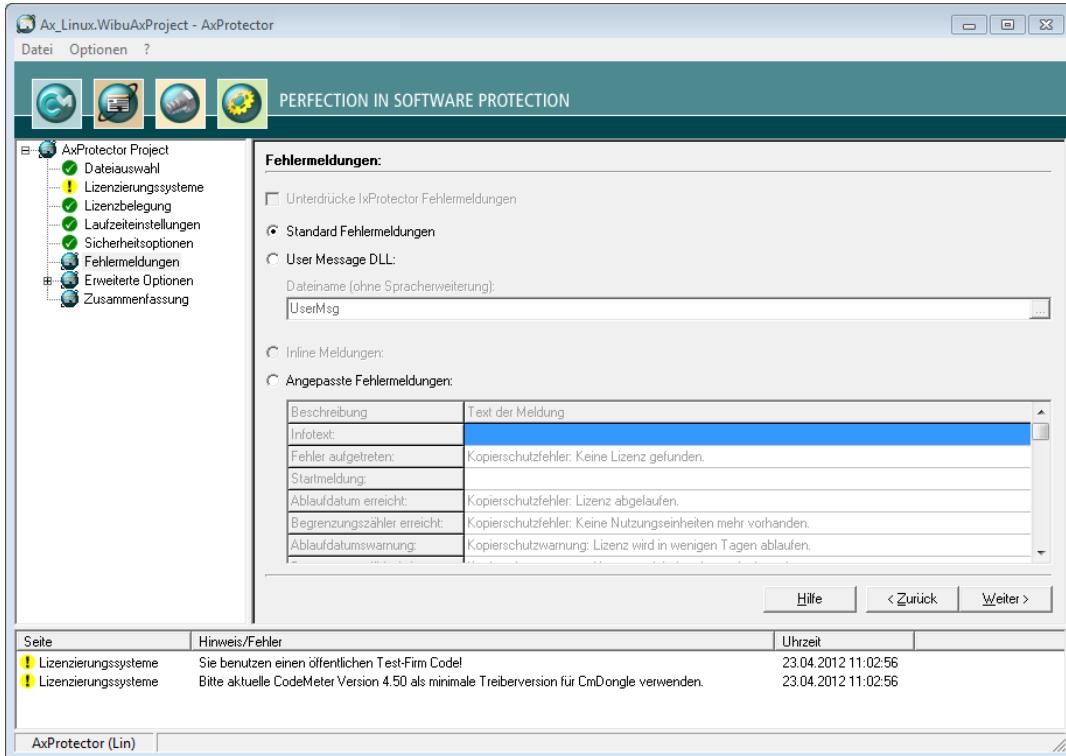
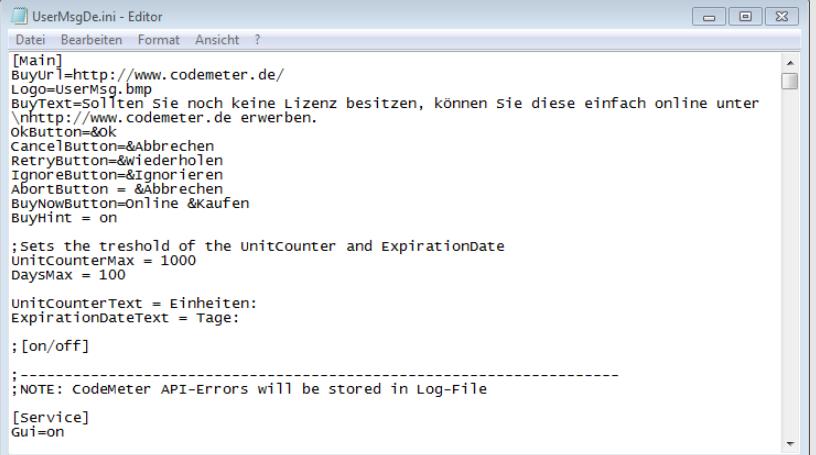


Abbildung 95: AxProtector - Linux "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier [310]).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlermeldungen können über * .ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen.

Element	Beschreibung
	<p>gen (Kommandozeilen-Option siehe hier³¹²).</p> <p> Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die Dll-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector geschützte Anwendung befindet.</p>  <p>Abbildung 96: AxProtector - UserMsg De.ini</p> <p>Dateiname (ohne Spracherweiterung)</p> <p>Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an. Die UserMsgDll wird aus dem Verzeichnis %Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage kopiert. Die jeweiligen Initialisierungsdateien sind ebenfalls in diesem Verzeichnis abgelegt.</p>
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.4.5.7 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

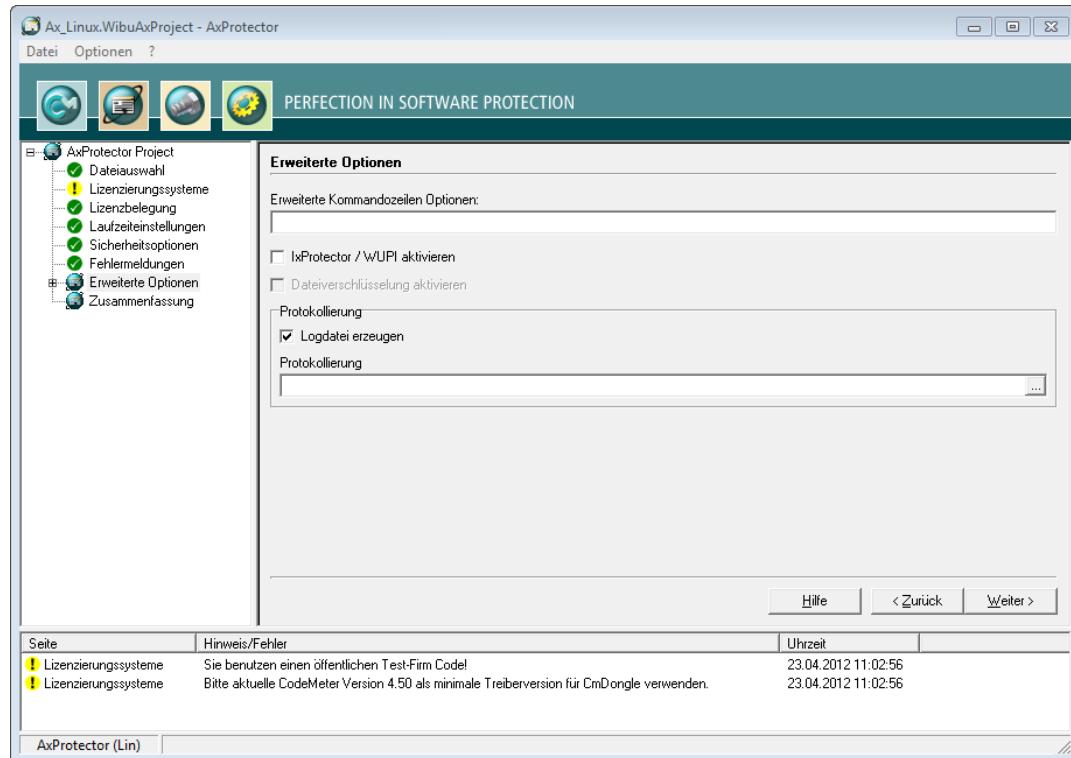


Abbildung 97: AxProtector - Linux "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
IxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das Softwareschutz-API (WUPI) verwenden (Kommandozeilen-Option siehe hier).
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.

Element	Beschreibung
	Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.

7.4.5.7.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit AxProtector über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

The screenshot shows the AxProtector software interface with the title bar "Ax_Linux.WibuAxProject - AxProtector". The menu bar includes "Datei" and "Optionen". The main window has a toolbar with icons for project management and protection levels. The left sidebar shows a tree view of the project structure under "AxProtector Project", including "Dateiauswahl", "Lizenzierungssysteme", "Lizenzbelegung", "Laufzeiteinstellungen", "Sicherheitsoptionen", "Fehlermeldungen", "Erweiterte Optionen", "Lizenzlisten" (which is selected and expanded), "IxProtector", and "Zusammenfassung". The right panel is titled "Lizenzlisten" and displays a table of existing license lists:

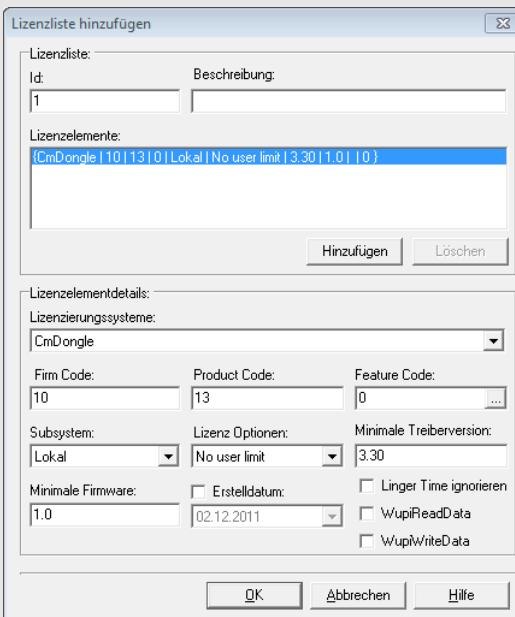
ID	Beschreibung	Elemente	Element Details
0	(Standardlizenz)	(1 Element)	(CmDongle 10 1310 Lokal Normal user limit 4.30 1.18 23.04.2012 0 none) ;

Buttons at the bottom of the right panel include "Hinzufügen", "Bearbeiten", and "Löschen". Below the main window, a status bar shows "AxProtector (Lin)" and a message about error/warning logs. A footer navigation bar includes "Hilfe", "Zurück", and "Weiter".

Abbildung 98: AxProtector - Linux "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "Hinzufügen" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	<p>Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.</p> <p> Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.</p>
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag.</p> <p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
	<p>Abbildung 99: AxProtector -Linux "Lizenzlisten hinzufügen"</p>
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (<i>CmDongle</i> , <i>CmActLicense</i> oder <i>WibuKey</i>).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	<p>Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt.</p> <p>Über die "... " Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal) .</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenzen:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt <i>CodeMeter®</i> die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzeigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

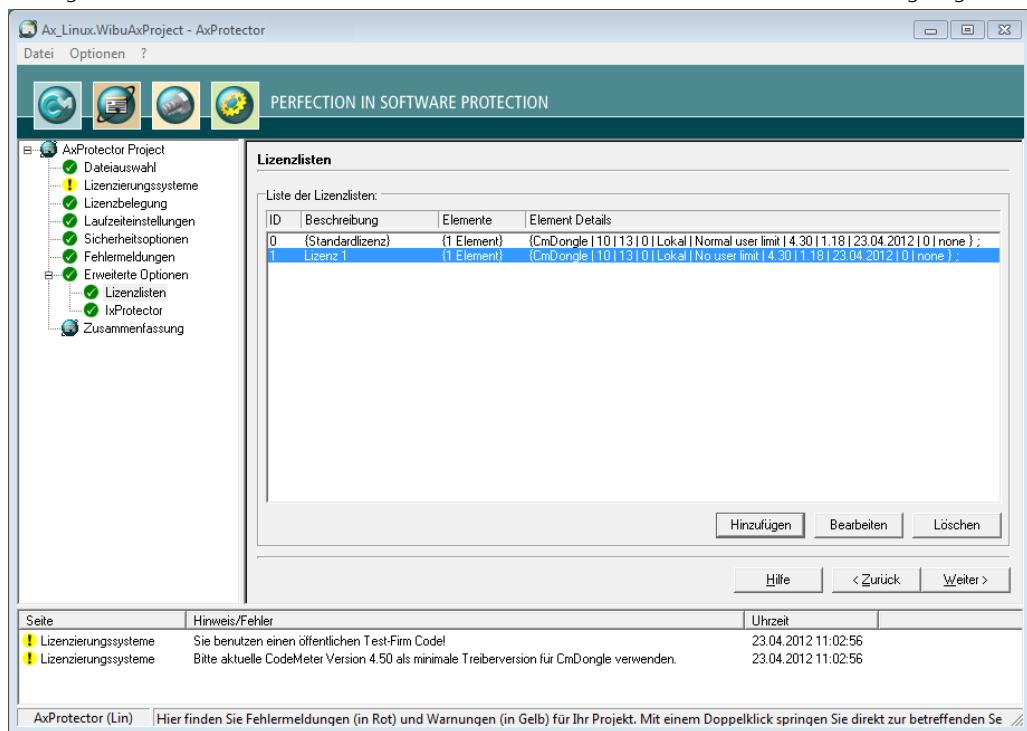


Abbildung 100: AxProtector - Linux "ausgefüllte Lizenzliste"

7.4.5.7.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

-  In diesem Fall werden *CodeMeter®* und *WibuKey API*-Aufrufe über die dynamische Bibliothek (*.d11) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

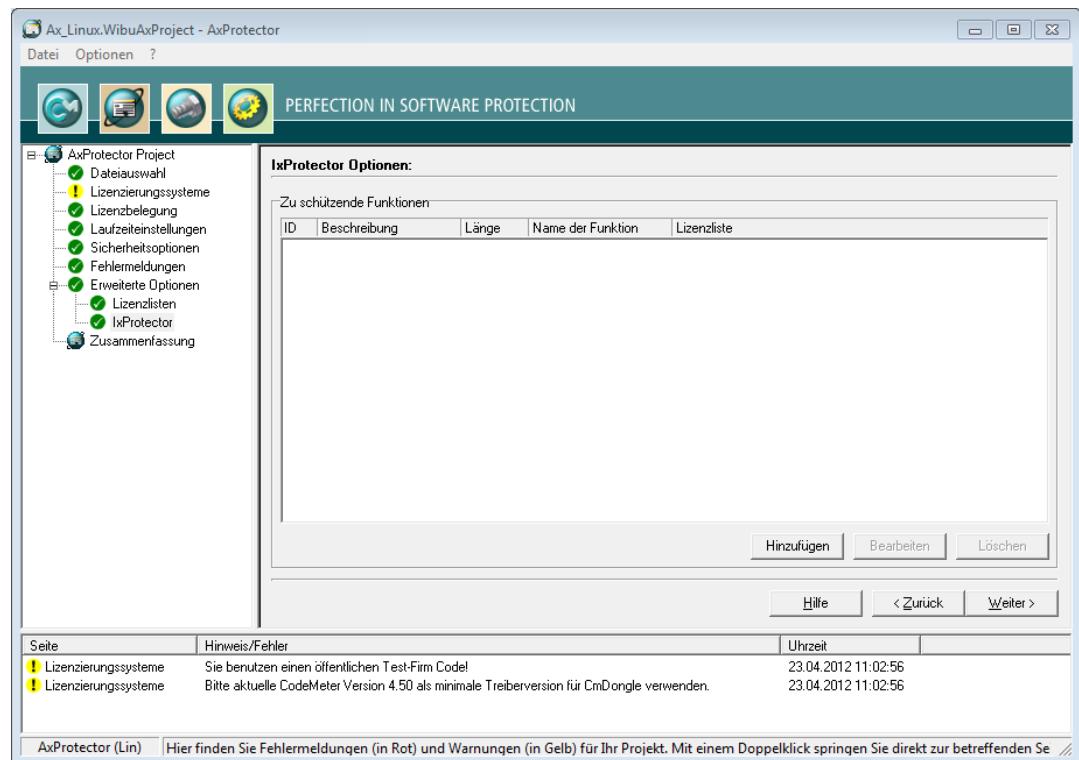
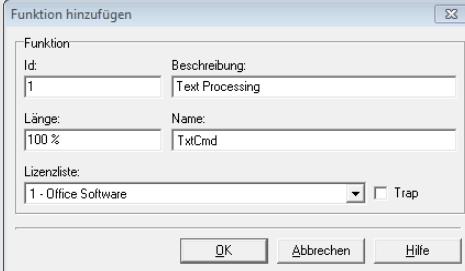


Abbildung 101: AxProtector - Linux - Funktionsliste

Element	Beschreibung
Zu schützende Funktionen	Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor: 1. Betätigen Sie im Bereich IxProtector Optionen die " Hinzufügen " Schaltfläche.

Element	Beschreibung
2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.	
	
Abbildung 102: AxProtector - Linux – Funktion hinzufügen	
Element	Beschreibung
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode³²² und WupiEncryptCode³²² verwenden.</p>
Beschreibung	Beschreibt die Funktion durch einen Texteintrag.
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an.</p> <p>Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>i Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p>
Lizenzliste	Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.
Trap	Aktiviert die Trap-Funktion für die Funktion. Kommandozeilen-Option siehe hier ³⁰⁸ .
3. Betätigen Sie die "OK" Schaltfläche. Die neuen Funktionen werden der Funktionsliste hinzugefügt.	

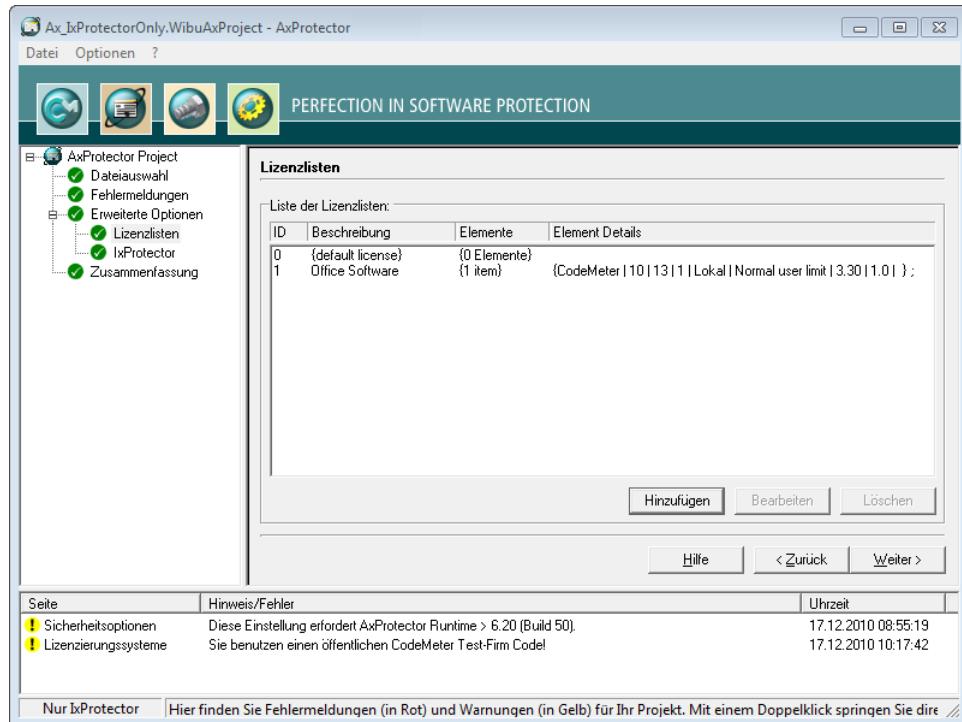


Abbildung 103: AxProtector - Linux "gefüllte Funktionsliste"

7.4.5.8 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf` ³¹⁹.

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

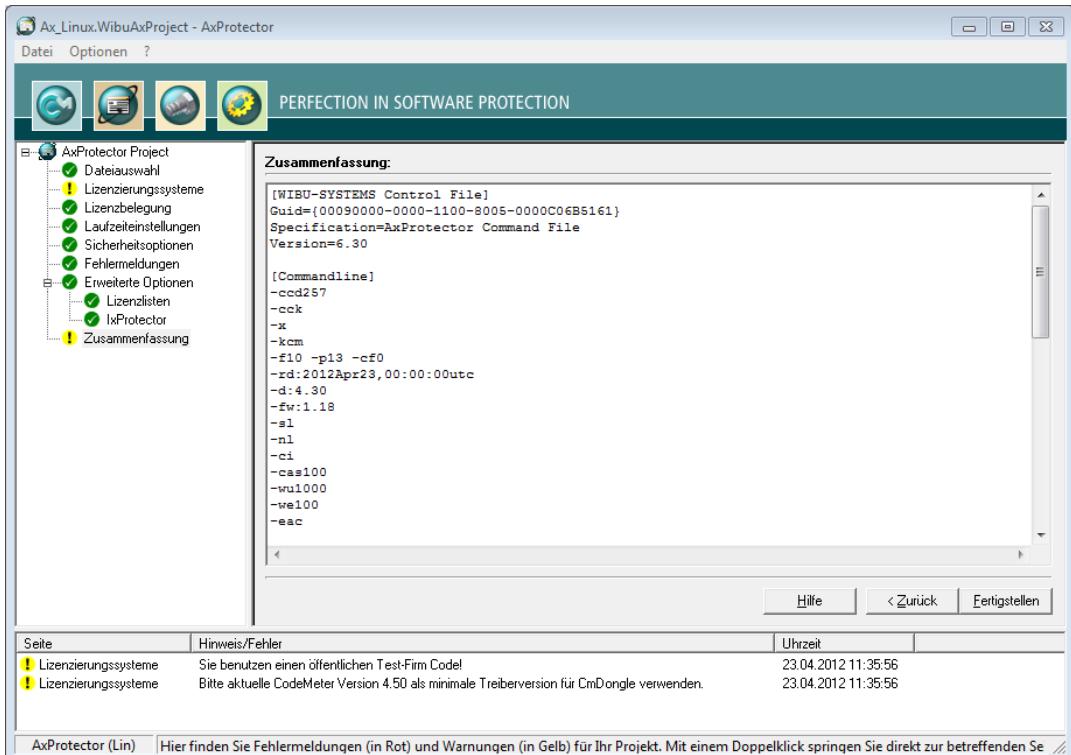


Abbildung 104: AxProtector - Linux "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

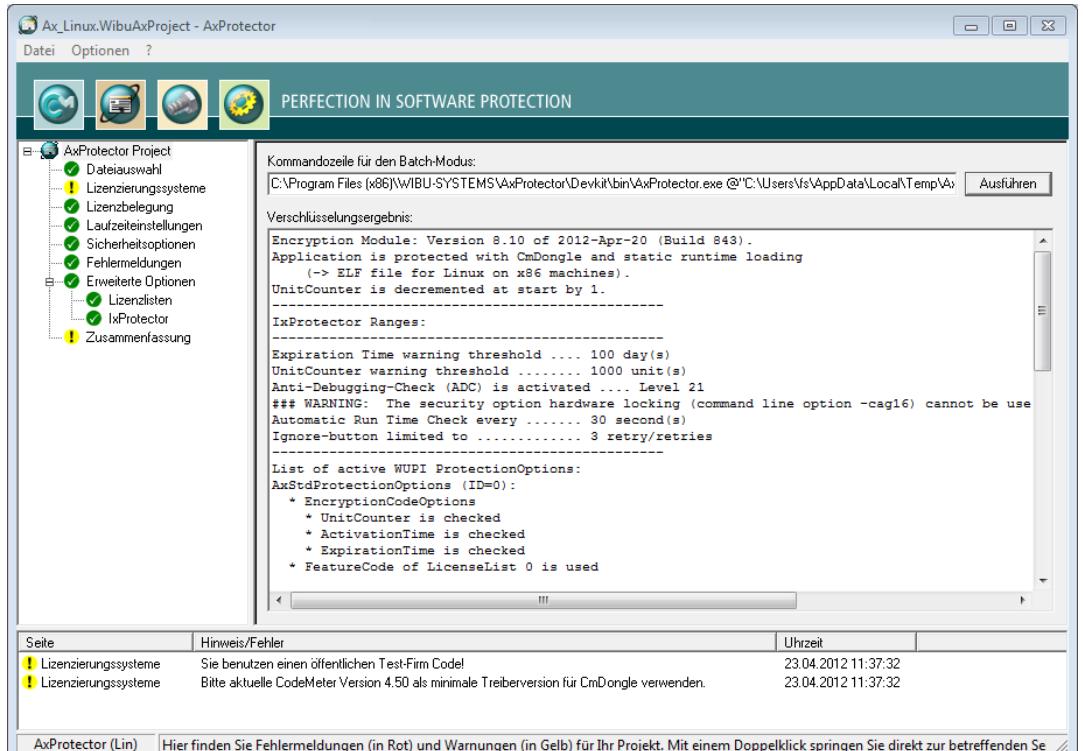


Abbildung 105: AxProtector - Linux "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	<p>Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.5 IxProtector Karteireiter

7.5.1 Windows Anwendung oder DLL

Diesen Projekttyp wählen Sie, wenn Sie eine indexbasierte Verschlüsselung von separaten Funktionen Ihrer Anwendung durchführen wollen, dies aber ohne die gesamte Anwendung noch zusätzlich mit AxProtector zu schützen.

 Wibu-Systems empfiehlt IxProtector jedoch innerhalb von AxProtector zu nutzen, falls keine besonderen Gründe dagegen sprechen.

Mit dieser Option sucht IxProtector dann die betreffenden Code-Bereiche heraus und verschlüsselt diese. Aber selbst im Fall, dass Sie den Projekttyp "Nur IxProtector" wählen, ist eine erhöhte Sicherheit des Schutzes gegeben, da die verwendete Dummy-DLL bei der Verwendung von IxProtector durch statischen Code ersetzt wird. Diese DLL wird später bei der Ausführung der Anwendung nicht mehr benötigt. Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Windows mit AxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Windows Anwendung oder DLL	 IxProtector Windows <small>219</small>		Windows Kommandozeile <small>292</small>

7.5.1.1 Dateiauswahl

Um betreffende Code-Bereiche einer ausführbaren Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

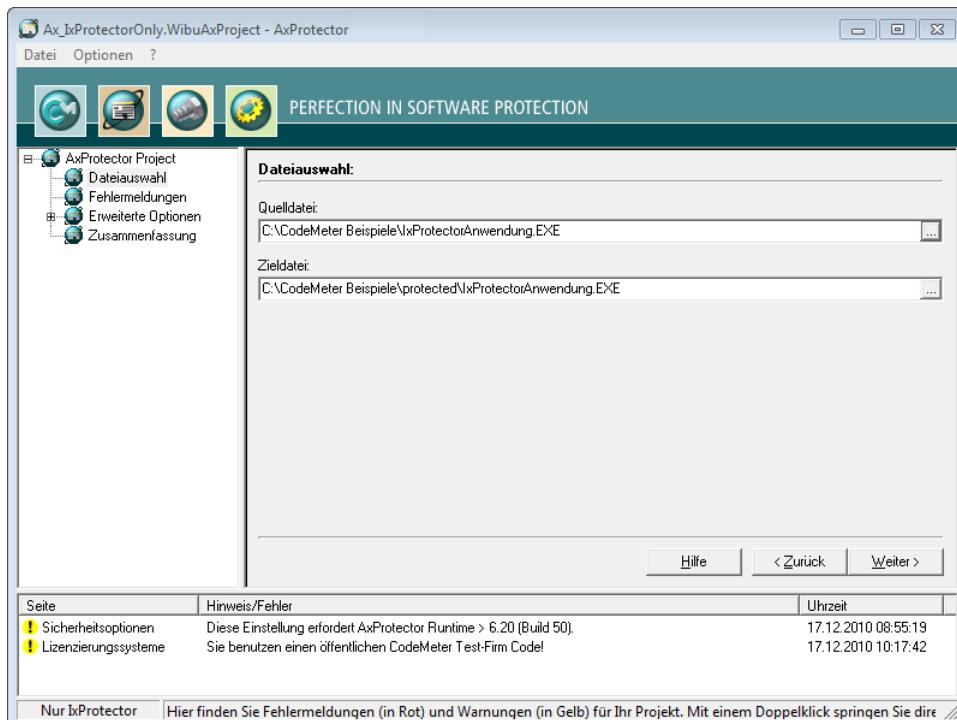


Abbildung 106: IxProtector – Windows "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein. i Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..\protected\..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.5.1.2 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

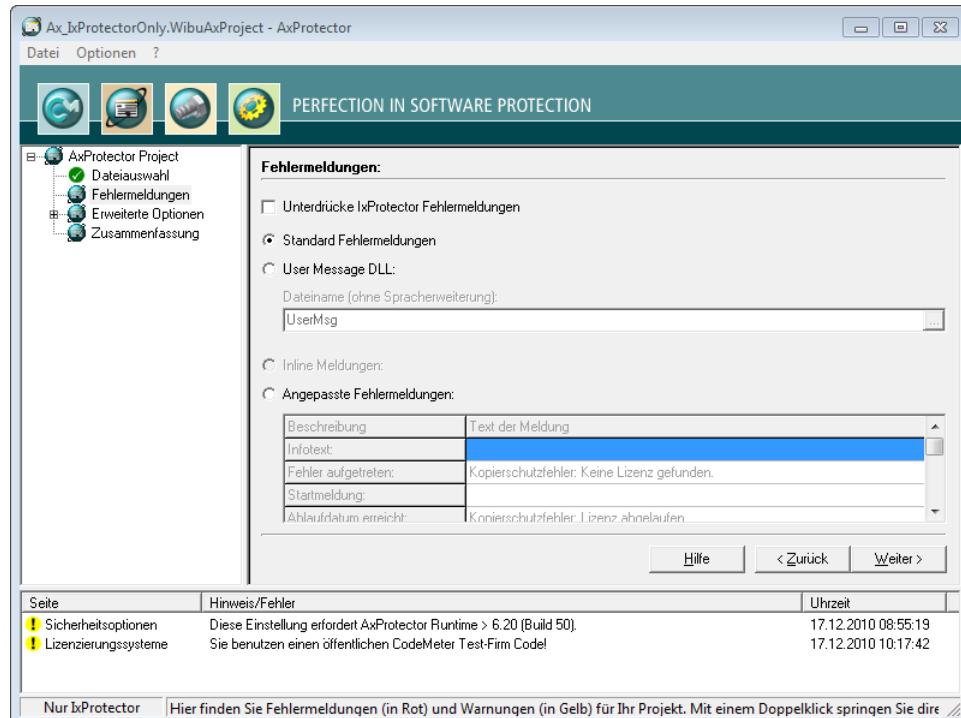
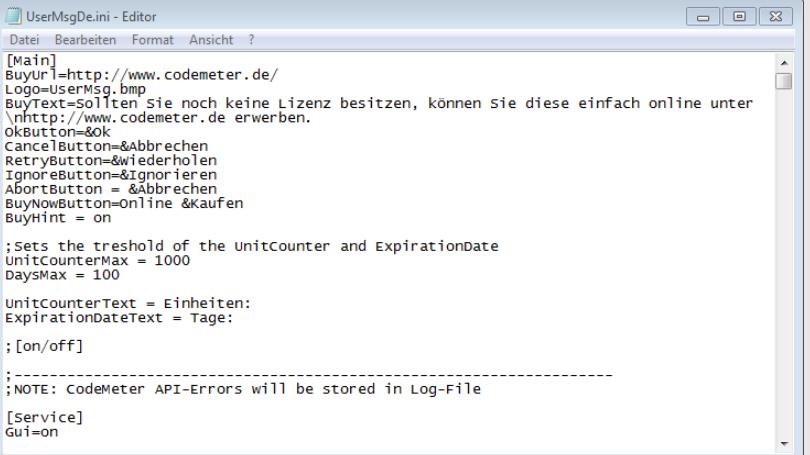


Abbildung 107: IxProtector – Windows "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Unterdrücke IxProtector Fehlermeldungen	Unterdrückt die Ausgabe von IxProtector Fehlermeldungen (Kommandozeilen-Option siehe hier). Setzen Sie diese Option nicht, so werden bei der Verwendung von IxProtector im Fehlerfalle zusätzliche Meldungsfenster angezeigt, und zwar zusätzlich zu den im Projekt selbst ausprogrammierten Meldungen.
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoage ausgegeben (Kommandozeilen-Option siehe hier).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlermeldungen können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen.

Element	Beschreibung
	<p>gen (Kommandozeilen-Option siehe hier³¹²).</p> <p> Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector geschützte Anwendung befindet.</p>  <pre>[Main] BuyUrl=http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten Sie noch keine Lizenz besitzen, können Sie diese einfach online unter \nhttp://www.codemeter.de erwerben. OkButton=&OK CancelButton=&Abbrechen RetryButton=&Wiederholen IgnoreButton=&Ignorieren AbortButton = &Abbrechen BuyNowButton=Online &kaufen BuyHint = on ;sets the threshold of the UnitCounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 UnitCounterText = Einheiten: ExpirationDateText = Tage: ;[on/off] ;NOTE: CodeMeter API-Errors will be stored in Log-File [Service] Gui=on</pre>
	<p>Abbildung 108: AxProtector – UserMsgDe.ini</p> <p>Dateiname (ohne Spracherweiterung)</p> <p>Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an. Die UserMsgDll wird aus dem Verzeichnis %Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage kopiert. Die jeweiligen Initialisierungsdateien sind ebenfalls in diesem Verzeichnis abgelegt.</p>
Inline Meldungen	<p>Link für .NET Projekte eine inline assembly und kann ebenfalls über *.ini-Dateien konfiguriert werden.</p> <p> Diese Option ist nur bei der Verschlüsselung von .NET Anwendungen verfügbar.</p>
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.5.1.3 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

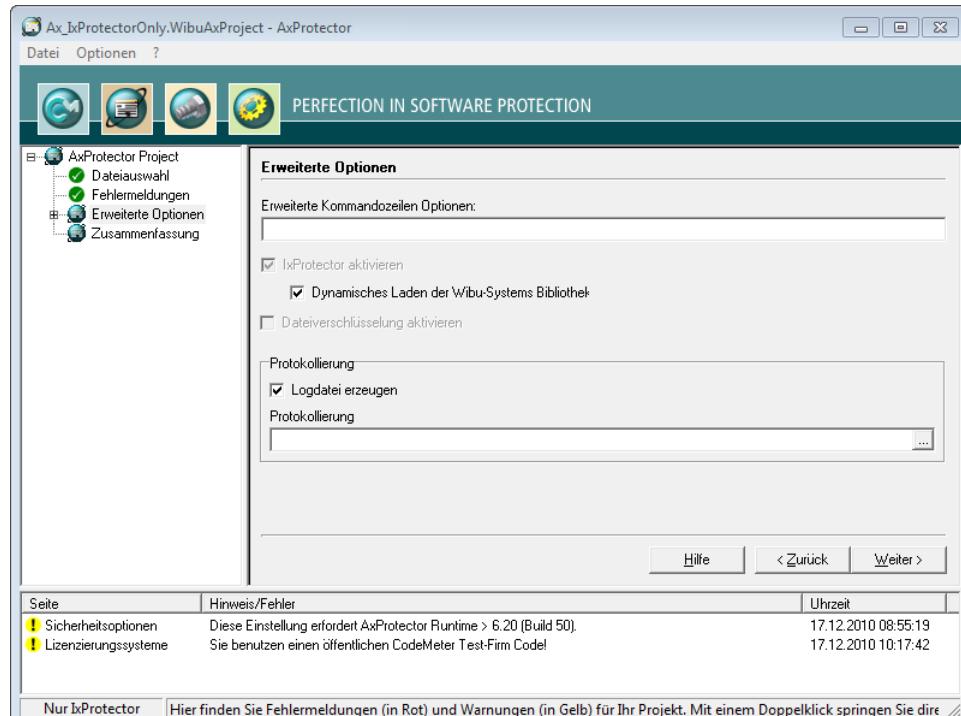


Abbildung 109: IxProtector – Windows "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
Dynamisches Laden der Wibu-Systems Bibliotheken	Das Aktivieren des Auswahlkästchens bewirkt, dass für VB6-Anwendungen oder beim dynamischen Laden von Wibu-Systems Bibliotheken ein besonderes, laufzeitintensives Verfahren eingesetzt wird (Kommandozeilen-Option siehe hier)
Aktivieren WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten aus dem CmContainer, wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.

Element	Beschreibung
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an. i Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.

7.5.1.3.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

i Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

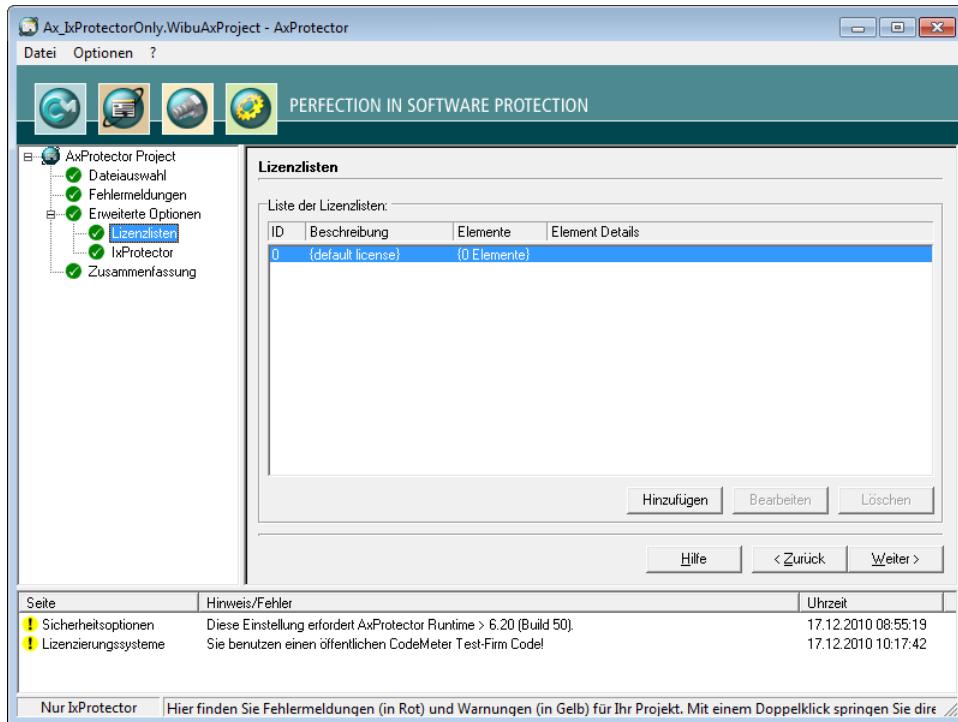
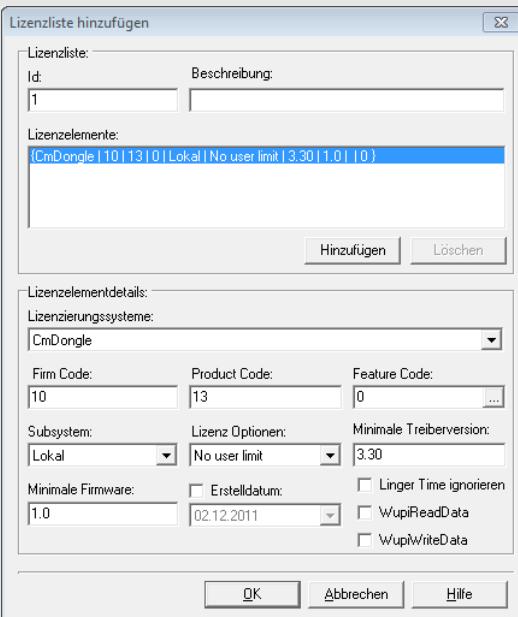


Abbildung 110: IxProtector – Windows "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "Hinzufügen" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	<p>Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.</p> <p> Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.</p>
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag.</p> <p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p>  <p>The dialog shows a list of license elements: [CmDongle 10 13 0 Lokal No user limit 3.30 1.0 0].</p> <p>Lizenzelementdetails:</p> <ul style="list-style-type: none"> Lizenzierungssysteme: CmDongle Firm Code: 10 Product Code: 13 Feature Code: 0 Subsystem: Lokal Lizenz Optionen: No user limit Minimale Treiberversion: 3.30 Minimale Firmware: 1.0 Erstelltdatum: 02.12.2011 Checkboxes: Linger Time ignorieren, WupiReadData, WupiWriteData
	Abbildung 111: AxProtector - Nur lxDestructor "Lizenzlisten hinzufügen"
Lizenzierungs Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (CmDongle, CmActLicense oder WibuKey).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt. Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich. 
Subsystem	Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal). Lizenz Optionen Auswahl der Lizenz Optionen zur Belegung von Lizenzen: <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	Ab der Firmware-Version 1.18 unterstützt <i>CodeMeter®</i> die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen. Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor: <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenzeigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

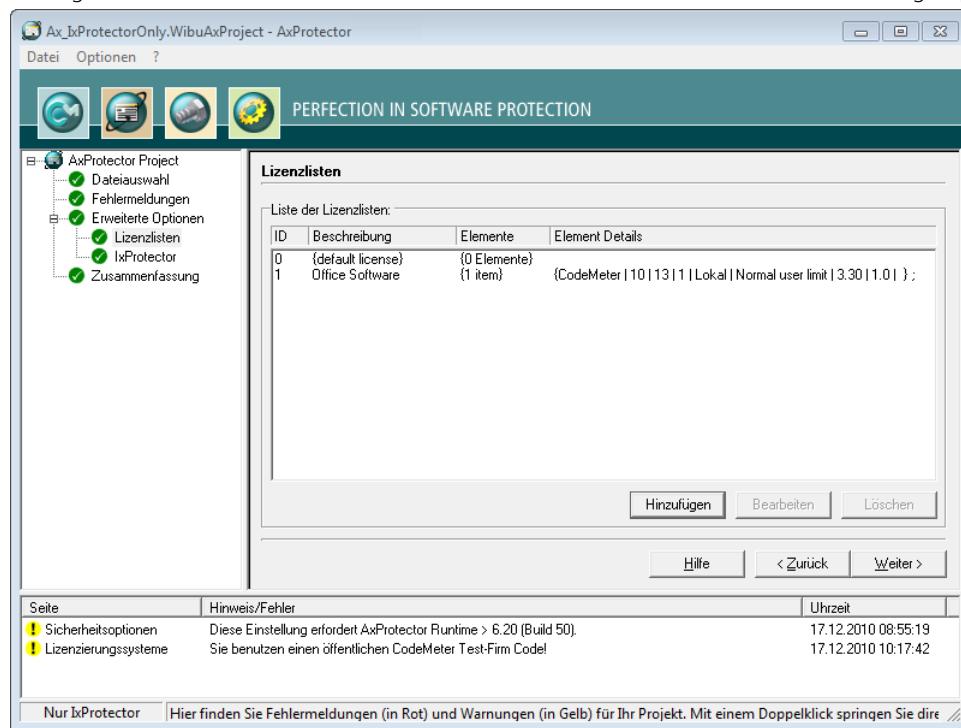


Abbildung 112: AxProtector - Nur *IxProtector* "ausgefüllte Lizenzliste"

7.5.1.3.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

 In diesem Fall werden *CodeMeter®* und *WibuKey API*-Aufrufe über die dynamische Bibliothek (*.d11) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

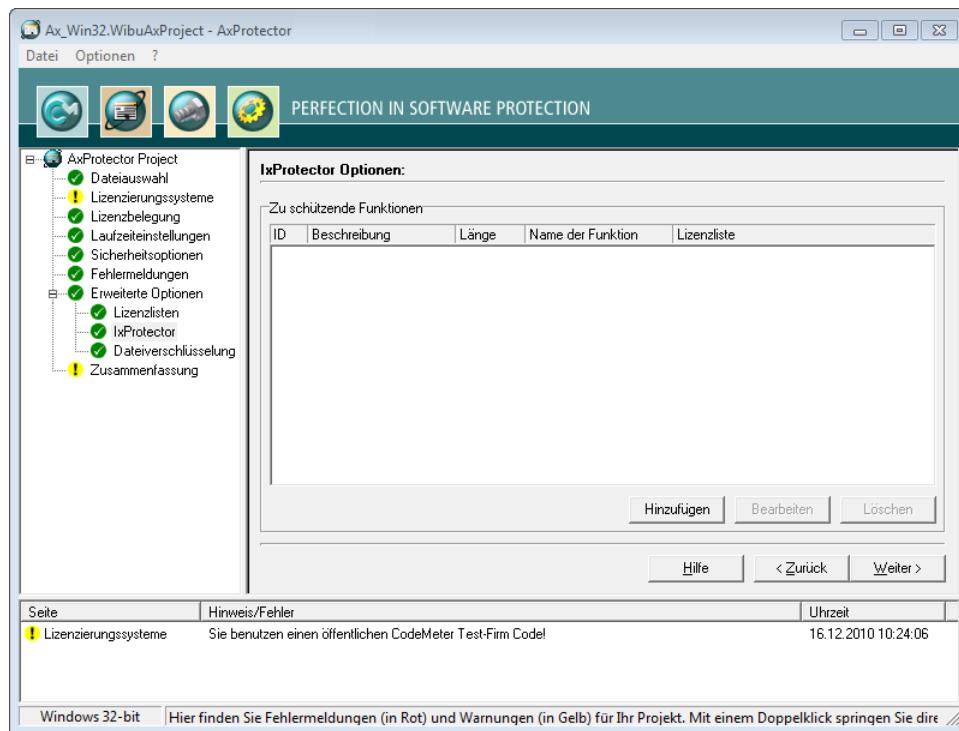
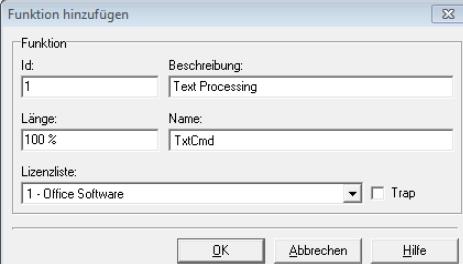


Abbildung 113: *IxProtector – Windows "Funktionsliste"*

Element	Beschreibung
Zu schützende Funktionen	Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor: 1. Betätigen Sie im Bereich IxProtector Optionen die " Hinzufügen " Schaltfläche.

Element	Beschreibung
	<p>2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.</p> 
	<p>Abbildung 114: IxProtector – Windows "Funktion hinzufügen"</p>
Element	Beschreibung
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode³²² und WupiEncryptCode³²² verwenden.</p>
Beschreibung	<p>Beschreibt die Funktion durch einen Texteintrag.</p>
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an. Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p>
Lizenzliste	<p>Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.</p>
Trap	<p>Aktiviert die Trap-Funktion für die Funktion. Kommandozeilen-Option siehe hier³⁰⁸.</p>

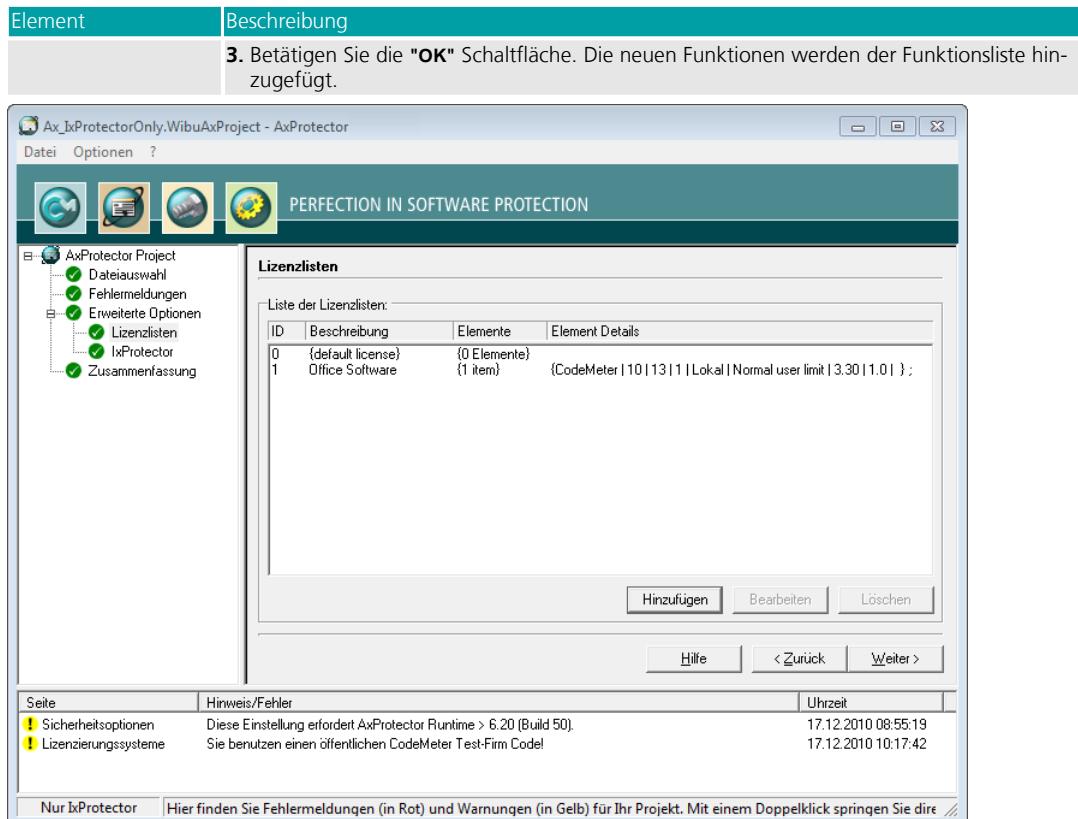


Abbildung 115: IxProtector – Windows "gefüllte Funktionsliste"

7.5.1.4 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (VIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`

319

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

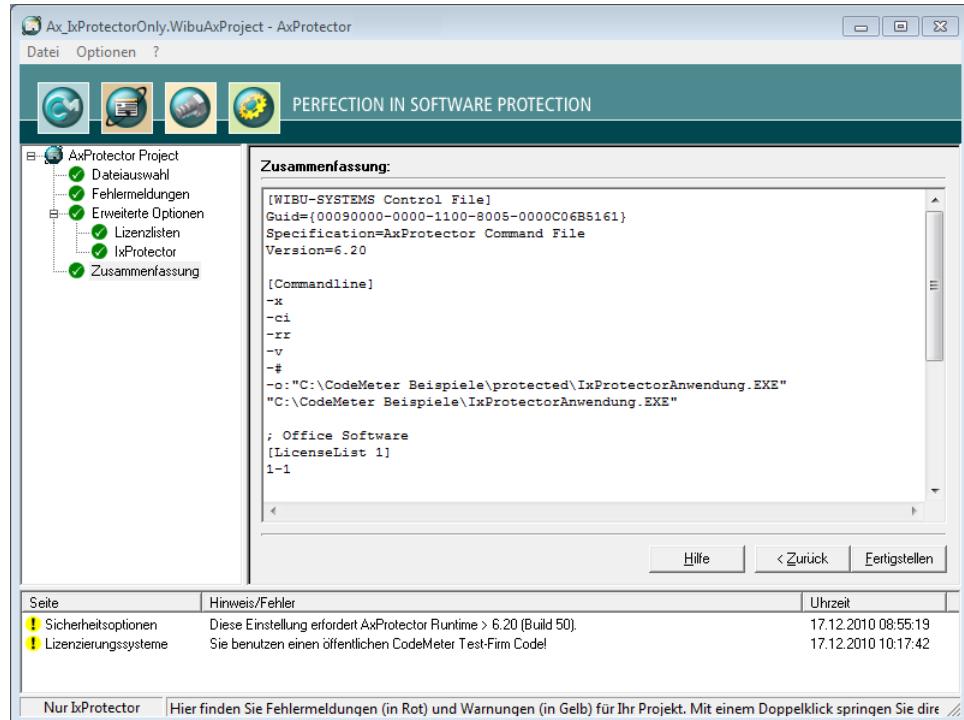


Abbildung 116: IxProtector – Windows "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

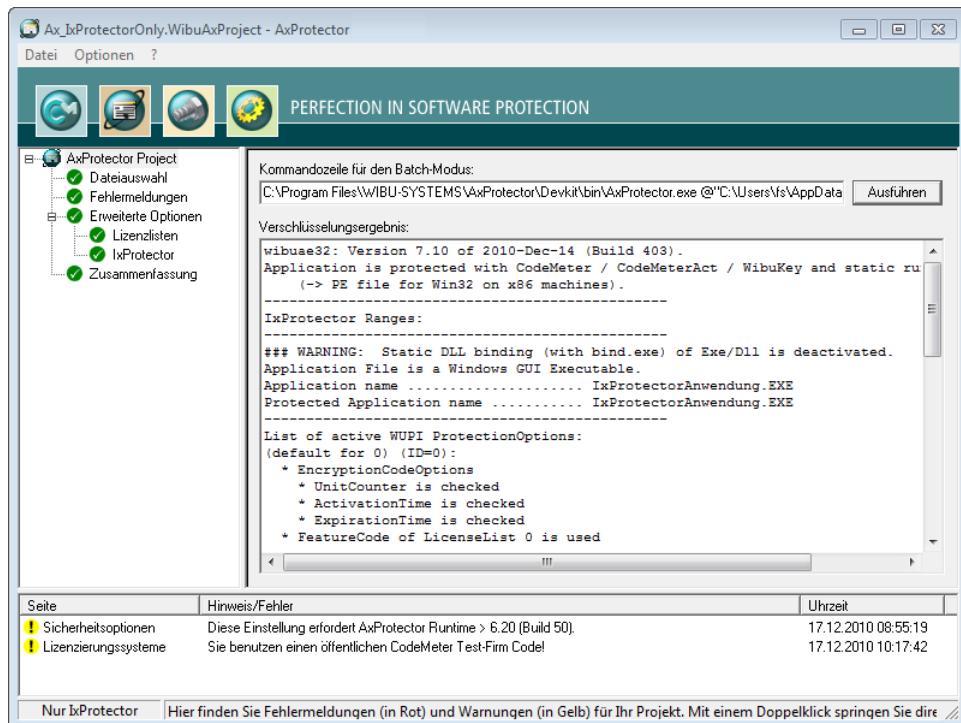


Abbildung 117: IxProtector – Windows "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die " Ausführen " Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt. <div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">  Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.5.2 .NET Assembly

Diesen Projekttyp wählen Sie, wenn Sie eine indexbasierte Verschlüsselung von separaten Funktionen Ihrer Anwendung durchführen wollen, dies aber ohne die gesamte Anwendung noch zusätzlich mit AxProtector zu schützen.



Wibu-Systems empfiehlt *IxProtector* jedoch innerhalb von *AxProtector* zu nutzen, falls keine besonderen Gründe dagegen sprechen.

Mit dieser Option sucht *IxProtector* dann die betreffenden Code-Bereiche heraus und verschlüsselt diese. Aber selbst im Fall, dass Sie den Projekttyp "Nur *IxProtector*" wählen, ist eine erhöhte Sicherheit des Schutzes gegeben, da die verwendete Dummy-DLL bei der Verwendung von *IxProtector* durch statischen Code ersetzt wird. Diese DLL wird später bei der Ausführung der Anwendung nicht mehr benötigt. Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für .NET mit *IxProtector* verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
.NET Assembly	IxProtector .NET ²³³	✓	.NET Kommandozeile ²³²

7.5.2.1 Dateiauswahl

Um betreffende Code-Bereiche einer ausführbaren Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

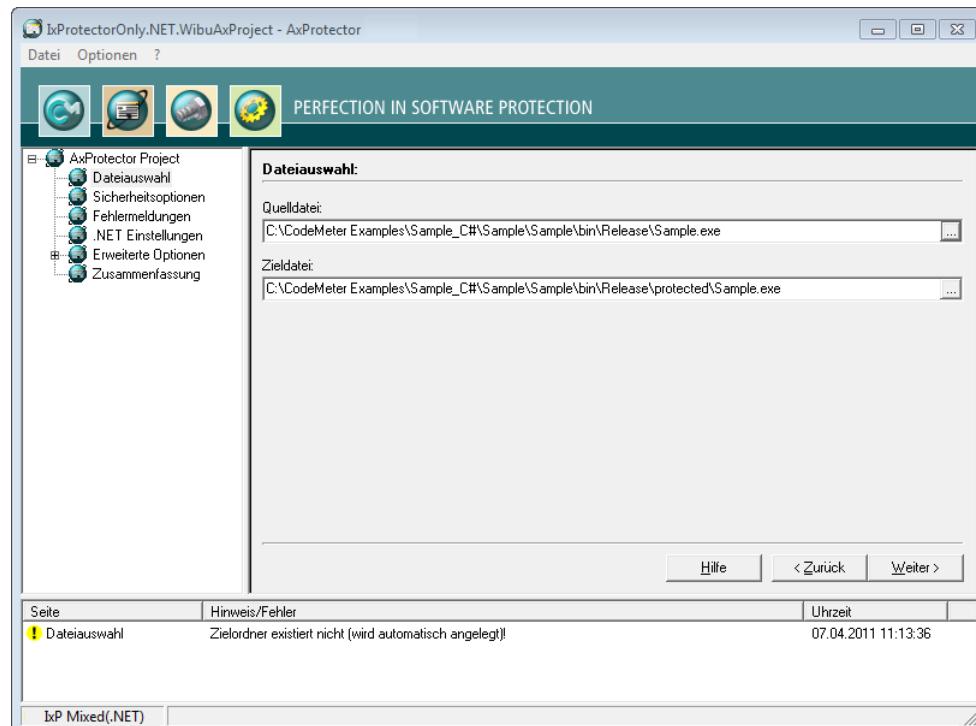


Abbildung 118: IxProtector - NET "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsselnde Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein.  Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [.. \protected \..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier .

7.5.2.2 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

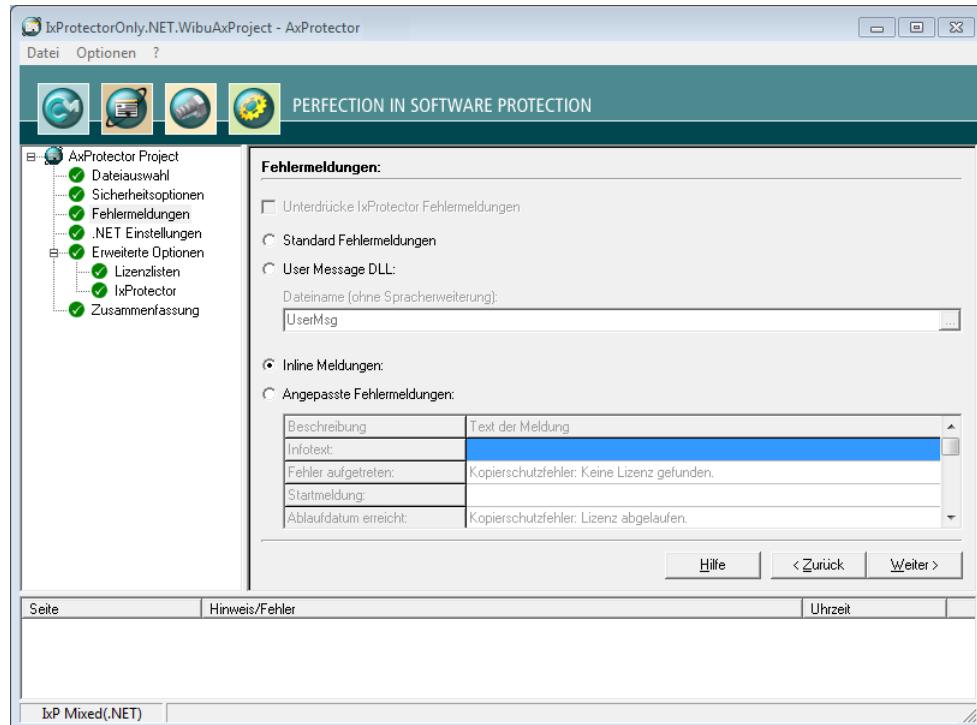
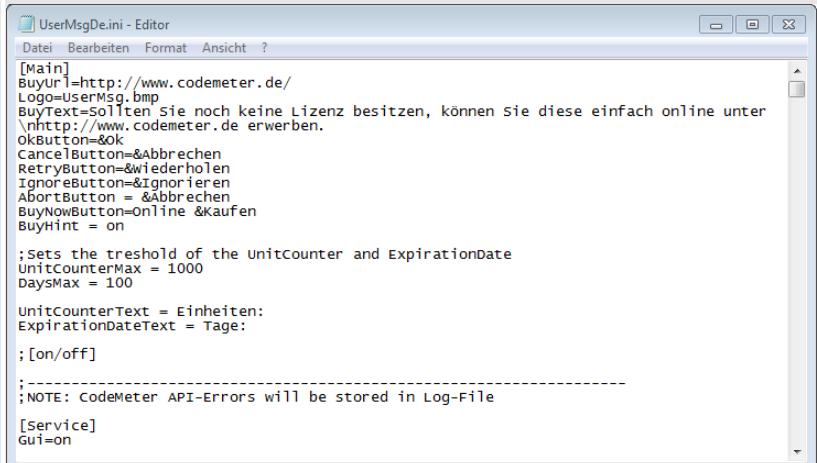


Abbildung 98: IxProtector - NET "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlermeldungen können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen (Kommandozeilen-Option siehe hier ³¹²). <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector geschützte Anwendung befindet. </div>

Element	Beschreibung
	 <pre> [UserMsgDe.ini - Editor] Datei Bearbeiten Format Ansicht ? [Main] Buyurl=http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten sie noch keine Lizenz besitzen, können sie diese einfach online unter \nhttp://www.codemeter.de erwerben. OkButton=&OK CancelButton=&Abbrechen RetryButton=&Wiederholen IgnoreButton=&Ignorieren AbortButton = &Abbrechen BuyNowButton=Online &kaufen BuyHint = on ;sets the threshold of the UnitCounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 UnitCounterText = Einheiten: ExpirationDateText = Tage: ;[on/off] ;-----; ;NOTE: CodeMeter API-Errors will be stored in Log-File [Service] Gui=on </pre>
	<p>Abbildung 119: AxProtector - UserMsgDe.ini</p> <p>Dateiname (ohne Spracherweiterung)</p> <p>Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an. Die UserMsgDll wird aus dem Verzeichnis %Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin\UserMessage kopiert. Die jeweiligen Initialisierungsdateien sind ebenfalls in diesem Verzeichnis abgelegt.</p> <p>Inline Meldungen</p> <p>Linkt für .NET Projekte eine inline assembly und kann ebenfalls über *.ini-Dateien konfiguriert werden (Kommandozeilen-Option siehe hier <small>312</small>).</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Bei Verwendung der Inline UserMessages erfolgt das Logging in das Verzeichnis "%CommonApplicationData%". Ein anderer Pfad kann in der *.INI-Datei mit LoggingPath=<Pfad> angegeben werden </div> <p>Anangepasste Fehlermeldungen</p> <p>Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.</p>

7.5.2.3 .NET Einstellungen

Auf dieser Eingabeseite haben Sie die Möglichkeit, zusätzliche Einstellungen für die .NET-Verschlüsselung zu setzen.

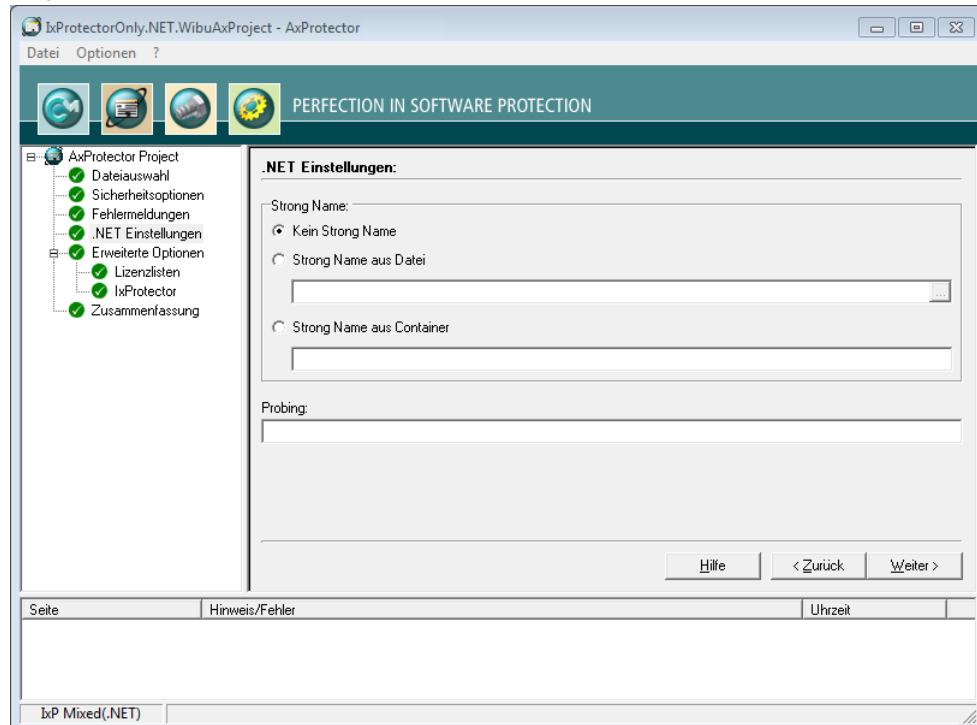


Abbildung 120: *lxDprotector - NET ".NET Einstellungen"*

Hier stellen Sie ein, ob Ihre Assembly von *AxProtector* signiert wird.

Element	Beschreibung
Kein Strong Name	Die Assembly wird nicht signiert.
Strong Name aus Datei	Zur Signierung der Programmklasse, können Sie hier eine Datei als Quelle für das Schlüsselpaar zur Generierung eines Strong Names angeben (Kommandozeilen-Option siehe hier ³¹²).
Strong Name aus Container	Geben Sie hier den Namen des Containers zur Signierung Ihrer Programmklasse an (Kommandozeilen-Option siehe hier ³¹²).
Probing	<p>Der Bereich erlaubt Angaben zum Lageort für signierte Programmklassen in einer <code>app.config</code> Datei.</p> <p> Geben Sie den Pfad an, auf dem der Zugriff auf die Programmklasse erfolgt, getrennt durch ";". Oder geben Sie die jeweilige <code>app.config</code> Datei an.</p> <p>Kommandozeilen-Option siehe hier³¹².</p>

7.5.2.4 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

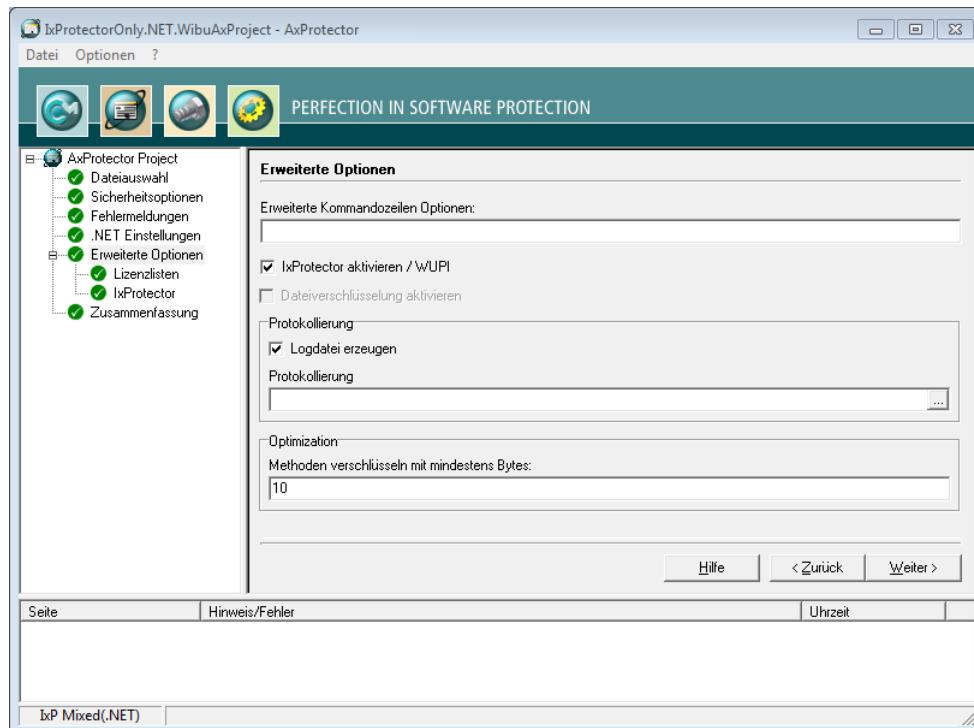


Abbildung 121: *IxProtector - .NET "Erweiterte Optionen"*

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
IxProtector aktivieren	Das Aktivieren des Auswahlkästchens lässt nachfolgend das Anlegen und Bearbeiten von Lizenzlisten und Funktionslisten zu, die Sie beim modularen Schutz Ihrer Anwendung mit IxProtector über das Softwareschutz-API (WUPI) ³²⁰ verwenden.
Aktivieren WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateinamen dieser Protokolldatei an.

Element	Beschreibung
	 Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.
Optimierung	Geben Sie hier zu Optimierungszwecken ein, welche Mindestgröße ein Assembly besitzen muss, damit es verschlüsselt wird. Die Standard-Einstellung beträgt 10 Bytes (Kommandozeilen-Option siehe hier ³⁰⁶).

7.5.2.4.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *IxProtector* über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

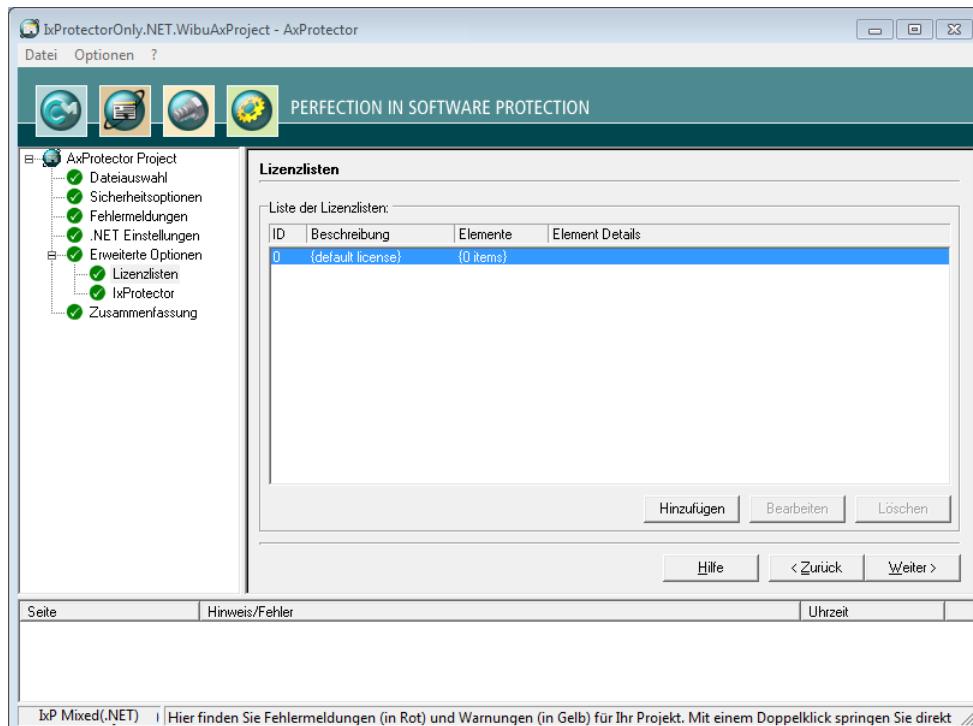
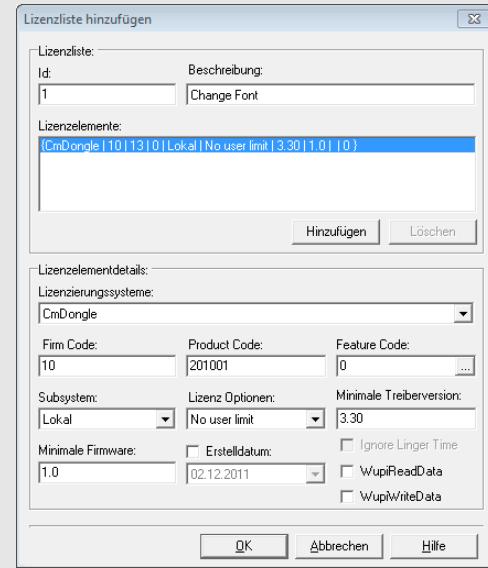


Abbildung 122: IxProtector - .NET "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung. i Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.
Beschreibung	Beschreibt die Lizenzliste über einen Texteintrag.

Element	Beschreibung
	<p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (<i>CmDongle</i> , <i>CmActLicense</i> oder <i>WibuKey</i>).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.
Feature Code	Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt. Über die "... " Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.



Element	Beschreibung
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenz suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal).</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenz:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversi-on	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

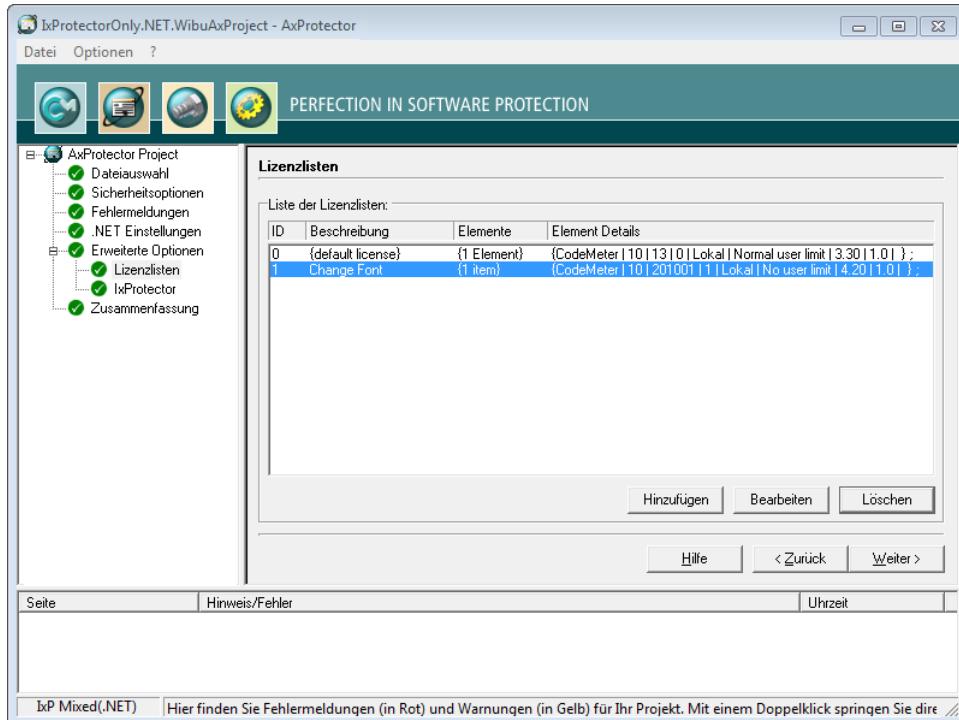


Abbildung 124: AxProtector - Nur IxProtector (.NET) "ausgefüllte Lizenzliste"

7.5.2.4.2 IxProtector

Über diesen Menü-Eintrag definieren Sie Verschlüsselungstypen für einzelne Assembly-Elemente. Haben Sie im Menü-Eintrag "**Erweiterte Optionen**" den Eintrag "**IxProtector**" aktiviert, so wird die Quell-Assembly geladen und die gesamte Baumstruktur aus Namespaces, Klassen und Modulen ist verfügbar und wird angezeigt.

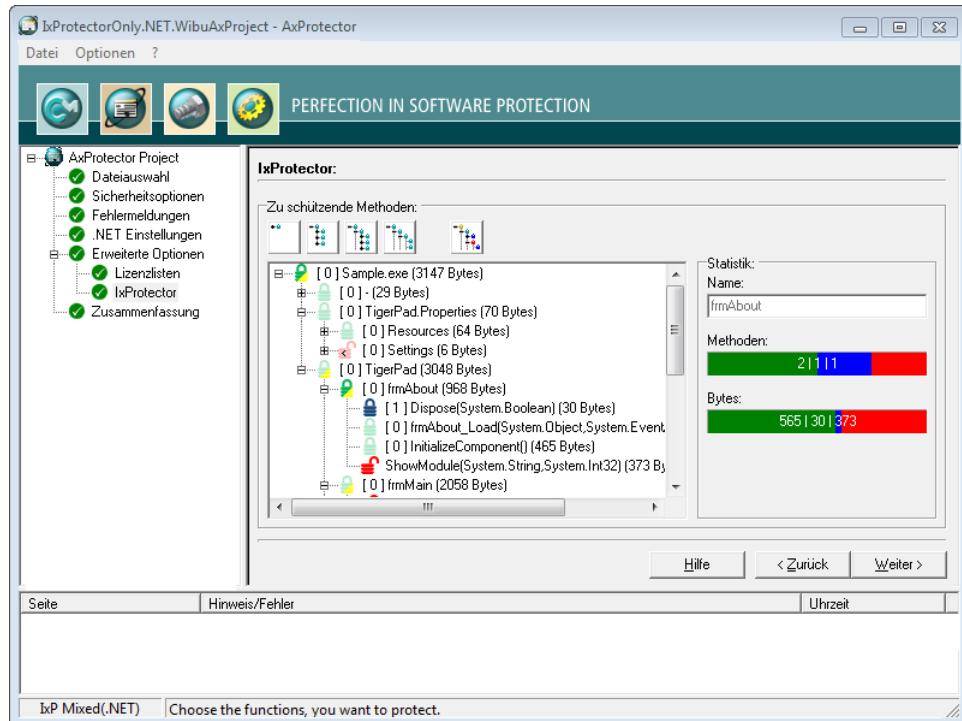


Abbildung 125: IxProtector - NET "IxProtector"

Sie können aus verschiedenen Assembly-Ansichten wählen, indem Sie die oberen Schaltflächen im IxProtector-Bereich benutzen.

Ansichten

Schaltfläche	Beschreibung
	klappt alle Ebenen der Assembly in der Baumstruktur zu.
	klappt die Namespace-Ebene der Assembly auf.
	klappt die Klassen-Ebene der Assembly auf.
	klappt die Methoden-Ebene der Assembly auf.
	klappt alle Ebenen der Assembly auf, für die Änderungen vorgenommen worden sind.

Der Statistik-Bereich auf der rechten Seite zeigt Ihnen weitere Verschlüsselungsdetails in Abhängigkeit der Auswahl, die Sie in der Baumansicht getroffen haben.

Element	Beschreibung
Name	Dieses Feld verweist auf den Namen des Elementes, das Sie in der Baumansicht ausgewählt haben.

Element	Beschreibung	
Methoden	Farbe	Beschreibung
	Grün	zeigt an, dass die Methode mit <i>AxProtector</i> verschlüsselt wird und die Lizenzlisten ID einen Wert von 0 hat (die Default-Lizenz).
	Blau	zeigt an, dass die Methode mit <i>IxProtector</i> verschlüsselt wird und die Lizenzlisten ID einen Wert ungleich 0 hat.
	Rot	zeigt an, dass die Methode unverschlüsselt ist.
Bytes	Farbe	Beschreibung
	Grün	zeigt an, dass die Methode mit <i>AxProtector</i> verschlüsselt wird und die Lizenzlisten ID einen Wert von 0 hat (die Default-Lizenz).
	Blau	zeigt an, dass die Methode mit <i>IxProtector</i> verschlüsselt wird und die Lizenzlisten ID einen Wert ungleich 0 hat.
	Rot	zeigt an, dass die Methode unverschlüsselt ist.

Sie besitzen auch die Möglichkeit, die Schutztechnologien *AxProtector* und *IxProtector* separat einzelnen Assembly-Elementen zuzuweisen bzw. Elemente von der Verschlüsselung auszuschließen. Diese Zuweisung nehmen Sie über ein Kontextmenü wie folgt vor:

1. Wählen Sie das gewünschte Assembly-Element (Namespace, Klasse oder Methode) aus der links stehenden Baumstruktur.
2. Klicken Sie die rechte Maustaste.
Das Kontextmenü öffnet sich.
3. Weisen Sie über die Symbole den gewünschten Verschlüsselungstypen zu.

Die zur Auswahl angebotenen Lizenzlisten IDs richten sich automatisch nach den Einträgen, die Sie der Lizenzliste hinzugefügt haben.

Symbol	Beschreibung
	schließt das gewählte Element aus der Verschlüsselung aus.
	verschlüsselt das gewählte Element mit <i>AxProtector</i> (Lizenzlisten ID mit einem Wert von 0, d.h. der Default-Lizenz).
	verschlüsselt das gewählte Element mit <i>IxProtector</i> (Lizenzlisten ID mit einem Wert von ungleich 0, d.h. nach vorhandenen Lizenzlisten-Einträgen).
	zeigt an, dass diese Methode aufgrund der Größe von der Verschlüsselung ausgenommen wird. Die Schwelle für die Methodengröße können Sie auf der Seite 'Erweiterte Einstellungen' unter Optimierung setzen.

Die Änderungen werden sofort im linken Bereich angezeigt.

7.5.2.5 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

 Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile AxProtector.exe @*.wbc [auf](#)

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

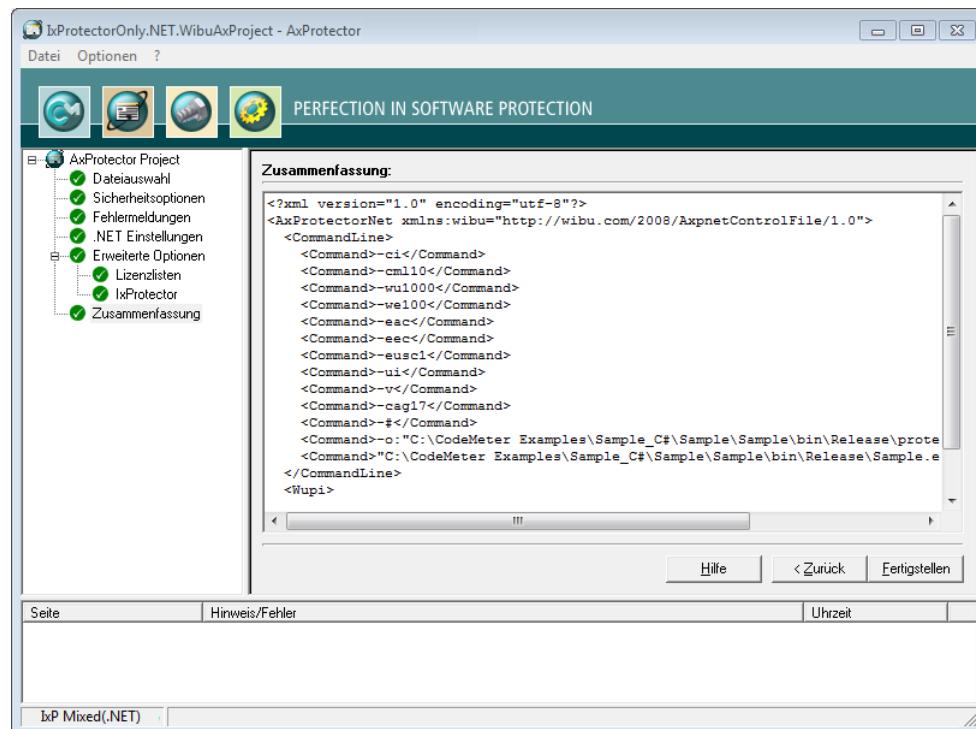


Abbildung 126: IxProtector - NET "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster an-

gezeigt.

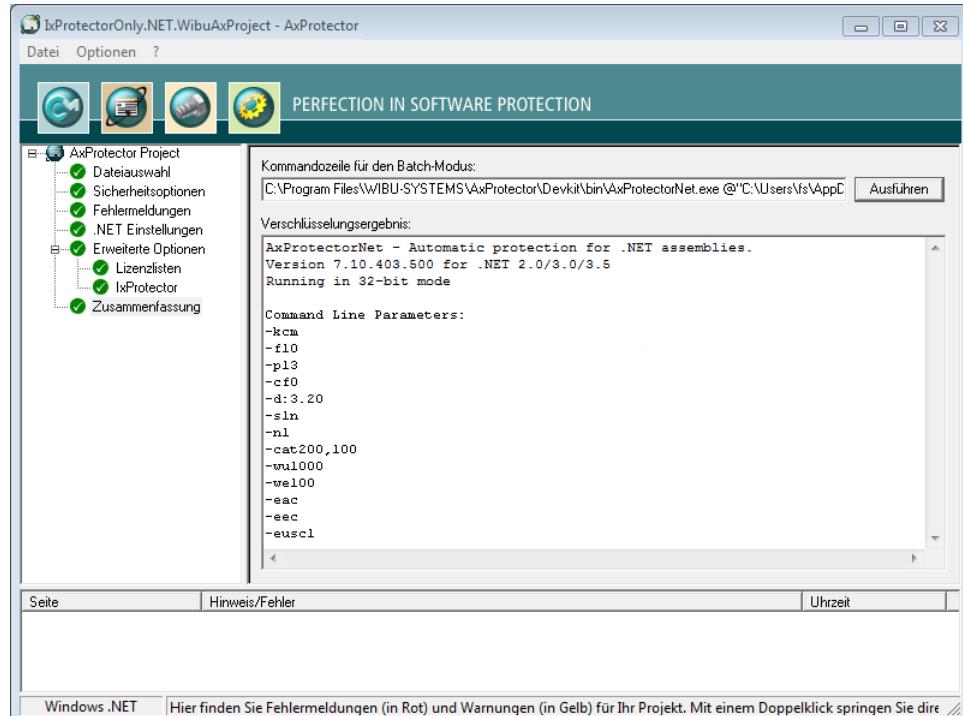


Abbildung 127: IxProtector - NET "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt. i Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen.

7.5.3 Max OS X Anwendung oder Dylib

Diesen Projekttyp wählen Sie, wenn Sie eine indexbasierte Verschlüsselung von separaten Funktionen Ihrer Anwendung durchführen wollen, dies aber ohne die gesamte Anwendung noch zusätzlich mit AxProtector zu schützen.



Wibu-Systems empfiehlt IxProtector jedoch innerhalb von AxProtector zu nutzen, falls keine besonderen Gründe dagegen sprechen.

Mit dieser Option sucht IxProtector dann die betreffenden Code-Bereiche heraus und verschlüsselt diese. Aber selbst im Fall, dass Sie den Projekttyp "Nur IxProtector" wählen, ist eine erhöhte Sicherheit des Schutzes gegeben, da die verwendete Dummy-DLL bei der Verwendung von IxProtector durch statischen Code ersetzt wird. Diese DLL wird später bei der Ausführung der Anwendung nicht mehr benötigt.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Mac OS X mit IxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Mac OS X Anwendung oder Dylib	IxProtector Linux <small>▷ 262</small>		Windows Kommandozeile <small>▷ 292</small> In einer separaten Kommandozeile für Mac, die auf Mac OS X-Betriebssystemen läuft, können Sie ebenfalls Verschlüsselungsparameter <small>▷ 260</small> eingeben.

7.5.3.1 Dateiauswahl

Um betreffende Code-Bereiche einer ausführbaren Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

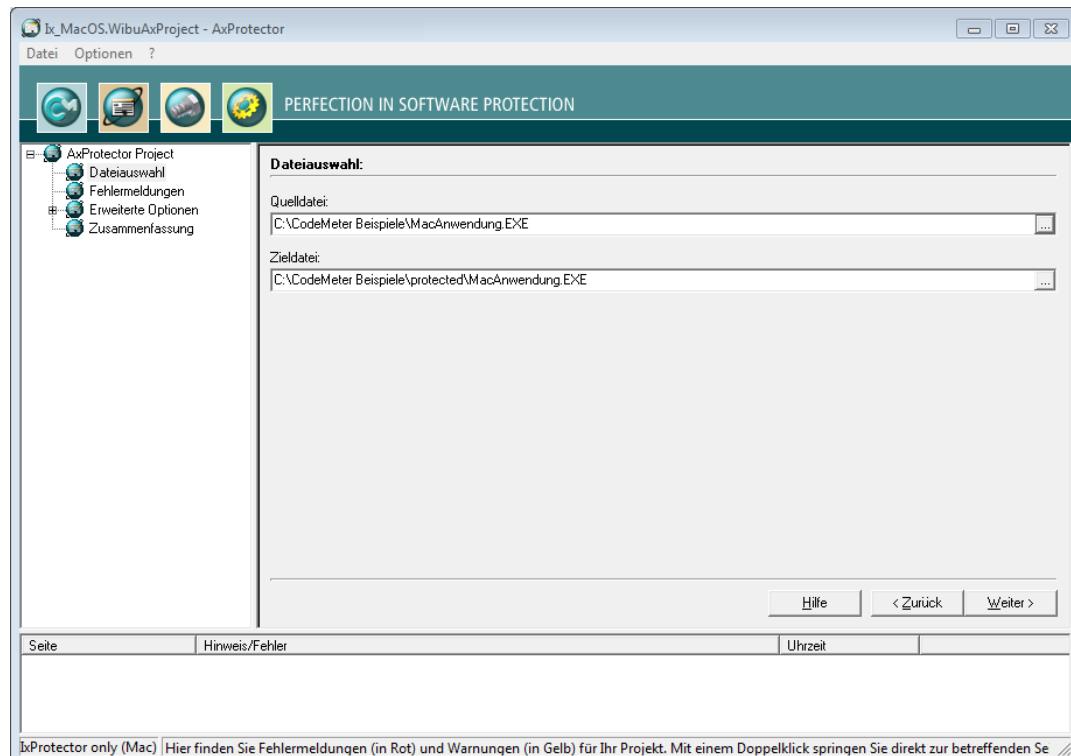


Abbildung 128: IxProtector – Mac OS "Dateiauswahl"

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den " Öffnen " Systemdialog die zu verschlüsseln-de Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> i Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen. </div>
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..]\protected\..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.5.3.2 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

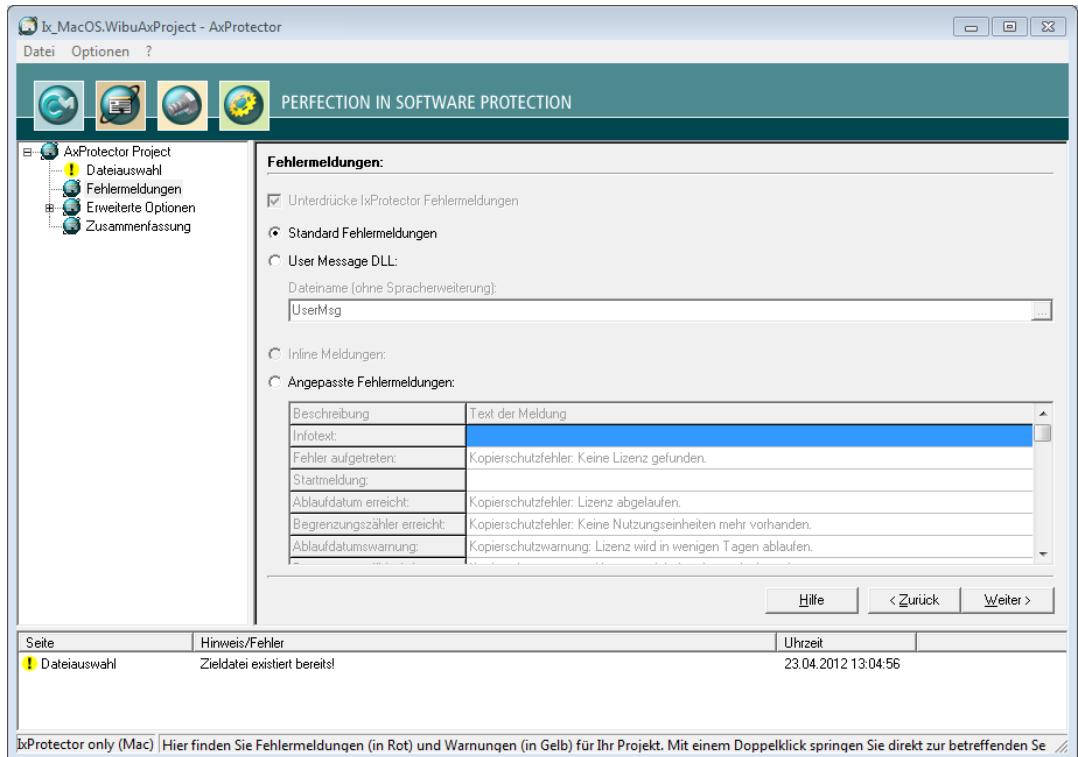


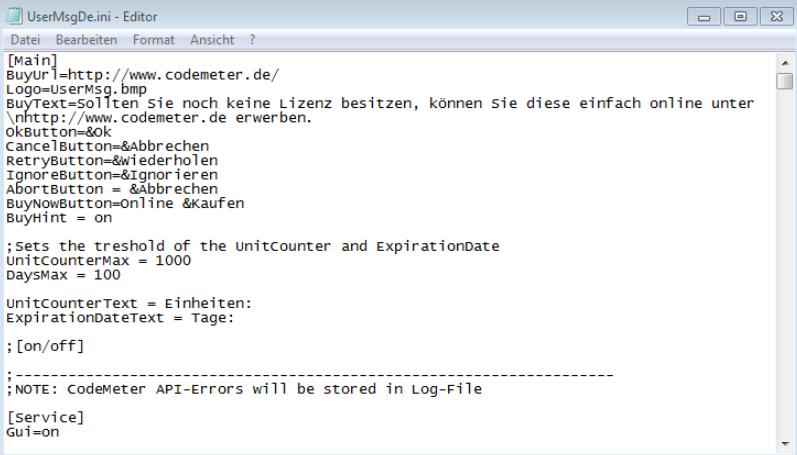
Abbildung 129: IxProtector – Mac OS "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlerbeschreibungen können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupfliegen (Kommandozeilen-Option siehe hier ³¹²).



Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector geschützte Anwendung befindet.

Element	Beschreibung
	 <pre> [UserMsgDe.ini - Editor] Datei Bearbeiten Format Ansicht ? [Main] BuyUrl=http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten sie noch keine Lizenz besitzen, können Sie diese einfach online unter \nhttp://www.codemeter.de erwerben. OkButton=&OK CancelButton=&Abbrechen RetryButton=&Wiederholen IgnoreButton=&Ignorieren AbortButton = &Abbrechen BuyNowButton=Online &kaufen BuyHint = on ;Sets the threshold of the unitcounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 UnitCounterText = Einheiten: ExpirationDateText = Tage: ; [on/off] ;----- ;NOTE: CodeMeter API-Errors will be stored in Log-File [service] Gui=On </pre>
Angepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.5.3.3 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

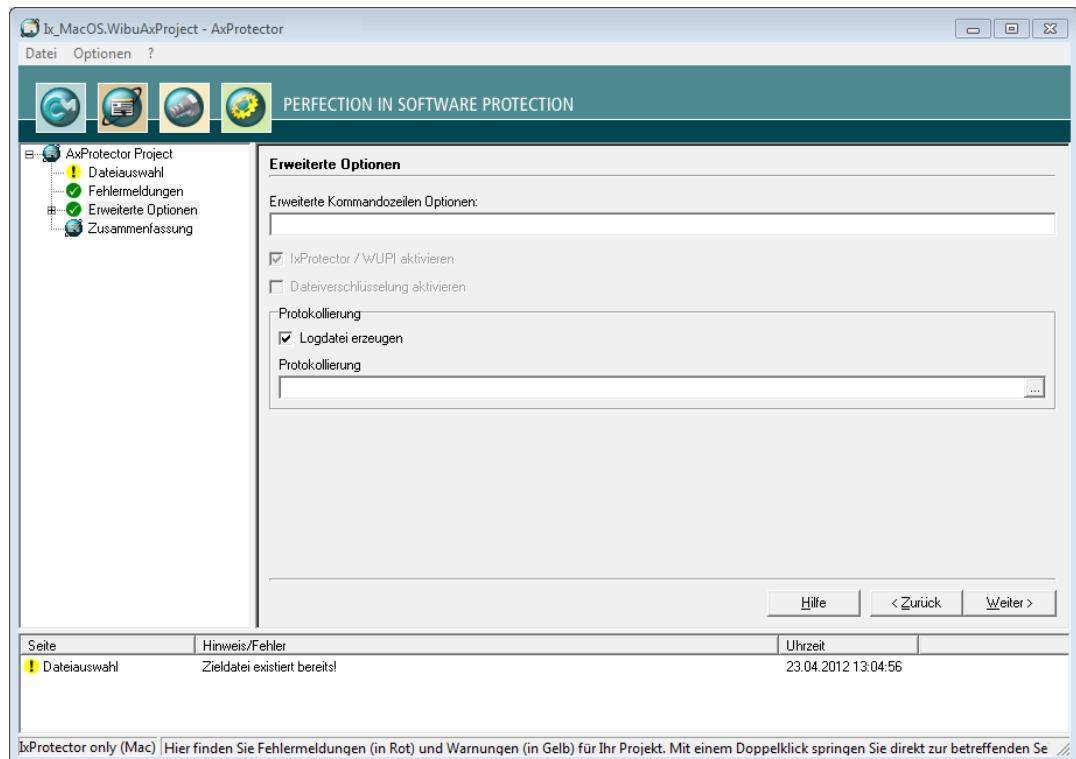


Abbildung 131: IxProtector – Mac OS "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen.  Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.  Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin abgelegt.

7.5.3.3.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *IxProtector* über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

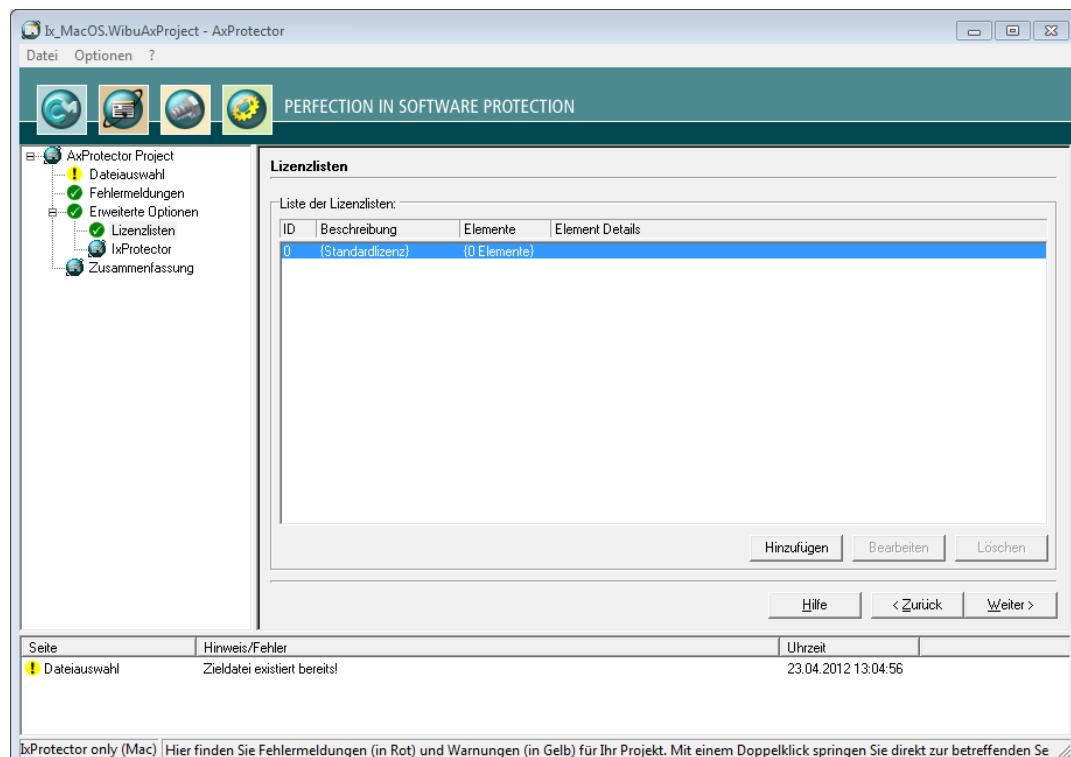
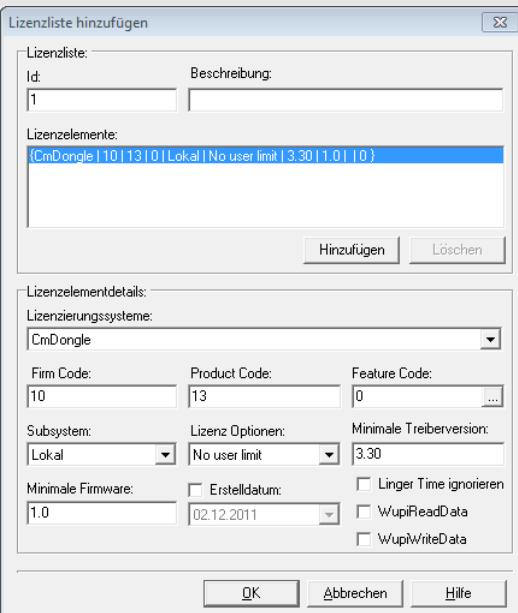


Abbildung 132: *IxProtector* – Mac OS "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.

Element	Beschreibung
	<p>i Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.</p>
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag. 3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
	<p align="center">Abbildung 133: AxProtector - Nur IxProtector "Lizenzlisten hinzufügen"</p>
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (<i>CmDongle</i> , <i>CmActLicense</i> oder <i>WibuKey</i>).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	<p>Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt.</p> <p>Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal).</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenzen:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	<p>Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren.</p> <p>Mit dieser Lizenz Eigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden</p>

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

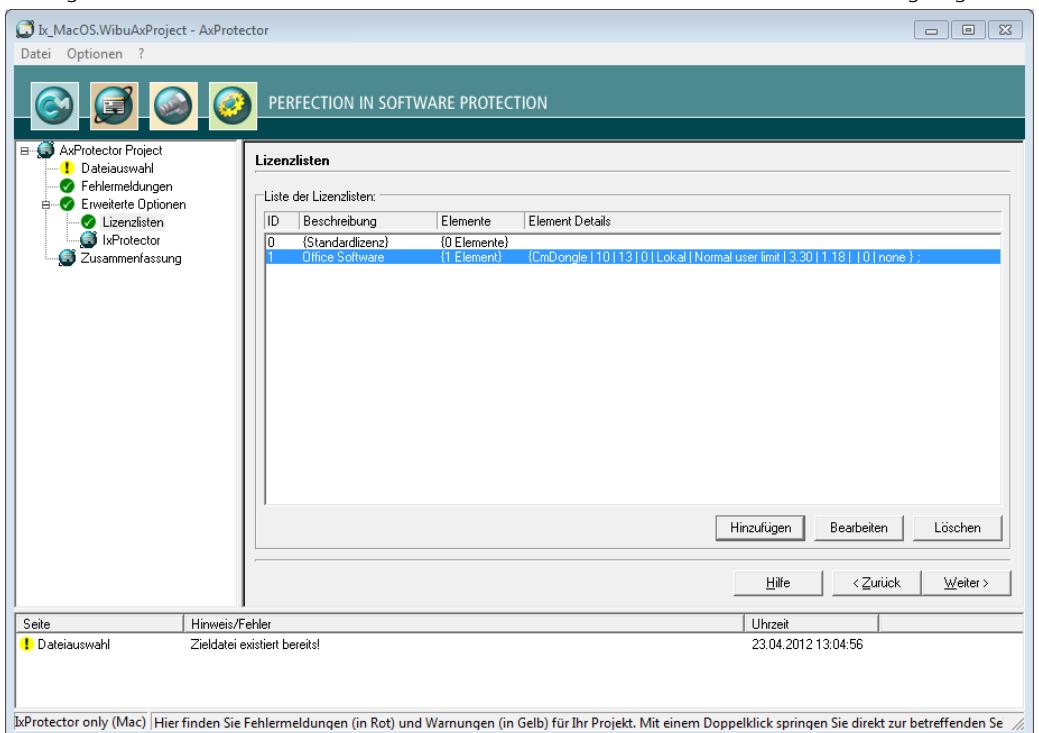


Abbildung 134: AxProtector - Nur *IxProtector* "ausgefüllte Lizenzliste"

7.5.3.3.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

 In diesem Fall werden *CodeMeter®* und *WibuKey API*-Aufrufe über die dynamische Bibliothek (*.d11) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

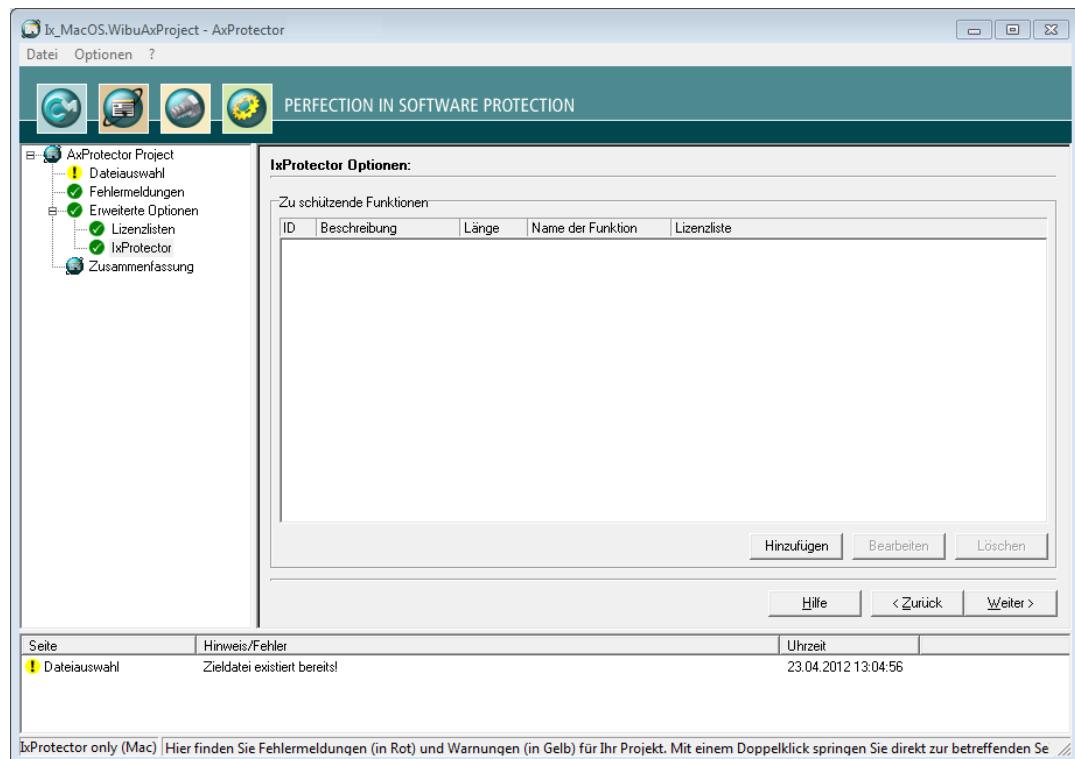
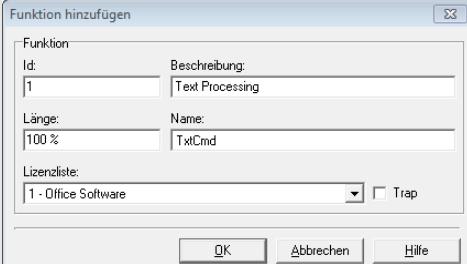


Abbildung 135: *IxProtector* – Mac OS "Funktionsliste"

Element	Beschreibung
Zu schützende Funktionen	<p>Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Betätigen Sie im Bereich <i>IxProtector Optionen</i> die "Hinzufügen" Schaltfläche.

Element	Beschreibung
<p>2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.</p>	
	
	<p>Abbildung 136: AxProtector - Mac OS IxProtector "Funktion hinzufügen"</p>
Element	Beschreibung
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode³²² und WupiEncryptCode³²² verwenden.</p>
Beschreibung	<p>Beschreibt die Funktion durch einen Texteintrag.</p>
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an. Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p>
Lizenzliste	<p>Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.</p>
Trap	<p>Aktiviert die Trap-Funktion für die Funktion. Kommandozeilen-Option siehe hier³⁰⁸.</p>

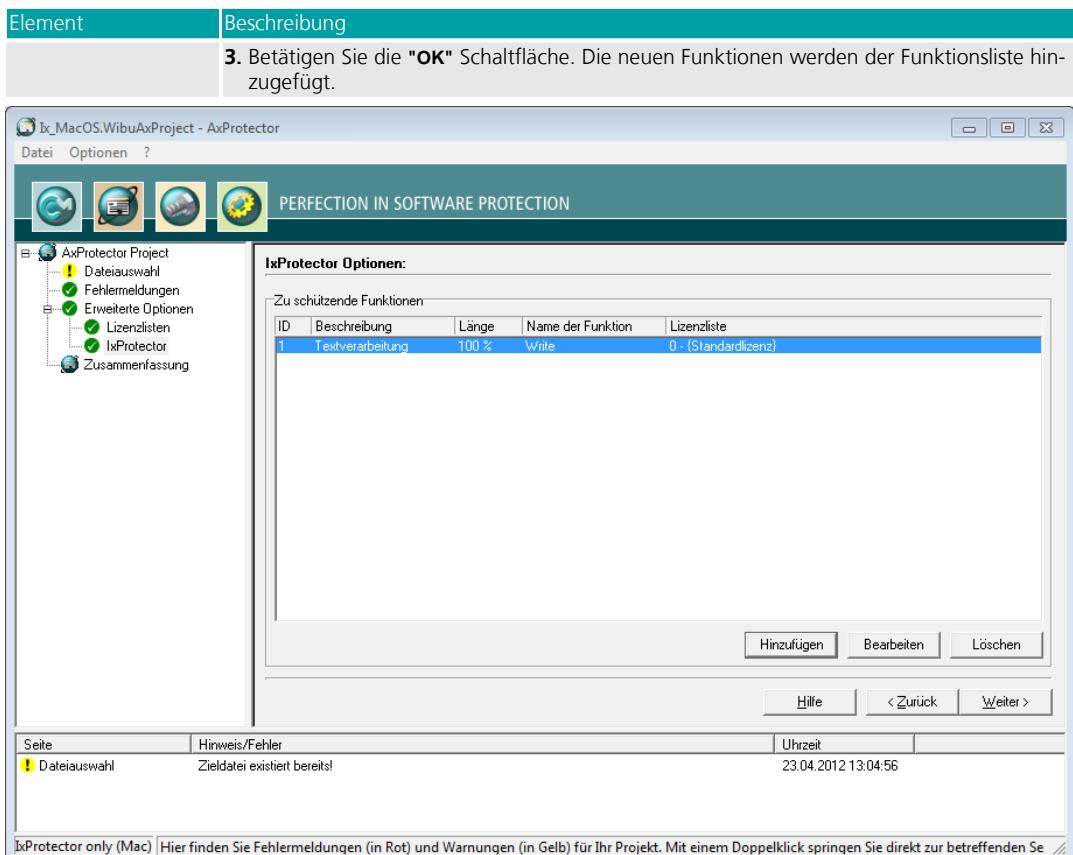


Abbildung 137: AxProtector - Mac OS IxProtector "gefüllte Funktionsliste"

7.5.3.4 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

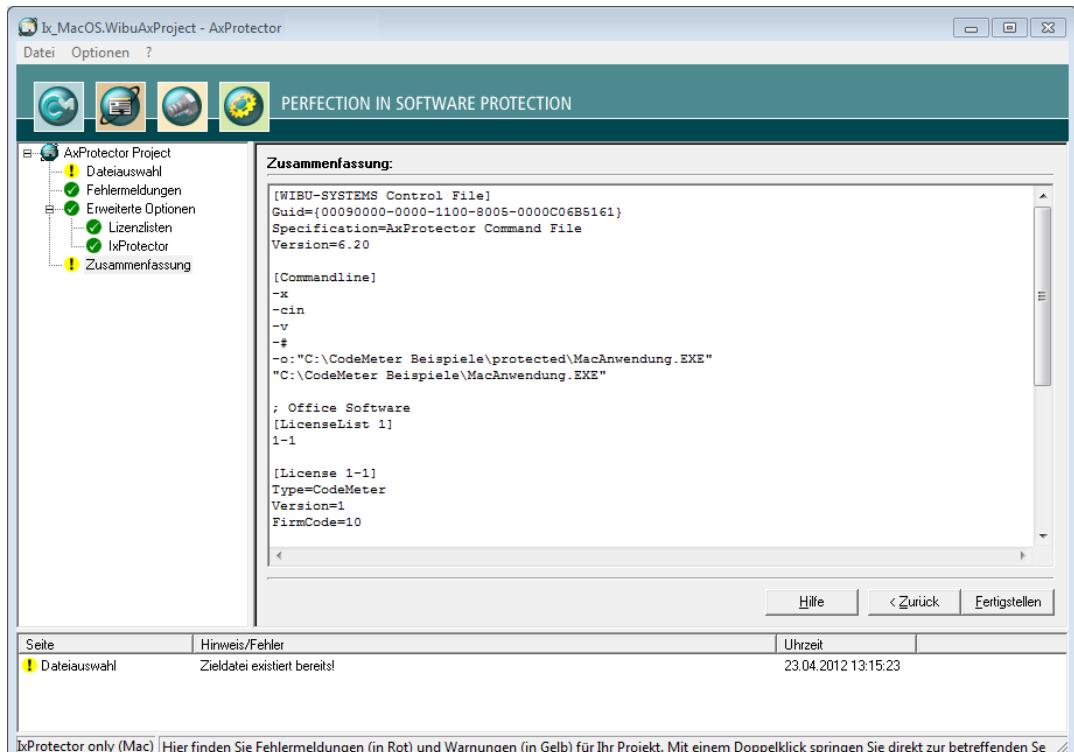


Abbildung 138: IxProtector – Mac OS "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

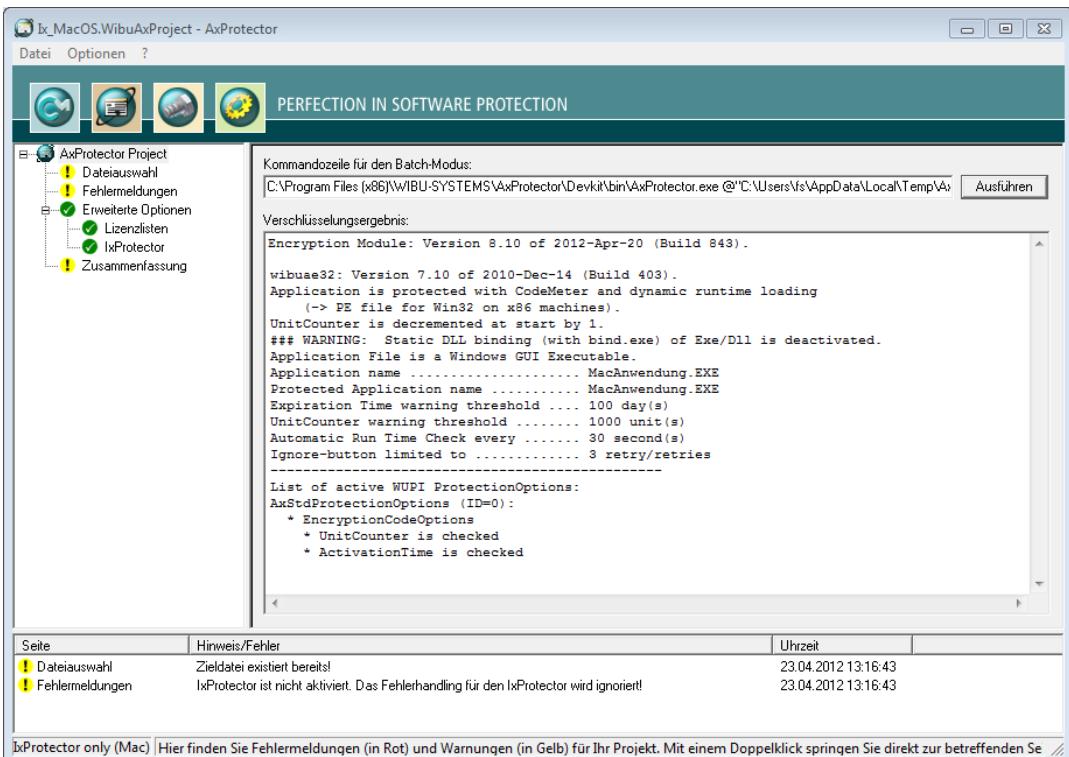


Abbildung 139: IxProtector – Mac OS "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die " Ausführen " Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.  Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen.

7.5.4 Linux Anwendung oder Shared Object

Diesen Projekttyp wählen Sie, wenn Sie eine indexbasierte Verschlüsselung von separaten Funktionen Ihrer Anwendung durchführen wollen, dies aber ohne die gesamte Anwendung noch zusätzlich mit AxProtector zu schützen.



Wibu-Systems empfiehlt IxProtector jedoch innerhalb von AxProtector zu nutzen, falls keine besonderen Gründe dagegen sprechen.

Mit dieser Option sucht IxProtector dann die betreffenden Code-Bereiche heraus und verschlüsselt diese. Aber selbst im Fall, dass Sie den Projekttyp "Nur IxProtector" wählen, ist eine erhöhte Sicherheit des Schutzes gegeben, da die verwendete Dummy-DLL bei der Verwendung von IxProtector durch statischen Code ersetzt wird. Diese DLL wird später bei der Ausführung der Anwendung nicht mehr benötigt.

Die folgende Tabelle fasst zusammen, welche Dateien wie über unterschiedliche Projekttypen und Werkzeuge für Linux mit IxProtector verschlüsselt werden können:

Zu verschlüsselnde Anwendung	Projekttyp	GUI Windows	Kommandozeile
Linux Anwendung oder Shared Object	IxProtector Linux <small>262</small>	✓	Windows Kommandozeile <small>292</small> In einer separaten Kommandozeile für Linux, die auf Linux-Betriebssystemen läuft, können Sie ebenfalls Verschlüsselungsparameter <small>272</small> eingeben.

7.5.4.1 Dateiauswahl

Um betreffende Code-Bereiche einer ausführbaren Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Datei aus, die Sie schützen wollen.

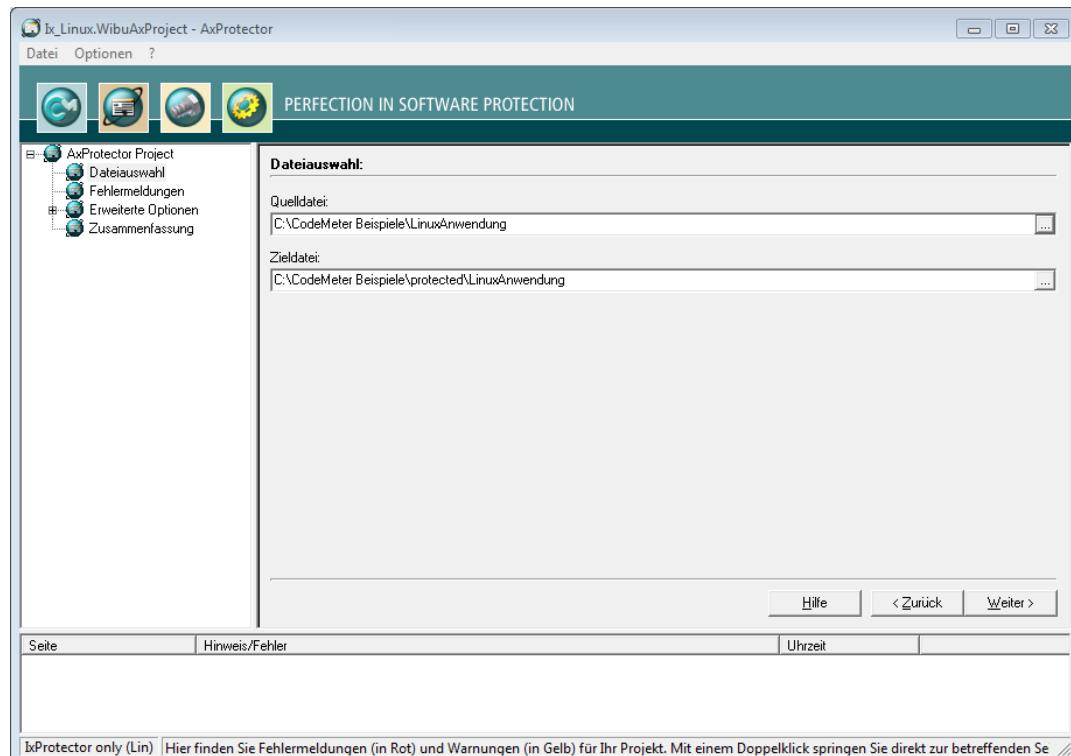


Abbildung 140: *lxProtector – Linux "Dateiauswahl"*

Element	Beschreibung
Quelldatei	Klicken Sie die "..." Schaltfläche und wählen Sie über den " Öffnen " Systemdialog die zu verschlüsseln-de Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> i Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen. </div>
Zieldatei	Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..]\protected\..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung. Kommandozeilen-Option siehe hier ³¹³ .

7.5.4.2 Fehlermeldungen

Über dieses Eingabefenster stellen Sie ein, welche Art von Meldungen im Fehlerfall angezeigt wird. Sie legen fest, ob entweder eine Message DLL mit einer eigenen Fehlerausgabe verwendet wird, oder ob Standard-Hinweisfenster angezeigt werden sollen.

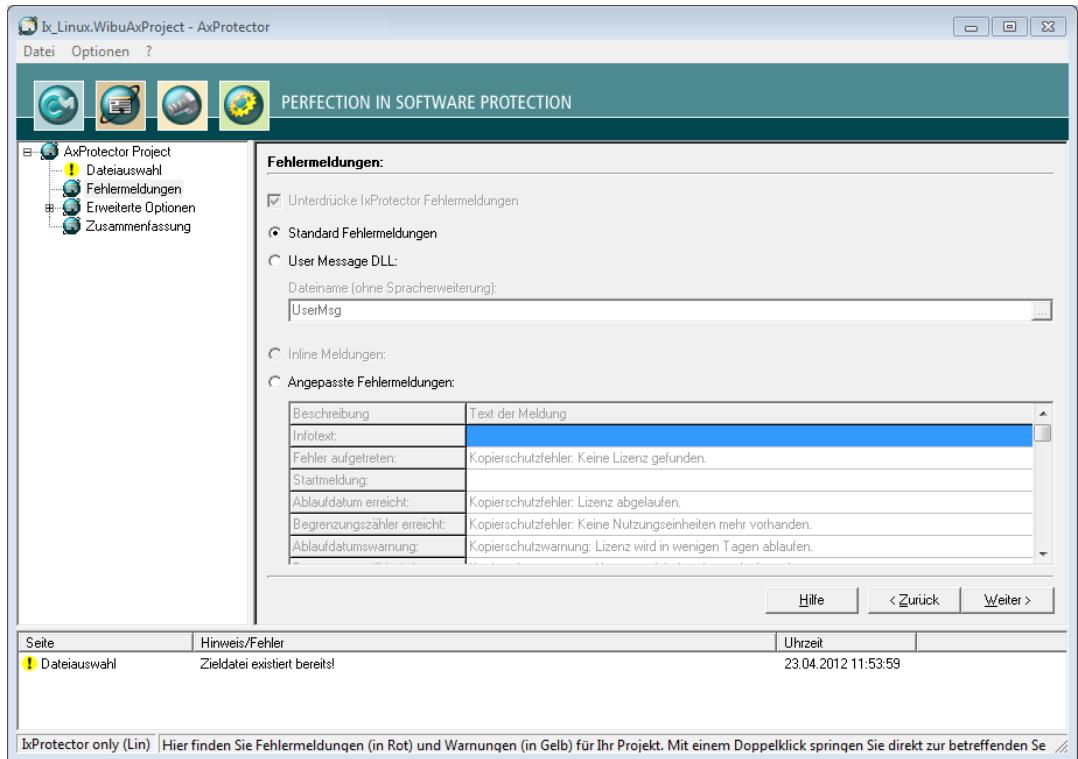
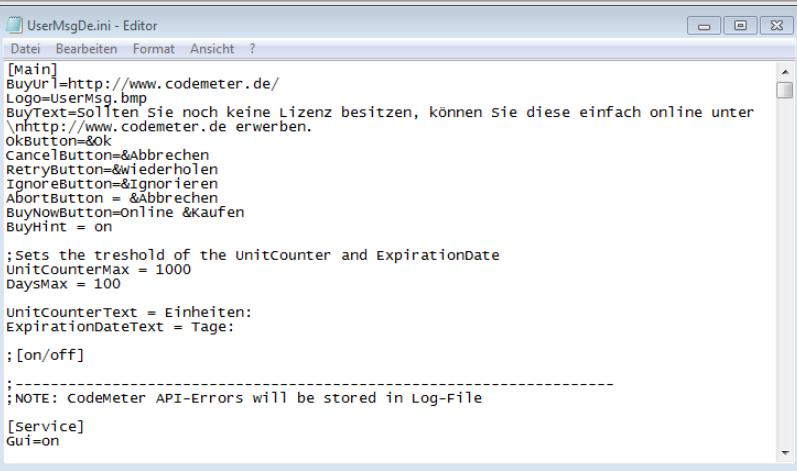


Abbildung 141:IxProtector – Linux "Fehlermeldungen"

Fehlermeldungen

Element	Beschreibung
Unterdrücke IxProtector Fehlermeldungen	Unterdrückt die Ausgabe von IxProtector Fehlermeldungen (Kommandozeilen-Option siehe hier ³⁰⁴). Setzen Sie diese Option nicht, so werden bei der Verwendung von IxProtector im Fehlerfalle zusätzliche Meldungsfenster angezeigt, und zwar zusätzlich zu den im Projekt selbst ausprogrammierten Meldungen.
Standard Fehlermeldungen	Sämtliche Fehlermeldungen, die bei der Ausführung der geschützten Anwendung werden über Standard-Dialoge ausgegeben (Kommandozeilen-Option siehe hier ³¹⁰).
User Message DLL	Aktiviert die Benutzung der User Message DLL. Die Fehlerbeschreibungen können über *.ini-Dateien für verschiedene Sprachen angepasst werden. Sie haben darüber hinaus die Möglichkeit, zur

Element	Beschreibung
	<p>eigenen optischen Gestaltung, z.B. ein eigenes Logo und eigene Texte in die Datei einzupflegen (Kommandozeilen-Option siehe hier³¹²).</p> <p> Die *.ini-Dateien mit dem jeweiligen Länder-Suffix und die DLL-Programmbibliothek werden automatisch in das Verzeichnis abgelegt, in dem sich die AxProtector geschützte Anwendung befindet.</p>  <pre>[Main] Buyurl=http://www.codemeter.de/ Logo=UserMsg.bmp BuyText=Sollten Sie noch keine Lizenz besitzen, können Sie diese einfach online unter \nhttp://www.codemeter.de erwerben. okButton=&OK CancelButton=&Abbrechen Retrybutton=&Wiederholen Ignorebutton=&Ignorieren Abortbutton = &Abbrechen BuyNowButton=Online &kaufen BuyHint = on ;Sets the threshold of the unitCounter and ExpirationDate UnitCounterMax = 1000 DaysMax = 100 UnitCounterText = Einheiten: ExpirationDateText = Tage: ;[on/off] ;-----; ;NOTE: CodeMeter API-Errors will be stored in Log-File [service] Gui=on</pre> <p>Abbildung 142: AxProtector – UserMsgDe.ini</p> <p>Dateiname (ohne Spracherweiterung)</p> <p>Geben Sie hier den Dateinamen ohne Pfadangabe und Datei-Endung an. Die UserMsgDll wird aus dem Verzeichnis %Program Files%\WIIBU-SYSTEMS \AxProtector\DevKit\bin\UserMessage kopiert. Die jeweiligen Initialisierungsdateien sind ebenfalls in diesem Verzeichnis abgelegt.</p>
Inline Meldungen	<p>Linkt für .NET Projekte eine inline assembly und kann ebenfalls über *.ini-Dateien konfiguriert werden (Kommandozeilen-Option siehe hier³¹²).</p> <p> Diese Option ist nur bei der Verschlüsselung von .NET Anwendungen verfügbar.</p>
Anangepasste Fehlermeldungen	Mit dieser Option können Sie eigene Fehlertexte hinterlegen, die in MessageBoxen angezeigt werden.

7.5.4.3 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

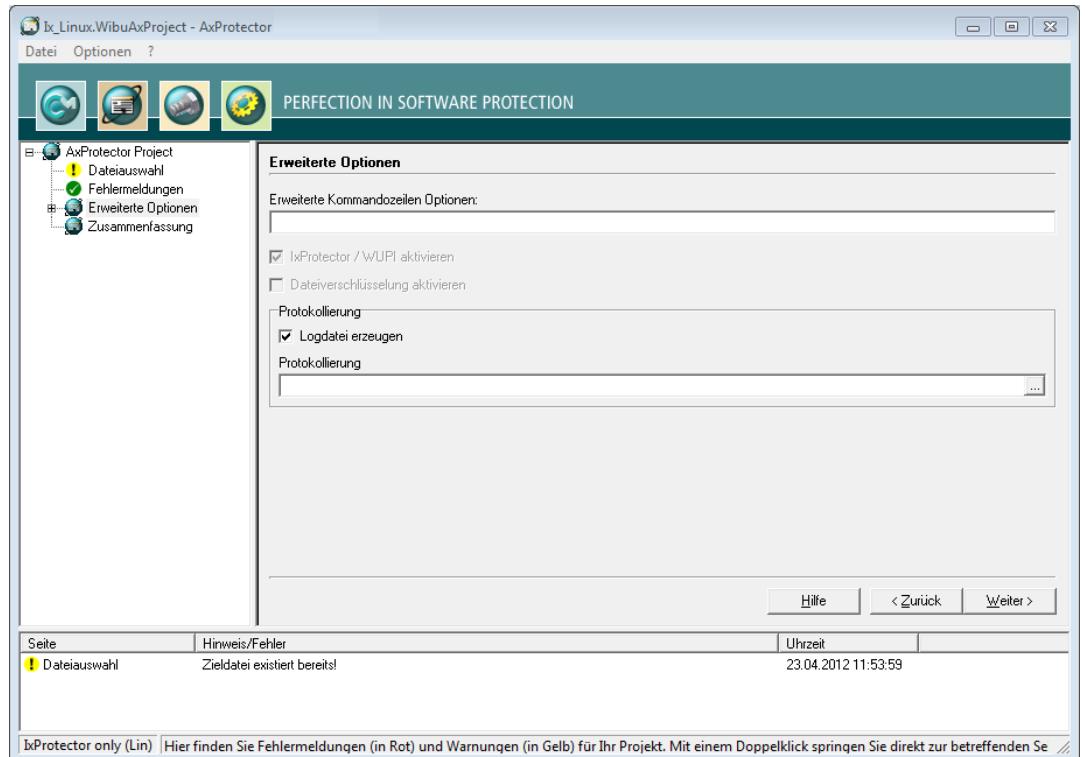


Abbildung 143: IxProtector – Linux "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen.  Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an.  Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin abgelegt.

7.5.4.3.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *IxProtector* über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

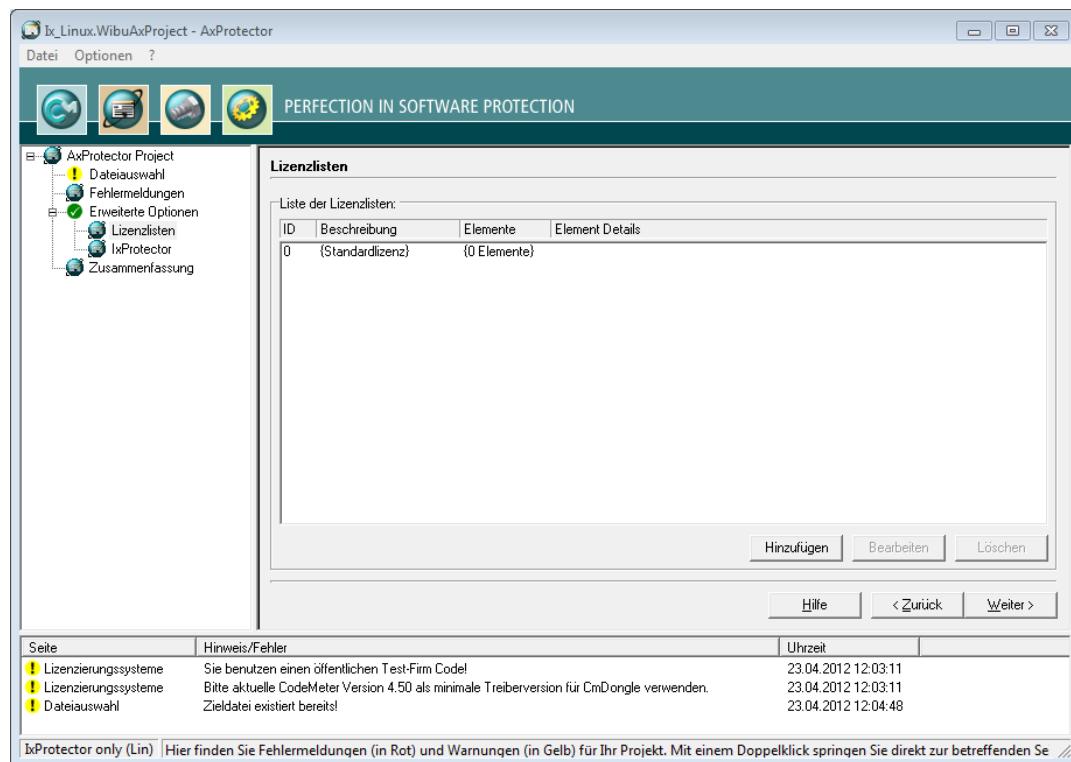
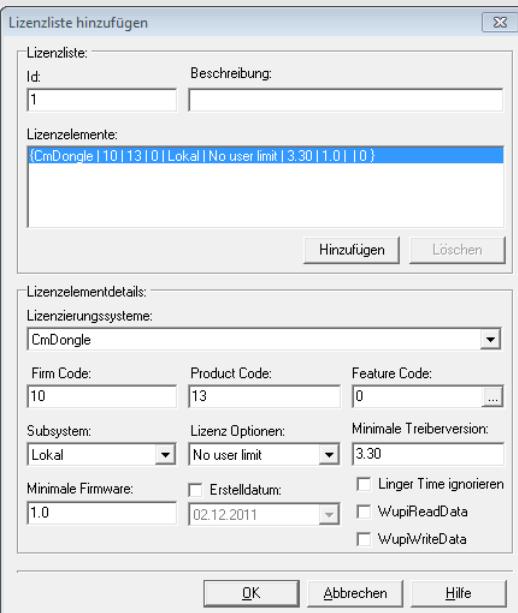


Abbildung 144: *IxProtector – Linux "Lizenzlisten"*

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.

Element	Beschreibung
	<p>i Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.</p>
Beschreibung	<p>Beschreibt die Lizenzliste über einen Texteintrag. 3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p> 
	<p align="center">Abbildung 145: AxProtector - Linux IxProtector "Lizenzlisten hinzufügen"</p>
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (<i>CmDongle</i> , <i>CmActLicense</i> oder <i>WibuKey</i>).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.

Element	Beschreibung
Feature Code	Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt. Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich. 
Subsystem	Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenzen suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal). Lizenz Optionen Auswahl der Lizenz Optionen zur Belegung von Lizenzen: <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversion	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegriffen. Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor: <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenz Eigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden

Element	Beschreibung
	der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer Angaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

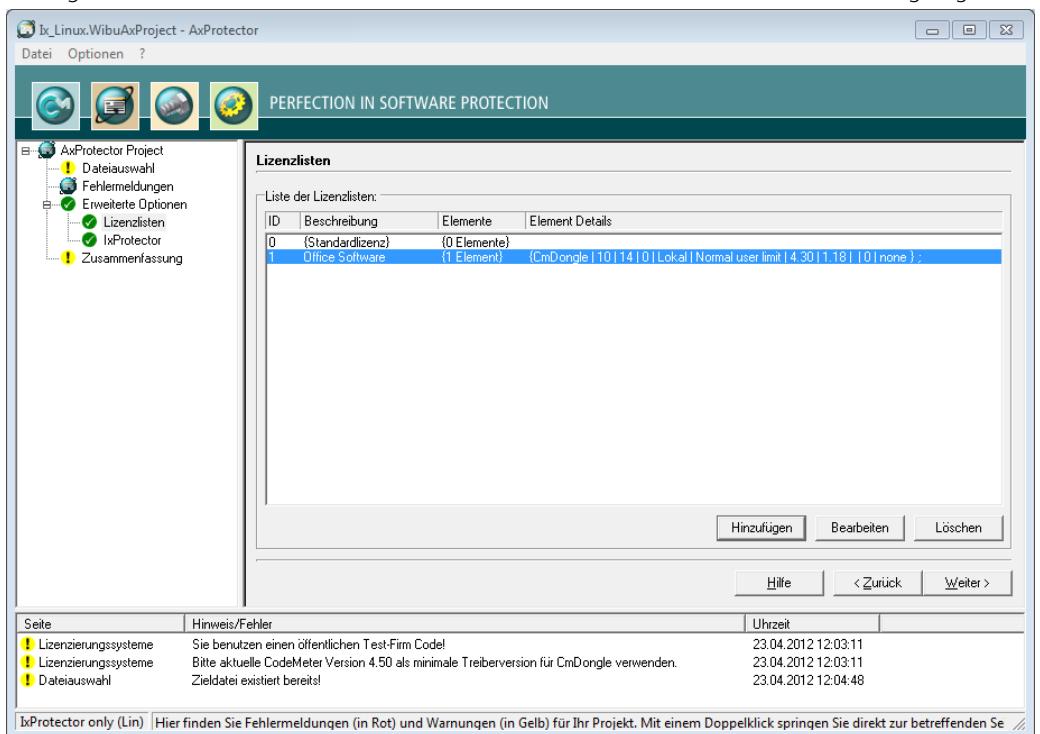


Abbildung 146: AxProtector - Linux *IxProtector* "ausgefüllte Lizenzliste"

7.5.4.3.2 IxProtector

Über diesen Menü-Eintrag definieren Sie einzelne Module (Programmfunktionen), die verschlüsselt werden sollen.

Setzen Sie *IxProtector* ohne Optionen ein, d.h. ohne die explizite Verschlüsselung von Funktionen, erhöht sich die Sicherheit Ihrer Anwendung trotzdem.

 In diesem Fall werden *CodeMeter®* und *WibuKey API*-Aufrufe über die dynamische Bibliothek (*.dll) auf die entsprechenden statischen Bibliotheken umgeleitet und diese an die Anwendung angehängt. Durch den Wegfall der DLL-Schnittstelle erhöht sich die Sicherheit, ohne dass Sie eine Änderung an Ihrer Anwendung vornehmen müssen.

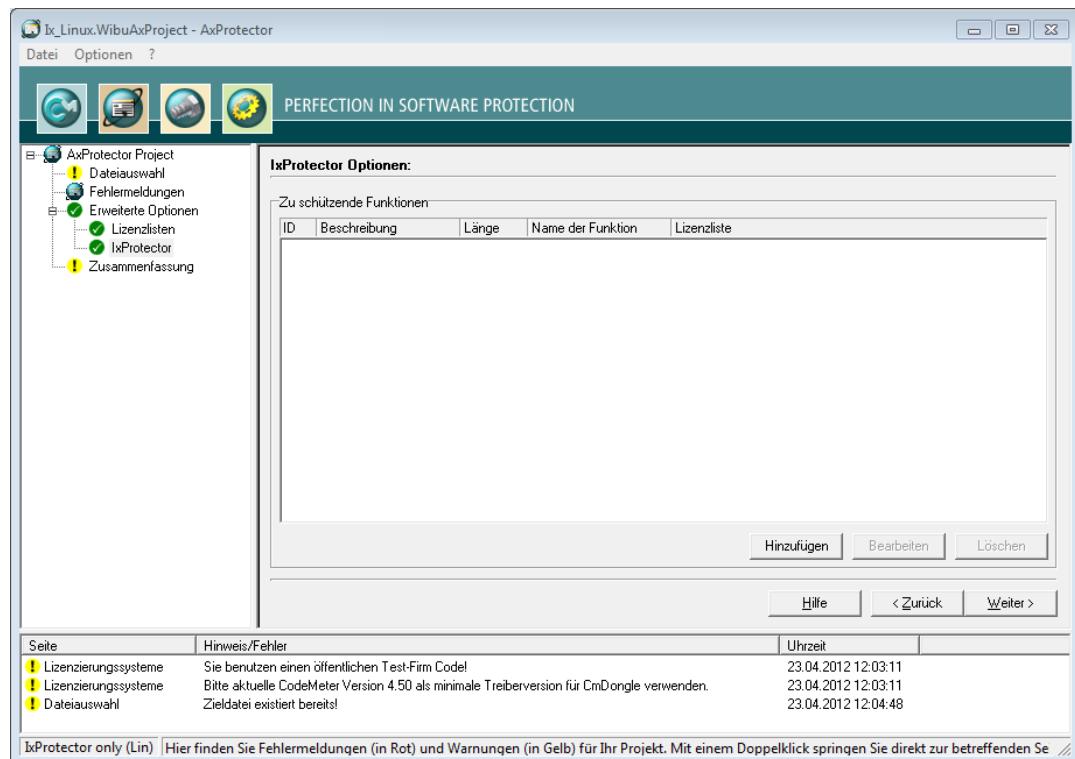
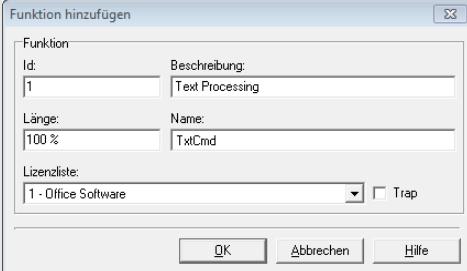


Abbildung 147: *IxProtector – Linux "Funktionsliste"*

Element	Beschreibung
Zu schützende Funktionen	<p>Listet alle angegebenen Funktionslisten inklusive Eigenschaften auf. In diesem Bereich legen Sie auch Funktionslisten an. Dazu gehen Sie wie folgt vor:</p> <p>1. Betätigen Sie im Bereich IxProtector Optionen die "Hinzufügen" Schaltfläche.</p>

Element	Beschreibung
<p>2. Definieren Sie im Bereich Funktion die Funktion durch das Setzen und Ausfüllen der Felder.</p>	
	
	<p>Abbildung 148: AxProtector - Linux IxProtector "Funktion hinzufügen"</p>
Element	Beschreibung
Id	<p>Kennzeichnet die Funktion eindeutig.</p> <p>i Diese Id entspricht der Id, die sie beim Aufrufen der WUPI Befehle WupiDecryptCode³²² und WupiEncryptCode³²² verwenden.</p>
Beschreibung	<p>Beschreibt die Funktion durch einen Texteintrag.</p>
Länge	<p>Gibt die Länge des zu verschlüsselnden Bereichs der Funktion an. Die Länge kann in Prozent (0...100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich. AxProtector ermittelt die Länge dann automatisch.</p> <p>i Geben Sie kein Prozentzeichen hinter der Zahl, so wird die angegebene Zahl als Anzahl Bytes interpretiert.</p>
Name	<p>Eingabe des Namens der zu verschlüsselnden Funktion.</p> <p>Der Funktionsname muss exakt dem in der Exportliste der Linker-Map-Datei entsprechen. Achten Sie daher auf die korrekte Schreibweise (Groß-/Kleinschreibung, Unterstrich, ...).</p> <p>Um den exakten Funktionsnamen aus der ausführbaren Datei zu ermitteln, kann z.B. die Anwendung Microsoft Dependency Walker verwendet werden.</p> <p>i Microsoft Dependency Walker zeigt die Abhängigkeiten zwischen 32- oder 64-Bit Windows-PE-Dateien an. Eine Übersicht über alle verlinkten Module gibt ein Baumdiagramm, die im- und exportierten Funktionen werden jeweils in tabellarischer Form dargestellt. Der Dependency Walker ist Bestandteil der Windows XP SP2 Support Tools und des Microsoft Visual Studio bis Version 8.0 (das Visual Studio 2008, also die Version 9.0, enthält den Dependency Walker nicht mehr).</p>
Lizenzliste	<p>Wählt eine Lizenzliste aus den vorhandenen aus, zu der die Funktion zugeordnet wird. Mit dieser Lizenzliste wird dann die Funktion verschlüsselt.</p>
Trap	<p>Aktiviert die Trap-Funktion für die Funktion. Kommandozeilen-Option siehe hier³⁰⁸.</p>

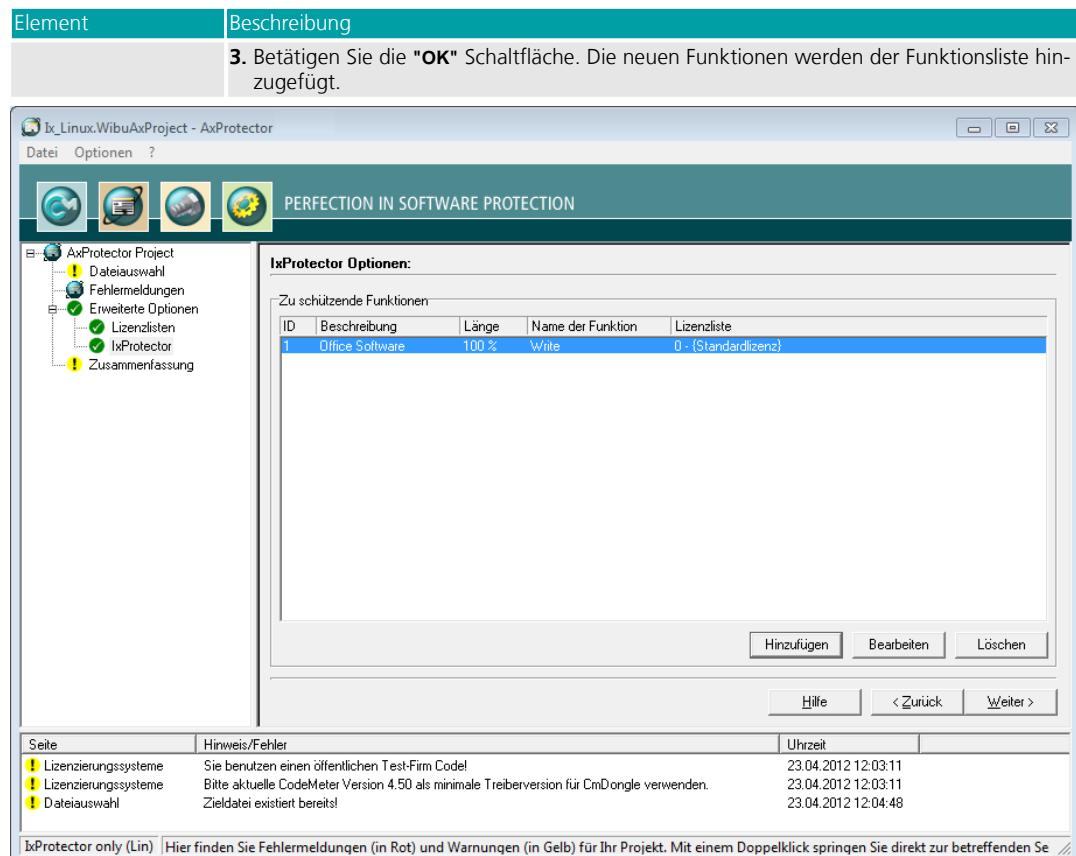


Abbildung 149: IxProtector – Linux "gefüllte Funktionsliste"

7.5.4.4 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.

 Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile `AxProtector.exe @*.wbc auf`

319

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

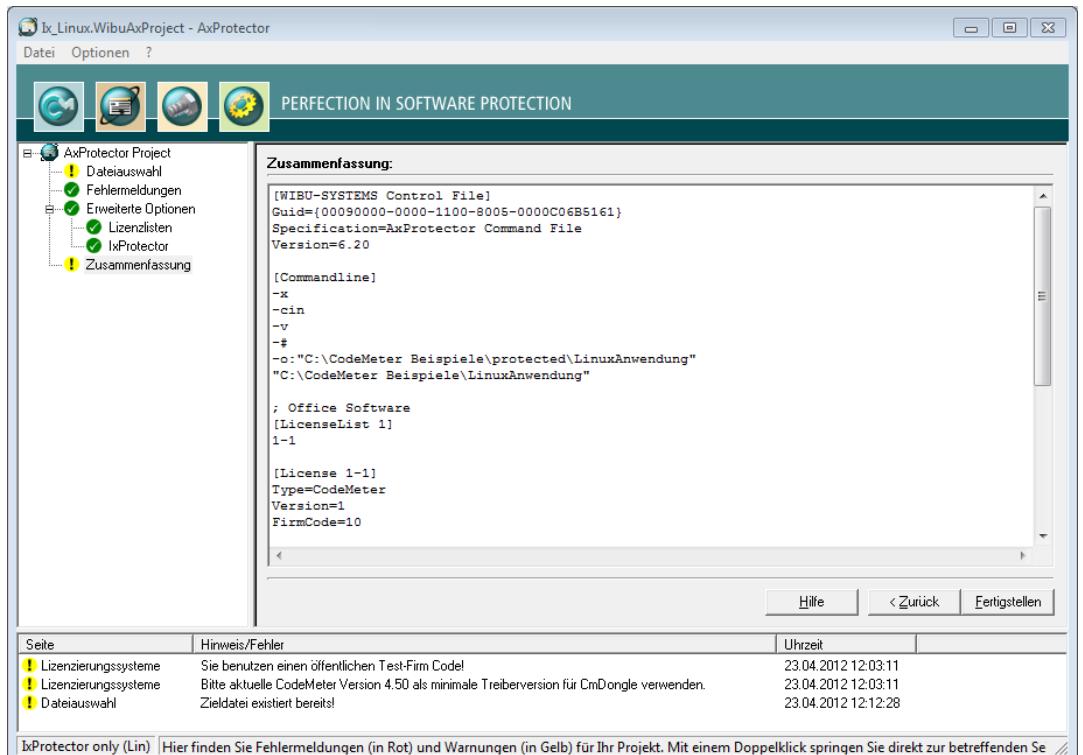


Abbildung 150: IxProtector – Linux "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

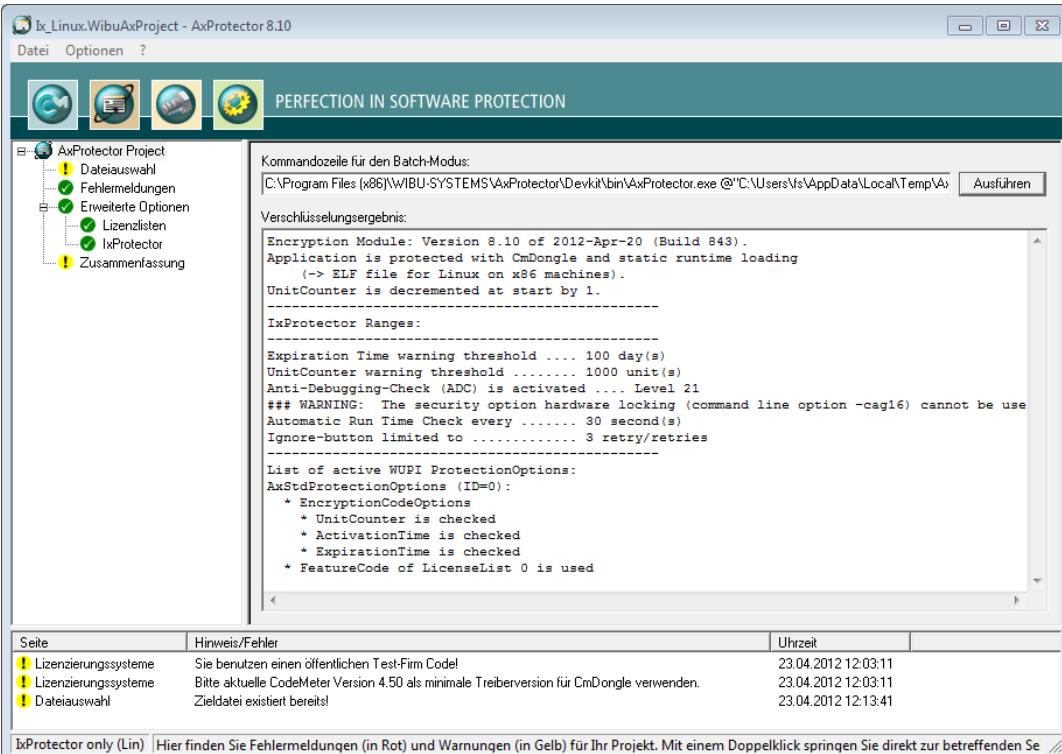


Abbildung 151: AxProtector - Linux IxProtector "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	<p>Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.6 Sonstige Karteireiter

7.6.1 Dateiverschlüsselung

AxProtector erlaubt den automatischen Schutz von Daten-Dateien, die Ihre geschützte Anwendung verwendet. Dieser Schutz durch Verschlüsselung ohne Eingriff in den Quelltext Ihrer Anwendung umfasst beispielsweise:

- Flash-Anwendungen, die aus einer *.exe und vielen *.swf Dateien bestehen
- Datenbankanwendungen (z.B. Visual Fox Pro Anwendungen, die aus einer *.exe und Datenbank-Dateien bestehen)
- Konfigurationsdaten, die in separaten Dateien abgelegt und von Ihrer Software eingelesen werden
- Skripte, die in separaten Dateien liegen und in Ihrer Software abgearbeitet werden
- Daten, die in Ihrer Anwendung erfasst oder visualisiert werden (z.B. Messdaten)
- Dokumente, die der Anwender mit Ihrer Anwendung erstellt.

7.6.1.1 Dateiauswahl

Um eine ausführbare Datei sicher mit AxProtector zu verschlüsseln, wählen Sie zunächst die Daten-Datei aus, die Sie schützen wollen.

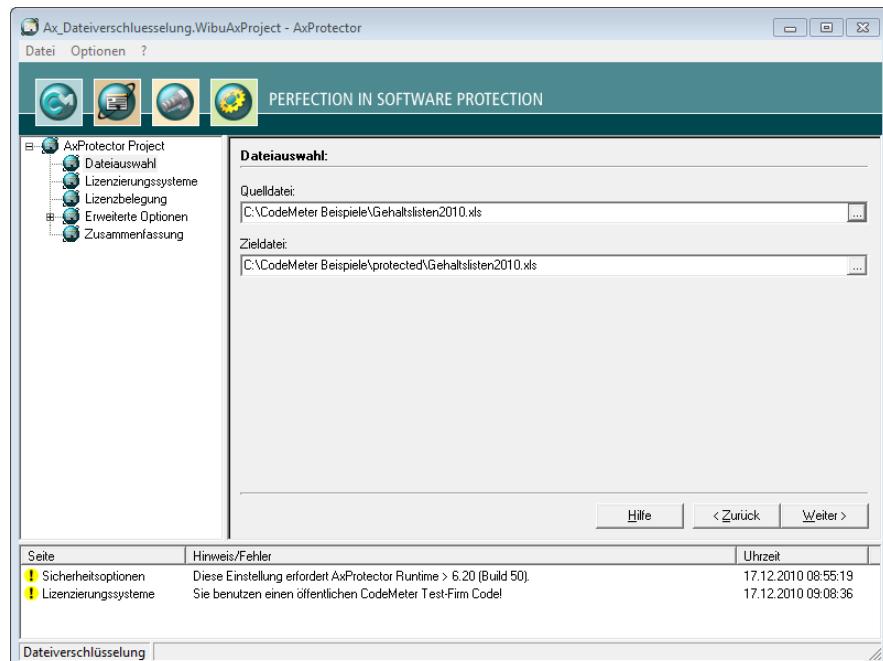


Abbildung 152: AxProtector - Dateiverschlüsselung "Dateiauswahl"

Element	Beschreibung
Quelldatei	<p>Klicken Sie die "..." Schaltfläche und wählen Sie über den "Öffnen" Systemdialog die zu verschlüsseln-de Datei aus. Oder tragen Sie den Pfad und den Dateinamen manuell in das Feld ein.</p> <p> Als Alternative zur "..." Schaltfläche können Sie die Quelldatei auch direkt aus dem Windows-Explorer per Drag&Drop in das Quelldatei-Feld ziehen.</p>
Zieldatei	<p>Nach Auswahl der Quelldatei setzt AxProtector automatisch einen Ziel-Unterordner [..\protected \..]. Sie können diese Vorgabe auch verändern, oder den Pfad und den Namen der Zieldatei manuell eintragen. Die Zieldatei entspricht dann Ihrer geschützten Anwendung.</p> <p>Kommandozeilen-Option siehe hier³¹³.</p>

7.6.1.2 Lizenzierungssysteme

Nach Auswahl der zu schützenden Daten-Datei nehmen Sie hier Einstellungen zum verwendeten Lizenzierungssystem *CodeMeter* vor.

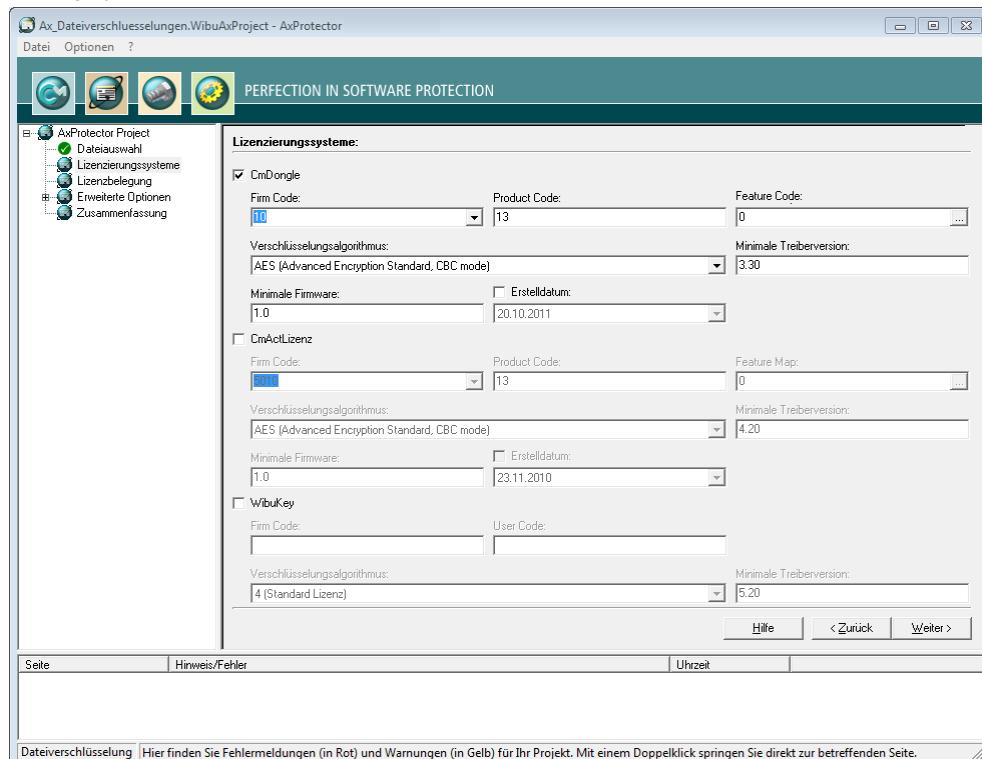


Abbildung 153: AxProtector - Dateiverschlüsselung "Lizenzierungssysteme"

Wenn Sie von *WibuKey* zu *CodeMeter®* umsteigen, aktivieren Sie bitte beide Schutz- und Lizenzierungssysteme.

So können Sie Bestandskunden, die bereits eine *WibuBox* haben ohne Hardwareaustausch mit Updates und Upgrades beliefern. Neukunden erhalten mit der geschützten Anwendung dann zusätzlich einen *CmDongle* oder eine *CmActLicense*.

Außerdem ist hier zusätzlich auch die Verschlüsselung mit dem softwarebasierten Lizenzierungssystem *CmActLicense* möglich. Für mehr Informationen besuchen Sie die *Wibu-Systems* Inter-

netseiten.
Für *CmDongle* und *CmActLicense* sind die folgenden Einstellungen möglich (siehe Kommandozeilen-Option [hier](#)²⁰³):

Element	Beschreibung
Firm Code	<p>Tragen Sie den Firm Code ein, der für die Verschlüsselung der Software verwendet wird.</p> <p> Der Firm Code 10 in der obigen Abbildung ist der <i>CmDongle Evaluation-Firm Code</i> des <i>CodeMeter® Software Development Kits (SDK)</i> und wird für den späteren Schutz Ihrer eigenen Software nicht verwendet. Der Test Firm Code für <i>CmActLicense</i> ist 5010. Als Lizenzgeber tragen Sie an dieser Stelle später Ihre(n) eigenen Firm Code(s) ein.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Product Code	<p>Tragen Sie den Product Code ein, der die Verschlüsselung eines bestimmten Produkts festlegt. Diese Kennung können Sie frei wählen, z.B. für ein separates Modul einer Software-Anwendung, oder eine einzelne Anwendung.</p> <p>Kommandozeilen-Option siehe hier²⁹³.</p>
Feature Code	<p>Tragen Sie einen Feature Code ein, der z.B. die Verschlüsselung verschiedener Versionen bewirkt.</p> <p> Standardmäßig ist hier ein Feature Code von 0 gesetzt. Dadurch ist die Verwendung der Product Item Option Feature Map deaktiviert. Abweichend können Sie hier einen 32-Bit Wert eintragen.</p> <p>Über die "..." Schaltfläche ist die Eingabe dieses Wertes als Hexadezimalzahl, Dezimalzahl und binär möglich.</p> 
Verschlüsselungs-Algorithmus	<p>Wählen Sie den Algorithmus zur Verschlüsselung Ihrer Software aus. <i>CodeMeter®</i> unterstützt derzeit nur AES (Advanced Encryption Standard).</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Minimale Treiberversion	<p>Geben Sie die minimal benötigte Treiberversion des installierten <i>CodeMeter Lizenzservers</i> an.</p> <p>Ein automatisches Session-Handling auf Terminalservern erreichen Sie durch Setzen der minimalen Treiberversion auf die Version 3.20. Damit übernimmt <i>AxProtector</i> automatisch das Session-Handling, wenn die geschützte Anwendung auf Terminalservern läuft, und jede einzelne Sitzung belegt eine der verfügbaren Lizzen.</p> <p> Das Setzen der Treiberversion ist ebenso notwendig, wenn z.B. einige neue Features beim Schutz der Anwendung zum Einsatz kommen. Möglicherweise unterstützt eine ältere Treiberversion diese Features dann nicht und reagiert mit Fehlermeldungen</p>

Element	Beschreibung
	<p>beim Starten Ihrer geschützten Software. Kommandozeilen-Option siehe hier²⁹⁴.</p>
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum⁴⁷ (Maintenance Period)⁴⁷. In der PIO werden zwei Datumswerte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden.</p> <p>Eine Lizenz berechtigt dann nur zur Verwendung der Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft, ob das Erstelldatum (Release Date) innerhalb dieses Zeitraumes liegt. Liegt das Erstelldatum (Release Date) außerhalb des Wartungszeitraums (Maintenance Period), so ist die Verwendung nicht durch die Lizenz abgedeckt.</p> <p>Zum Hinterlegen des Erstelldatums gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe des Erstelldatums. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet. <p>Nach der Aktivierung des Kontrollkästchens ändert sich automatisch der Inhalt des "Minimum Firmware"-Feldes auf die Version 1.18, die mindestens benötigt wird, um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>
Minimale Firmware	<p>Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.</p> <p>Kommandozeilen-Option siehe hier²⁹⁴.</p>

WibuKey

Über Einstellungen für das hardware-basierte Lizenzierungssystem *WibuKey* informiert separat das *WibuKey* Entwicklerhandbuch.

7.6.1.3 Lizenzbelegung

Über dieses Eingabefenster legen Sie fest, ob die geschützte Daten-Datei vorhandene Lizenzen im CmContainer lokal, im Netzwerk oder beides suchen soll, und wie die Lizenzen belegt werden sollen.

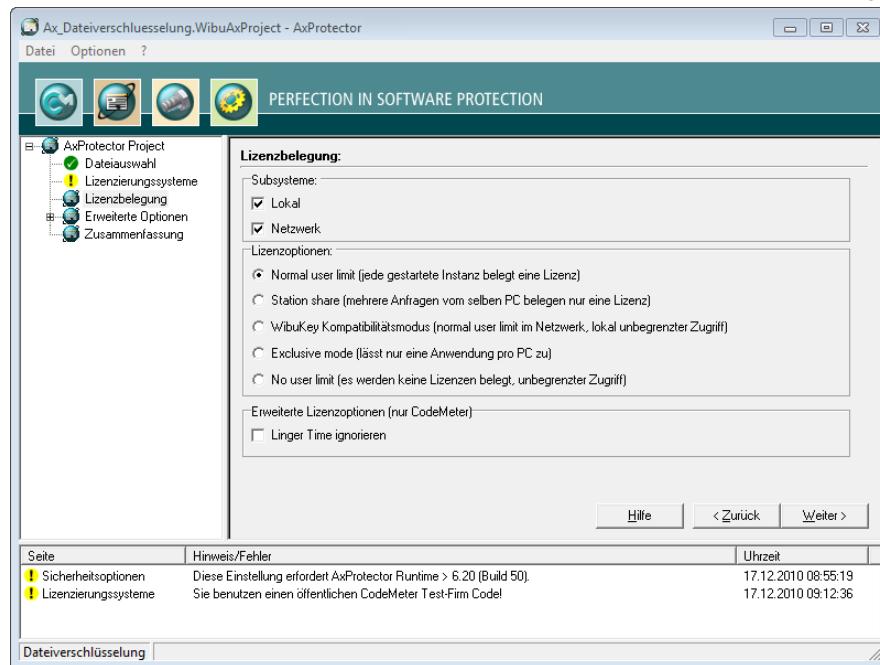


Abbildung 155: AxProtector - Dateiverschlüsselung "Lizenzbelegung"

Subsysteme

Hier legen Sie fest in welchem Subsystem (lokal oder im Netzwerk) die geschützte Anwendung die passende(n) Lizenz(en) suchen soll (Kommandozeilen-Option siehe [hier](#)²⁹⁴).

Element	Beschreibung
Lokal	Diese Einstellung definiert, dass die geschützte Anwendung ausschließlich nach Lizenzen sucht, die sich auf demselben PC befinden bzw. derselben virtuellen Machine (VM) zugeordnet sind.
Netzwerk	Diese Einstellung definiert, dass die Lizenz für die geschützte Anwendung im Netz gesucht werden soll, d.h. es wird nur auf Computer zugegriffen, auf dem der CodeMeter Lizenzserver mit einem aktvierten Netzwerkzugriff läuft. Bei gleichzeitiger Auswahl beider Subsysteme wird die Lizenz zunächst lokal und danach im Netzwerk gesucht.

Lizenzoptionen

Im Bereich Lizenzoptionen legen Sie fest, wie sich gestartete Instanzen der geschützten Anwendung und die Belegung von Lizenzen zueinander verhalten sollen (Kommandozeilen-Option siehe [hier](#)²⁹⁵).

Element	Beschreibung
Normal user limit	Hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> lokal an einem PC, oder in einem Netzwerk gefunden wurde.
Station Share	Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz.  Diese Option setzen Sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.
WibuKey Kompatibilitäts-Modus	Hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).  Diese Belegungsoption besteht lediglich aus Kompatibilitätsgründen zu <i>WibuKey</i> . Wibu-Systems empfiehlt die Einstellungen 'Normal user limit' und 'Station Share'
Exclusive Mode	Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden.
No user limit	Hier können beliebig viele Instanzen der geschützten Anwendung lokal oder im Netzwerk gestartet werden wobei keine zusätzlichen Lizizen belegt werden. Belegte Lizizen können in diesem Modus nochmal verwendet werden.

Linger Time

Element	Beschreibung
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte Linger Time zu ignorieren. Mit dieser Lizenzenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im <i>CodeMeter Entwicklerhandbuch</i>).

7.6.1.4 Erweiterte Optionen

Über dieses Eingabefenster haben Sie die Möglichkeit weitere Einstellungen für die Verschlüsselung vorzunehmen.

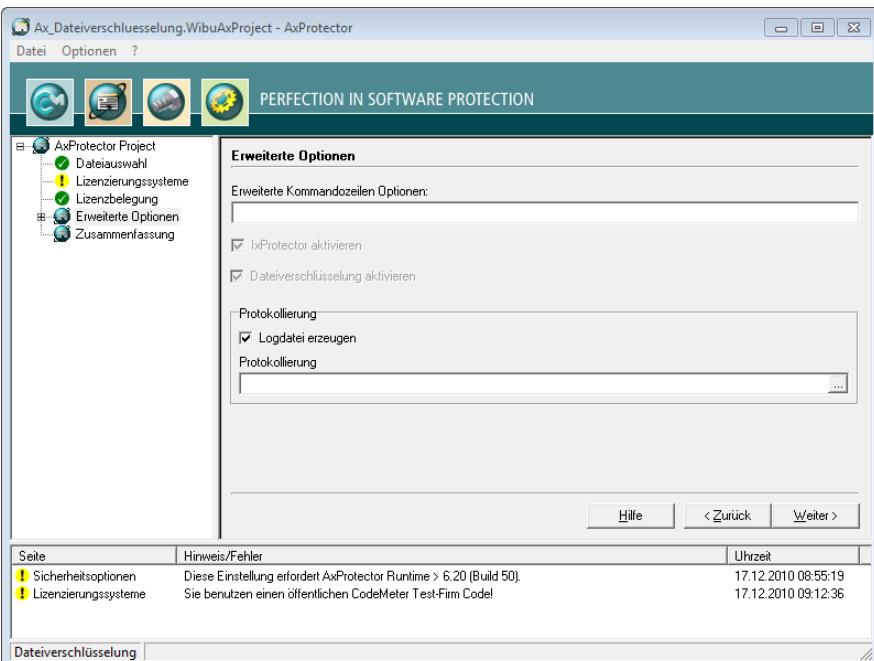


Abbildung 156: AxProtector - Dateiverschlüsselung "Erweiterte Optionen"

Element	Beschreibung
Erweiterte Kommandozeilen-Optionen	An dieser Stelle können Sie erweiterte Optionen oder neue Feature-Funktionen direkt in Form von Kommandozeilen-Parametern setzen. Für weitere Information setzen Sie sich bitte mit dem Support in Verbindung.
Logdatei erzeugen	Das Aktivieren des Auswahlkästchens legt zum Zwecke der Protokollierung eine Ausgabedatei an.
Protokollierung	Geben Sie hier den Pfad und den Dateiname dieser Protokolldatei an. Geben Sie nur den Namen der Datei ohne Verzeichnisnamen an, so wird sie standardmäßig in das Verzeichnis %\Program Files%\WIBU-SYSTEMS \AxProtector\DevKit\bin abgelegt.

7.6.1.4.1 Lizenzlisten

Über diesen Menü-Eintrag verwalten Sie Lizenzlisten, die Sie beim modularen Schutz Ihrer Anwendung mit *lxDestructor* über das [Softwareschutz-API \(WUPI\)](#)³²⁰ verwenden.

Lizenzlisten bestehen aus einer eindeutigen Kennung (**ID**), enthalten eine **Beschreibung** sowie Angaben über **Elemente** und **Element Details**.

 Diese **ID** entspricht der Kennung, die Sie beim Aufrufen der meisten [WUPI-Befehle](#)³²² zur Identifizierung der Lizenz benötigen.

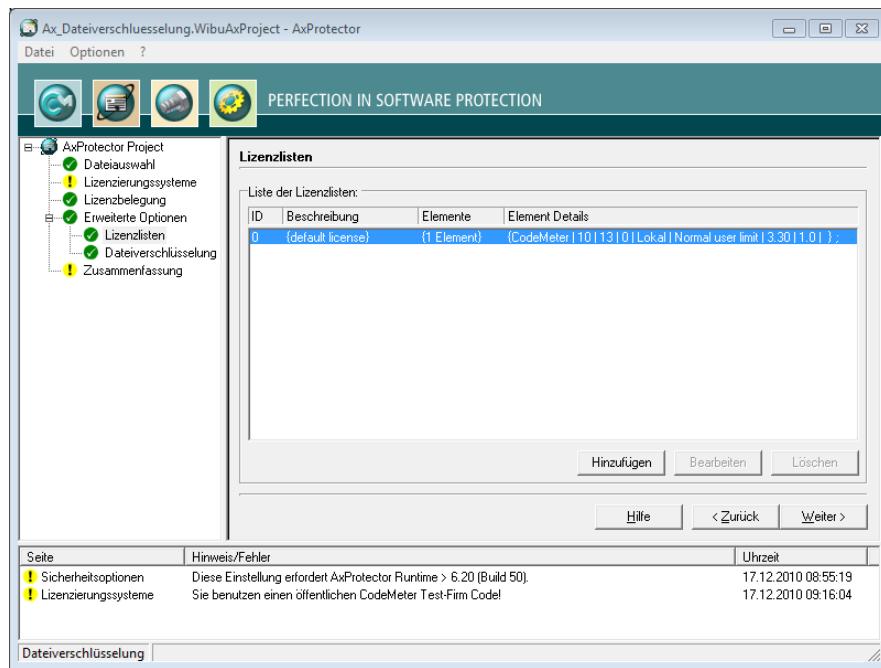
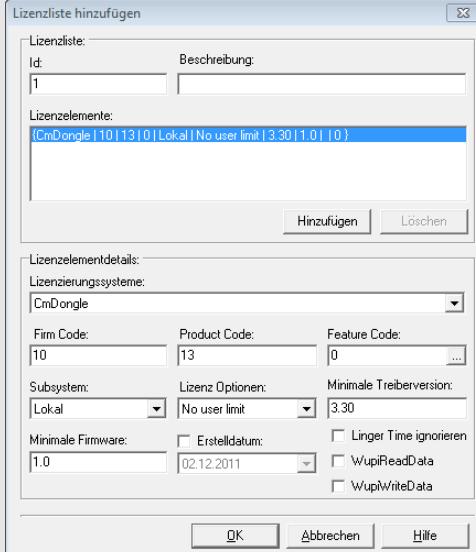


Abbildung 157: AxProtector - Dateiverschlüsselung "Lizenzlisten"

Über diesen Menü-Eintrag legen Sie ebenfalls Lizenzlisten an. Dazu gehen Sie wie folgt vor:

1. Betätigen Sie die "**Hinzufügen**" Schaltfläche.
2. Vergeben Sie im Bereich **Lizenzliste** eine **Id** und füllen das Feld **Beschreibung** aus.

Element	Beschreibung
Id	Kennzeichnet die Lizenzliste eindeutig und dient zur Referenzierung.  Die ID 0 ist durch Ihre Auswahl des Lizenzierungssystems am Anfang als Standard gesetzt. Sie können hier im Folgenden Lizenzlisteneinträge mit IDs ab 1 angelegen.
Beschreibung	Beschreibt die Lizenzliste über einen Texteintrag.

Element	Beschreibung
	<p>3. Definieren Sie im Bereich Lizenzelementdetails die Lizenz durch das Setzen und Ausfüllen der Felder.</p>  <p>The dialog shows a license list with one entry (Id: 1, Beschreibung: [CmDongle 10 13 10 Lokal No user limit 3.30 1.0 0]) and a detailed view of its settings:</p> <ul style="list-style-type: none"> Lizenzelementdetails: <ul style="list-style-type: none"> Lizenzierungssysteme: CmDongle Firm Code: 10, Product Code: 13, Feature Code: 0 Subsystem: Lokal, Lizenz Optionen: No user limit, Minimale Treiberversion: 3.30 Minimale Firmware: 1.0, Erstelltdatum: 02.12.2011, WupiReadData, WupiWriteData checkboxes
	Abbildung 158: AxProtector - Dateiverschlüsselung "Lizenzlisten hinzufügen"
Lizenzierungs-Systeme	Auswählen des Lizenzierungssystems, das zum Schutz der Lizenz verwendet wird (CmDongle, CmActLicense oder WibuKey).
Firm Code	Eingabe des Firm Code, der zum Schutz der Lizenz verwendet wird.
Product Code	Eingabe des Product Code, der zum Schutz der Lizenz verwendet wird.
Feature Code	Eingabe des Feature Code, der z.B. die Verschlüsselung verschiedener Versionen Ihrer Anwendung bewirkt. Über die "..." Schaltfläche ist die Eingabe als Hexadezimalzahl, Dezimalzahl und binär möglich.
	 <p>The dialog allows conversion between Hex, Dec, and Bin formats. The current input is 00000000.</p>

Element	Beschreibung
Subsystem	<p>Auswahl des Subsystems, in dem die geschützte Anwendung nach Lizenz suchen soll (nur lokal oder nur im Netzwerk) bzw. die Suchreihenfolge (erst lokal, dann im Netzwerk, oder erst im Netzwerk, dann lokal) .</p> <p>Lizenz Optionen</p> <p>Auswahl der Lizenz Optionen zur Belegung von Lizenz:</p> <ul style="list-style-type: none"> • Normal user limit • Station share • WK Kompatibilitätsmodus • Exclusive mode • No User limit
Minimale Treiberversi-on	Angabe der erforderlichen minimalen Treiberversion zur Ausführung der geschützten Anwendung.
Erstelldatum	<p>Ab der Firmware-Version 1.18 unterstützt CodeMeter® die Product Item Option Wartungszeitraum (Maintenance Period). In der PIO werden zwei Datums-werte abgespeichert: ein Anfangs- und ein Endwert. Damit lassen sich Lizenzmodelle umsetzen, die Support- und Wartungsleistungen bei der Nutzung der Software abbilden. Eine Lizenz berechtigt dann nur zum Bezug aller neuen Software-Versionen, Korrekturen und Erweiterungen, die innerhalb dieses Zeitraumes erstellt wurden. Dazu wird das Erstelldatum (Release Date) in der geschützten Anwendung hinterlegt und bei der Verwendung geprüft. Ist dieser Zeitraum abgelaufen, ist die Software zwar weiterhin verwendbar, aber der Bezug neuer Versionen, etc. nicht mehr inbegri-fen.</p> <p>Zum Hinterlegen des Erstelldatums (Release Date) gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Aktivieren Sie das "Erstelldatum"-Kontrollkästchen zur Eingabe. Das aktuelle Datum ist voreingestellt. 2. Ändern Sie, wenn gewünscht, das Datum entweder direkt im untenstehende Feld ab, oder verwenden den Kalender, der sich über die Pfeiltaste an linken Rand des Feldes öffnet.
Minimale Firmware	Geben Sie die minimal benötigte Firmware-Version an. Um die Product Item Option Wartungszeitraum (Maintenance Period) nutzen zu können, benötigen Sie die Firmware-Version 1.18.
Linger Time ignorieren	Aktivieren Sie diese Option um eine programmierte LingerTime zu ignorieren. Mit dieser Lizenz-eigenschaft kann eine Belegungszeit der Lizenz nach Freigabe oder Beenden der geschützten Anwendung angegeben werden (mehr Informationen im CodeMeter Entwicklerhandbuch).
WupiReadData	Das Aktivieren des Auswahlkästchens liest Daten ³²⁴ aus dem <i>CmContainer</i> , wenn diese Da-ten vorher an einer festgelegten Stelle gespeichert wurden.
WupiWriteData	Das Aktivieren des Auswahlkästchens schreibt Daten ³²⁵ in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde.

Nachdem Sie die alle gewünschten Einstellungen im Bereich Lizenzelementdetails definiert haben, fahren sie wie folgt fort:

4. Betätigen Sie im Bereich Lizenzliste die "**Hinzufügen**" Schaltfläche. Die Zusammenfassung Ihrer An-gaben entnehmen Sie der Auflistung der Lizenzelemente.
5. Betätigen Sie die "**OK**" Schaltfläche. Die neuen Lizenzdaten werden der Lizenzliste hinzugefügt.

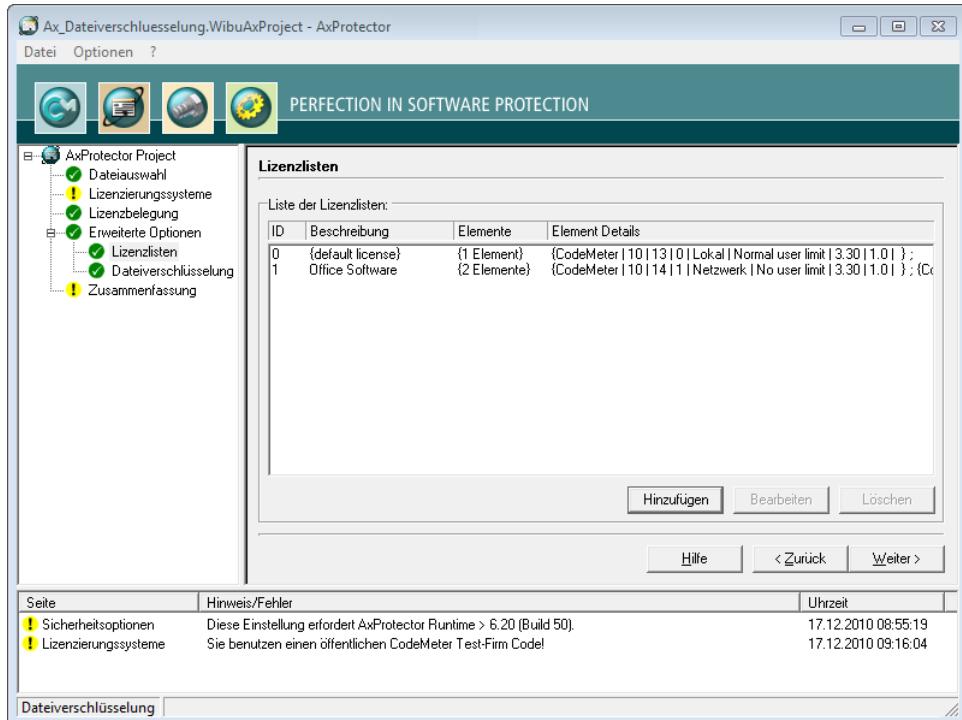


Abbildung 159: AxProtector - Dateiverschlüsselung "ausgefüllte Lizenzliste"

7.6.1.4.2 Dateiverschlüsselung

Über diesen Menü-Eintrag legen Sie fest, nach welchen Regeln eine Anwendung auf verschlüsselte Dateien zugreift. Außerdem haben Sie Möglichkeit dies für unterschiedliche Dateitypen in einer Liste zu definieren. Es können beliebig viele Dateitypen hinzugefügt werden. Für eine Daten-Datei wird lediglich ein Dateityp benötigt.

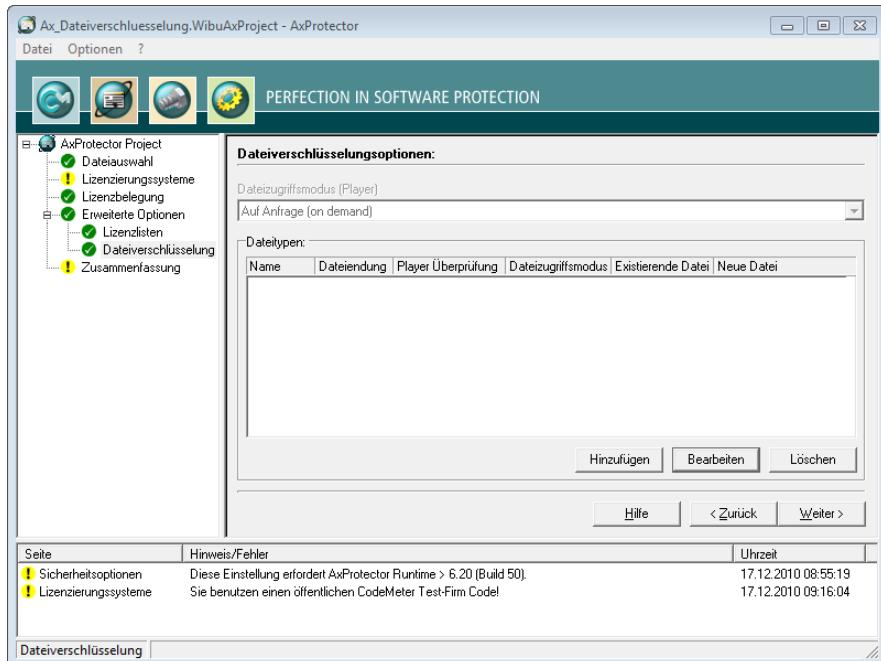


Abbildung 160: AxProtector - Dateiverschlüsselung "Dateiverschlüsselung"

Element	Beschreibung
Dateityp hinzufügen	<p>1. Klicken Sie auf die "Hinzufügen" Schaltfläche, um der Liste einen neuen Dateityp hinzuzufügen.</p> <p>Dateityp hinzufügen</p> <p>Name: _____ Dateiendung: _____</p> <p>Player Überprüfung: [0 - (default license)] Dateizugriffsmodus: [Auf Anfrage (on demand)]</p> <p>Schreiboptionen:</p> <p>Existierende Datei: _____ Neue Datei: _____</p> <p align="center">OK Abbrechen Hilfe</p>
	<p>Abbildung 161: AxProtector - Dateiverschlüsselung "Dateityp hinzufügen"</p> <p>2. Geben Sie im "Name"-Feld einen beschreibenden Namen des Dateityps an. Dieser hat keinen Einfluss auf die Verschlüsselung.</p> <p>2. Geben Sie im "Dateiendung"-Feld die Dateierweiterung des Dateityps an, den Sie anlegen möchten, z.B. .txt für Textdateien.</p> <p>2. Legen Sie im "Player Überprüfung"-Auswahlfeld fest, ob bei der Entschlüsselung eine Überprüfung</p>

Element	Beschreibung	
	der Lizenzoptionen der zugreifenden Anwendung stattfindet.	
	Lizenzliste	Der Player (zugreifende Anwendung) muss mit einer Lizenz aus dieser Lizenzliste verschlüsselt sein.  Dies erlaubt Ihnen beispielsweise festzulegen, dass auf einen bestimmten Datentyp ausschließlich mit einer von Ihnen verschlüsselten Anwendung zugegriffen werden kann.
	No player check	Hier findet keine Überprüfung der zugreifenden Anwendung statt.
2. Legen Sie im "Dateizugriffsmodus" -Auswahlfeld fest, wie der für den geschützten Dateizugriff vorbereitete Player (Anwendung zur Anzeige der Datendatei) auf die verschlüsselte Datendatei zugreift. Mit dem Dateizugriffsmodus können Sie den Speicherbedarf und das Laufzeitverhalten konfigurieren.	 Die Wahl des passenden Modus hängt von der Art der Anwendung (des Players) und der Größe der Datei ab. Bei großen Videodateien sollte zum Beispiel die "Modus für große Dateien" Option verwendet werden. Bei kleinen Dateien (Konfigurationsdateien), auf die mehrmals zugegriffen wird bietet sich der "Auf einmal" Modus an.  Da bei der Dateiverschlüsselungen auch unterschiedliche Laufzeiteinstellungen für zugreifende Anwendung und die Daten gewählt werden können, gelten zur Laufzeit die jeweils restriktiveren Einstellungen.	
Auf Anfrage	<p>Auf Anfrage</p> <p>Der Player reserviert im Hauptspeicher Platz für die komplette zu lesende Datei, liest aber nur den benötigten Teil - genaugenommen alle 4 kByte Blöcke, in denen dieser Teil enthalten ist - und entschlüsselt diese Blöcke. Bei weiteren Zugriffen auf die geschützte Datei werden weitere benötigte Blöcke (on demand) nachgeladen und entschlüsselt. Ist der benötigte Teil in bereits geladenen Blöcken, wird das entschlüsselte Abbild im Speicher verwendet. So baut der Player nach und nach ein komplettes Speicherabbild der benötigten Datei auf.</p> <p> Dieser Modus benötigt viel Speicher (genauso viel wie die zu ladende Datei), bietet aber durch das Caching der entschlüsselten Daten eine gute Performance zur Laufzeit, wenn auf einen bereits entschlüsselten Block zugegriffen. Dieser Modus ist für lesenden und schreibenden Zugriff möglich.</p>	
Auf einmal	<p>Auf einmal</p> <p>Der Player reserviert im Hauptspeicher Platz für die komplette zu lesende Datei, liest diese komplett ein und entschlüsselt sie komplett. Weitere Zugriffe auf die geschützte Datei erfolgen über das entschlüsselte Abbild im Speicher.</p> <p> Dieser Modus benötigt viel Speicher (genauso viel wie die zu ladende Datei), bietet aber durch das Caching der entschlüsselten Daten eine gute Performance zur Laufzeit. Im Vergleich zum "Auf Anfrage" Modus benötigt dieser Modus mehr Zeit beim ersten Zugriff (wenn die Datei komplett geladen und entschlüsselt wird). Dafür liegt die Datei aber danach komplett entschlüsselt im Speicher und jeder weitere Zugriff ist performant. Dieser Modus ist für lesenden und schreibenden Zugriff möglich.</p>	
Modus für große Dateien	<p>Modus für große Dateien</p> <p>Der Player liest die gerade benötigten Teile der geschützten Datei ein und entschlüsselt diese. Die Daten werden im Speicher nicht als Cache gehalten.</p> <p> Dieser Modus benötigt keinen zusätzlichen Speicher. Wenn auf die gleichen</p>	

Element	Beschreibung	
		Daten mehrmals zugegriffen wird, werden die Daten jedes Mal neu gelesen und neu entschlüsselt. In diesem Modus ist nur lesender Zugriff möglich.
6. Legen Sie im "Schreiboptionen"-Bereich fest, wie die Dateien gespeichert werden.		
Existierende Datei		Dieser Bereich regelt über Einstellungen, wie Änderungen an einer bestehenden Datei gespeichert werden.
Original	Hier sind Änderungen zugelassen. War die Datei verschlüsselt wird sie wieder verschlüsselt. Unverschlüsselte Dateien werden unverschlüsselt gespeichert.	
No writing	Hier sind Schreibvorgänge nicht erlaubt, es besteht ausschließlich ein Read-Only-Zugriff.	
Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.	
Neue Datei		Dieser Bereich regelt über Einstellungen, wie Änderungen an einer neuen Datei gespeichert werden.
Plain	Neue Dateien werden grundsätzlich unverschlüsselt gespeichert.	
No writing	Neue Dateien können nicht gespeichert werden.  Es wird zwar eine neue Datei angelegt, darin aber keinerlei Daten abgespeichert.	
Lizenzliste	Hier werden Änderungen grundsätzlich über die in der ausgewählten Lizenzliste definierten Lizenzoptionen verschlüsselt.	

7.6.1.5 Zusammenfassung

In diesem Eingabefenster sehen Sie eine Zusammenfassung aller von Ihnen zuvor getroffenen Einstellungen zum automatischen Schutz Ihrer Anwendung ein.

Der Inhalt dieser Seite kann zur späteren Wiederverwendung in eine *.wbc Datei kopiert werden (WIBU Configuration Datei). Kopieren Sie den Inhalt in eine Textdatei und geben Sie der Datei die Endung *.wbc.



Sie können anschließend Ihre Anwendung mit diesen Einstellungen auch über die Kommandozeile-Eingabe schützen. Rufen Sie hierzu in der Kommandozeile AxProtector.exe @*.wbc [auf](#)
 ³¹⁹.

Alternativ dazu können Sie die entsprechende *.wbc -Datei auch über den "**Datei – wbc-Datei exportieren**" Menü-Eintrag erstellen.

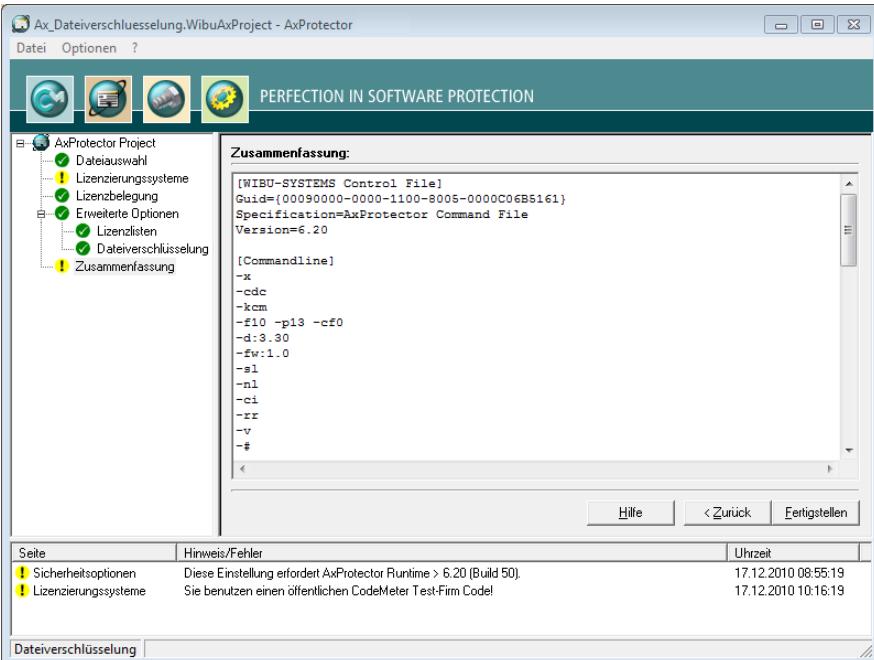


Abbildung 162: AxProtector - Dateiverschlüsselung "Zusammenfassung"

Element	Beschreibung
Fertigstellen	Startet die AxProtector Verschlüsselung mit den zuvor gesetzten Einstellungen.
Zurück	Erlaubt das Zurückkehren, um Änderungen der Einstellungen vornehmen zu können.

Das Ergebnis der Verschlüsselung mit allen relevanten Einstellungen wird in einem separaten Fenster angezeigt.

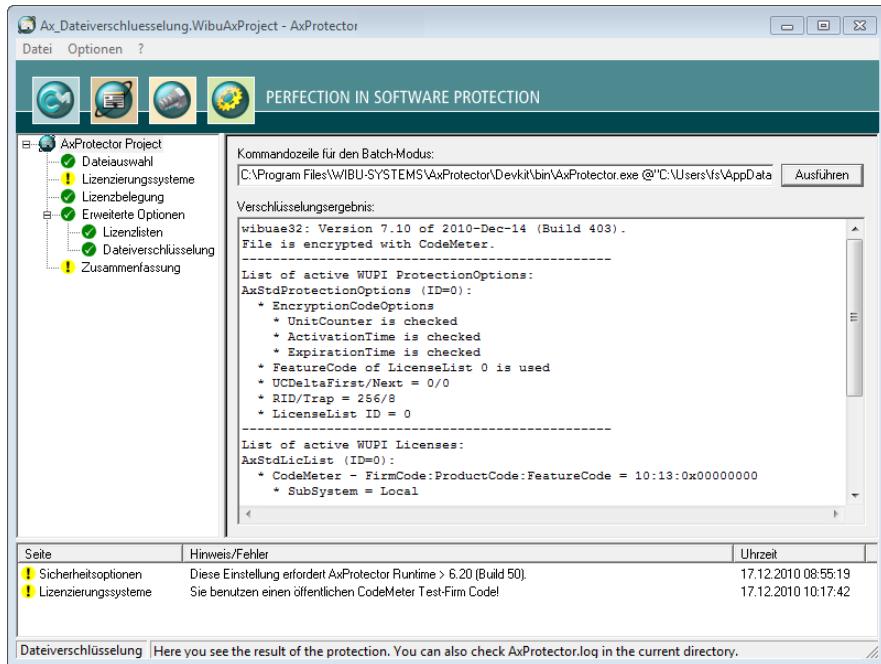


Abbildung 163: AxProtector - Dateiverschlüsselung "Verschlüsselungsergebnis"

Element	Beschreibung
Ausführen	<p>Sollten Sie aus irgendeinem Grund den Verschlüsselungsvorgang wiederholen müssen, so betätigen Sie die "Ausführen" Schaltfläche. Dann wird die links nebenstehende AxProtector Kommandozeile für den Batch-Modus ausgeführt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> i Die AxProtector Kommandozeile für den Batch-Modus können Sie auch über die Zwischenablage kopieren und in die Kommandozeile-Eingabe einfügen. Dort haben Sie dann die Möglichkeit noch gewünschte Änderungen vorzunehmen. </div>

7.7 Kommandozeilen-Optionen für AxProtector

Versionen der Kommandozeilen-Anwendung

Alternativ zur *AxProtector* Anwendungsoberfläche haben Sie auch die Möglichkeit, die betreffenden Einstellungen über die *AxProtector* Kommandozeilen-Optionen vorzunehmen.

Die Kommandozeilen-Anwendung gibt es in den folgenden Versionen. Alle befinden sich im Verzeichnis "%\Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\bin":

Version	Projekttypen
AxProtector.exe	
AxProtectorNet.exe	
AxProtectorMacX	
AxProtector.jar	
AxProtectorLin	in einer 32-Bit und 64-Bit Version



Welche Optionen für welchen *AxProtector* Projekttypen Gültigkeit besitzen, erkennen Sie an den Symbolen innerhalb einer separaten Zeile.

Syntax der Kommandozeile

Der Aufruf der Kommandozeile folgt dem untenstehenden Muster:

```
AxProtector-Aufruf -<Optionen> <Pfad und Name der zu schützenden Anwendung>
```

7.7.1 Grundsätzliche Einstellungen

Option -X

gilt für

linkt die statische Bibliothek des Lizenzierungssystems zur Anwendung, die geschützt werden soll.



Wird standardmäßig gesetzt.

Dieses Setzen erhöht die Sicherheit im Vergleich zur Standardeinstellung, die die dynamische Bibliothek dazulinkt.

Option -A

gilt für

sucht eine vorher festgelegte Lücke innerhalb der zu schützenden Anwendung, in die der Sicherheits-Code eingefügt wird.

Diese Lücke wird nur genutzt, wenn sie groß genug ist. Die Lücke muss eine *AxProtector* Signatur am Anfang aufweisen.

Option **-A[AES]**

gilt für    

legt den Verschlüsselungsalgorithmus fest (nur *CodeMeter*®).

7.7.2 Einstellungen zum Lizenzierungssystem

Option **-K[ICA][CM][WK]**

gilt für      

legt das Lizenzierungssystem fest.

Parameter **-KCA**

benutzt *CmActLicense*.

Parameter **-KCM**

benutzt *CmDongle* (Standard).

Parameter **-W_KK [-x]**

benutzt *WibuKey*. x steht für:

- 1 benutzt den Verschlüsselungsalgorithmus 1 (nur *WibuKey*).
- 2 benutzt den Verschlüsselungsalgorithmus 2 (nur *WibuKey*).
- 3 benutzt den Verschlüsselungsalgorithmus 3 (nur *WibuKey*).
- 4 benutzt den Verschlüsselungsalgorithmus 4 (nur *WibuKey*) (Standardeinstellung).
- 5 benutzt den Verschlüsselungsalgorithmus 5 (nur *WibuKey*).

 Die folgenden Optionen sollten sich unmittelbar hinter der **-K** Option befinden, da sie sich speziell auf das aktuell gesetzte Lizenzierungssystem beziehen.
Im Fall, dass Sie alle Lizenzierungssysteme für eine ausführbare Datei nutzen, bilden die vorgenommenen Einstellungen die Konfiguration für das gesetzte entsprechende Lizenzierungssystem.

Option **-Fx**

gilt für      

legt den gewünschten *Firm Code(x)* fest.

Eingabe eines ganzzahligen Wertes ohne Vorzeichen <n>.

Option **-Px**

gilt für      

legt den gewünschten Product Code (x) fest.

Im Fall von *WibuKey* den User Code.

Eingabe eines ganzzahligen Wertes ohne Vorzeichen <n>

Option	-CFx
gilt für	
	legt den gewünschten Feature Code (x) der Feature Map fest.
	Nur bei Verwendung von <i>CodeMeter®</i> .
	Eingabe eines ganzzahligen Wertes ohne Vorzeichen <n>.
Option	-RD([YYYYMonDD[HH:MM:SS[:<Timezone>]]][:now]) oder nach ISO-8601 mit T(Zeit)- und Z(Zonen)-Parameter: -RD([<yyyy>-<mm>-<dd>T<hh>:<mm>:<ss>[Z][±hh:mm oder ±hhmm] [±hh]])[:now]
gilt für	
	legt das gewünschte Erstelldatum (Release Date) für die Ver- und Entschlüsselung fest. (Nur <i>CmDongle</i> und <i>CmActLicense</i>)
	Die Angabe erfolgt im Format Jahr, Monat und optional Stunden, Minuten, Sekunden und der Zeitzone. Die Eingabe von [:now] übernimmt das aktuelle Datum.
	Erfordert <i>CodeMeter®</i> Version 4.30 und Firmware-Version 1.18.
Option	-D:v
gilt für	
	legt die Minimum Treiber-Version fest. Eingabe von v über (x.y). Standardeinstellung: <i>CodeMeter®</i> 4.20. Standardeinstellung: <i>WibuKey</i> 5.20.
Option	-FW:v
gilt für	
	legt die Minimum Firmware-Version fest. Eingabe von v über (x.y). Standardeinstellung: <i>CodeMeter®</i> 1.0 Wird nicht verwendet für <i>WibuKey</i>
Option	-S(L)[N W] C
gilt für	
	legt die Suchreihenfolge des Subsystems fest, auf dem nach gültigen Lizenzen gesucht werden soll. Die Optionen N und W können nur alternativ benutzt werden.
Parameter	-SL
	benutzt das lokale Subsystem (Lokal).

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

Parameter **-SN**

benutzt das Netzwerk-Subsystem (Netz).

Parameter **-SLN**

benutzt zuerst das lokale Subsystem, danach das Netzwerk-Subsystem (Netz).

Parameter **-SNL**

benutzt zuerst das Netzwerk-Subsystem (Netz), danach das lokale Subsystem (Lokal).

Parameter **-SW**

benutzt das Wide Area Network-Subsystem (WAN).

Parameter **-SLW**

benutzt zuerst das lokale Subsystem (Lokal), danach das Wide Area Network-Subsystem (WAN).

Parameter **-SWL**

benutzt zuerst das Wide Area Network-Subsystem (WAN), danach das das lokale Subsystem (Lokal).

Parameter **-SC**

sucht zuerst das lokale (Lokal) und danach das Netzwerk Subsystem (Netz) und im Falle eines gefundenen Netzwerkes das Netzwerklaufwerk.

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

gilt für 

legt den Netzwerkzugriff fest.

Parameter **-NC[A]**

convenient mode (Kompatibilitätsmodus): hier belegt jede gestartete Instanz im Netzwerk eine Lizenz (normal user limit), wobei lokal der Zugriff unbegrenzt ist (no user limit).

 Da dies der Standard-Lizendbelegung von WibuKey entspricht, sichert diese Option die Kompatibilität bei gleichzeitigem Einsatz beider Lizenzierungssysteme.

(A: benutzt auto cancel (nur WibuKey)).

Parameter **-NL[A]**

normal user limit: hier belegt jede gestartete Instanz eine Lizenz. Dabei spielt es keine Rolle, ob der CmContainer lokal oder im Netzwerk gefunden wird.

(A: benutzt auto cancel (nur WibuKey)).

Parameter **-NN**

no user limit: hier können beliebig viele Instanzen lokal oder im Netzwerk gestartet werden wobei keine Lizenzen belegt werden.

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

Parameter `-NS`

station share: hier belegen mehrere gestartete Instanzen auf einem Client lediglich eine Lizenz.

Diese Option setzen sie beispielsweise ein, wenn Sie dem Anwender die Möglichkeit bieten möchten, die geschützte Anwendung mehrmals zu starten. Auf Terminal Server belegt jede Session eine Lizenz. In virtuellen Maschinen belegt jede virtuelle Maschine eine Lizenz.

Parameter `-NX [A]`

exclusive mode: hier wird nur eine gestartete Instanz pro PC zugelassen. Dies entspricht dem restriktivsten Modus, d.h. auf dem lokalen System und dem Netzwerk-Client wird jede Programmkopie als eine Lizenz gewertet.

Wird der exklusive Zugriff gesetzt, wird nicht mehr automatisch der Runtime Check intern aktiviert. Dieser muss zukünftig explizit gesetzt werden ([Option '-car'](#) ⓘ³⁰⁰).
(A: benutzt auto cancel (nur WibuKey)).

7.7.3 Einstellungen zu Verschlüsselungsvorgängen

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

verschlüsselt die ausführbare Datei mit einer automatischen Verschlüsselung.

Parameter `-CAA <1>`



aktiviert die Sicherheitsoptionen (Advanced Protection Schemes, APS).

<1> umfasst die Optionen [0, 15]



Zur Verwendung von mehr als nur einer Sicherheitsoption (APS) können die Optionen 1, 2, 4 und 8 mit „oder“ verknüpft werden.

APS2 und APS8 schließen sich gegenseitig aus. Im Falle, dass beide gesetzt worden sind, wird automatisch -CAA8 verwendet.

Option **Beschreibung**

- 1 Ressourcenverschlüsselung wird verwendet (APS 1)
- 2 Statische Modifikation wird verwendet (APS 2)
- 4 Dynamische Modifikation wird verwendet (APS 3)
- 8 Erweiterte Statische Modifikation wird verwendet (APS 4)



CAA6 verwendet APS 2 und 3
CAA13 verwendet APS 1, 4 und 8

Parameter	-CACT<,u>
gilt für	   
	überprüft die <i>CmContainer</i> Systemzeit in Bezug auf die PC-Zeit. Ein geschützte Anwendung läuft nur, wenn die PC-Zeit in einem Zeitfenster < <i>t</i> > Minuten vor und optional < <i>u</i> > Minuten nach der <i>CmContainer</i> System Time liegt.
Parameter	-CAD<m>
gilt für	   
	gibt den Datei-Zugriffsmodus für die automatische Entschlüsselung von Dateien an, die mit der Option –CD verschlüsselt wurden.
	<ol style="list-style-type: none"> 0 entschlüsselt die Inhalte der Datei blockweise (4KB) auf Bedarf (on demand) 1 entschlüsselt den kompletten Inhalt der Datei auf einmal beim ersten Zugriff (at once). Je nach Größe der Datei kann es beim Zugriff zu Verzögerungen kommen 2 verhindert, dass bei der Dateiverschlüsselung die geschützte Anwendung Daten überhaupt noch auf die Platte schreiben kann (view only). Hier wird sämtliches Schreiben in eine Datei verhindert. 4 entschlüsselt die Inhalte auch sehr großer Dateien (z.B. 500 MB große MPG3 Dateien), die mit der Dateiverschlüsselung verwendet werden (read und decrypt on demand). Bei diesem Modus ist das (Zurück-)Schreiben von Dateien grundsätzlich abgeschaltet.
Parameter	-CAE
gilt für	   
	aktiviert die sofortige Erkennung von ‘plug-out’ (nur <i>CmDongle</i>): Auswurferkennung.
	<p> Im Fall, dass die <i>WibuBox</i> oder der <i>CmDongle</i> einmal irrtümlich abgezogen worden ist, wird dem Anwender über eine sinnvolle Zahl maximal erlaubtes Ignorieren die Möglichkeit gegeben, die Hardware wieder anzustecken, ohne dass die Anwendung gleich beendet wird und eventuell Daten verloren gehen. Damit verbleibt genügend Zeit, Daten abzuspeichern.</p>
Parameter	-CAG<1>
gilt für	   
	aktiviert Anti-Debugging-Mechanismen (Anti-Debugging-Checks, ADC). <i><1></i> umfasst die Optionen [0, 127]
	<p> Zur Verwendung von mehr als nur eines Anti-Debugging-Mechanismus (ADC) können die Optionen mit „oder“ verknüpft werden.</p>
Option	Beschreibung
1	überprüft, ob ein Debugger an Ihre Anwendung gebunden ist. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC1).

Parameter	-CAG<1>																						
	<table border="1"> <thead> <tr> <th>Option</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>überprüft zusätzlich auf Kernel-Debugger-Programme, wie z.B. "SoftICE". Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC2).</td> </tr> <tr> <td>4</td> <td>überprüft in einer erweiterten Suche auf Debug-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC3).</td> </tr> <tr> <td>8</td> <td>überprüft auf sämtliche Debugger-Programme. Dabei sind keine Debugger-Programme mehr erlaubt, d.h. auch keine innerhalb von Entwicklungsumgebungen (IDE) (z.B. Visual Studio, Delphi). Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC4).</td> </tr> <tr> <td>16</td> <td>Sperren des Lizenz-Eintrages und damit der Hardware, wenn ein Debugger erkannt wird (ADC5). <div style="border: 1px solid black; padding: 5px;"> <p>Damit diese Option genutzt werden kann, muss der <i>CmContainer</i> vom Entwickler über den Firm Access Counter vorher entsprechend programmiert worden sein. Der Firm Access Counter (FAC) liegt auf der Firm Item Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob eine Firm Item Ebene für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535. Er kann auf jeden anderen belieben Wert programmiert werden.</p> <p>Das Sperren der Hardware funktioniert nur, wenn der Firm Item Access Counter auf einen Wert <> 0, beziehungsweise <> -1 gesetzt ist.</p> </div> </td> </tr> <tr> <td></td> <td>Abhängig von den Einstellungen können alle Lizenzen eines Software-Herstellers bei einer Hacker-Attacke gesperrt werden.</td> </tr> <tr> <td></td> <td>Der Besitzer / Anwender des gesperrten <i>CmContainers</i> muss zwecks Aufhebung der Sperre mit dem Software-Hersteller in Kontakt treten. Ob und wie oft die Aufhebung bewilligt wird, hängt von der Verfahrensweise des Software-Herstellers ab.</td> </tr> <tr> <td>32</td> <td>fügt der Anwendung einen Mechanismus hinzu, der das Anhängen eines Debuggers an die laufende Anwendung verhindert (generische Debugger-Erkennung) (ADC6).</td> </tr> <tr> <td>64</td> <td>erkennt ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies (ADC7).</td> </tr> <tr> <td>128</td> <td>Die Hardware-Sperre wird nur bei einem gültigen Firm Access Counter durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).</td> </tr> <tr> <td>256</td> <td>Das Herunterzählen des Firm Access Counter um den Wert 1 wird durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).</td> </tr> </tbody> </table>	Option	Beschreibung	2	überprüft zusätzlich auf Kernel-Debugger-Programme, wie z.B. "SoftICE". Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC2).	4	überprüft in einer erweiterten Suche auf Debug-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC3).	8	überprüft auf sämtliche Debugger-Programme. Dabei sind keine Debugger-Programme mehr erlaubt, d.h. auch keine innerhalb von Entwicklungsumgebungen (IDE) (z.B. Visual Studio, Delphi). Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC4).	16	Sperren des Lizenz-Eintrages und damit der Hardware, wenn ein Debugger erkannt wird (ADC5). <div style="border: 1px solid black; padding: 5px;"> <p>Damit diese Option genutzt werden kann, muss der <i>CmContainer</i> vom Entwickler über den Firm Access Counter vorher entsprechend programmiert worden sein. Der Firm Access Counter (FAC) liegt auf der Firm Item Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob eine Firm Item Ebene für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535. Er kann auf jeden anderen belieben Wert programmiert werden.</p> <p>Das Sperren der Hardware funktioniert nur, wenn der Firm Item Access Counter auf einen Wert <> 0, beziehungsweise <> -1 gesetzt ist.</p> </div>		Abhängig von den Einstellungen können alle Lizenzen eines Software-Herstellers bei einer Hacker-Attacke gesperrt werden.		Der Besitzer / Anwender des gesperrten <i>CmContainers</i> muss zwecks Aufhebung der Sperre mit dem Software-Hersteller in Kontakt treten. Ob und wie oft die Aufhebung bewilligt wird, hängt von der Verfahrensweise des Software-Herstellers ab.	32	fügt der Anwendung einen Mechanismus hinzu, der das Anhängen eines Debuggers an die laufende Anwendung verhindert (generische Debugger-Erkennung) (ADC6).	64	erkennt ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies (ADC7).	128	Die Hardware-Sperre wird nur bei einem gültigen Firm Access Counter durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).	256	Das Herunterzählen des Firm Access Counter um den Wert 1 wird durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).
Option	Beschreibung																						
2	überprüft zusätzlich auf Kernel-Debugger-Programme, wie z.B. "SoftICE". Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC2).																						
4	überprüft in einer erweiterten Suche auf Debug-Programme, die eventuell parallel zu Ihrer Anwendung laufen, auch Cracker Tools wie ImpREC werden erkannt. Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC3).																						
8	überprüft auf sämtliche Debugger-Programme. Dabei sind keine Debugger-Programme mehr erlaubt, d.h. auch keine innerhalb von Entwicklungsumgebungen (IDE) (z.B. Visual Studio, Delphi). Wird ein Debugger gefunden, wird Ihre Anwendung nicht gestartet (ADC4).																						
16	Sperren des Lizenz-Eintrages und damit der Hardware, wenn ein Debugger erkannt wird (ADC5). <div style="border: 1px solid black; padding: 5px;"> <p>Damit diese Option genutzt werden kann, muss der <i>CmContainer</i> vom Entwickler über den Firm Access Counter vorher entsprechend programmiert worden sein. Der Firm Access Counter (FAC) liegt auf der Firm Item Ebene eines <i>CmContainers</i>. Über diesen Zähler ist es möglich zu kontrollieren, ob eine Firm Item Ebene für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535. Er kann auf jeden anderen belieben Wert programmiert werden.</p> <p>Das Sperren der Hardware funktioniert nur, wenn der Firm Item Access Counter auf einen Wert <> 0, beziehungsweise <> -1 gesetzt ist.</p> </div>																						
	Abhängig von den Einstellungen können alle Lizenzen eines Software-Herstellers bei einer Hacker-Attacke gesperrt werden.																						
	Der Besitzer / Anwender des gesperrten <i>CmContainers</i> muss zwecks Aufhebung der Sperre mit dem Software-Hersteller in Kontakt treten. Ob und wie oft die Aufhebung bewilligt wird, hängt von der Verfahrensweise des Software-Herstellers ab.																						
32	fügt der Anwendung einen Mechanismus hinzu, der das Anhängen eines Debuggers an die laufende Anwendung verhindert (generische Debugger-Erkennung) (ADC6).																						
64	erkennt ob die Anwendung in einer virtuellen Maschine gestartet werden soll und verhindert dies (ADC7).																						
128	Die Hardware-Sperre wird nur bei einem gültigen Firm Access Counter durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).																						
256	Das Herunterzählen des Firm Access Counter um den Wert 1 wird durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).																						

Parameter	-CAG<1>
gilt für	 aktiviert Anti-Debugging-Mechanismen (Anti-Debugging-Checks, ADC). <1> umfasst die Optionen [0 , 17]

Parameter

-CAG<1>



Zur Verwendung von mehr als nur eines Anti-Debugging-Mechanismus (ADC) können die Optionen bis zu einem Maximum von 17 mit „oder“ verknüpft werden.
Die Standardeinstellung für <1> ist 17.

Option	Beschreibung
0	kein Debugger-Check. Standardeinstellung, wenn -CAG nicht angegeben.
1	überprüft mit einem einfachen Debugger-Check (ADC1).
16	Sperren des Lizenz-Eintrages und damit der Hardware, wenn ein Debugger erkannt wird (ADC5). Damit diese Option genutzt werden kann, muss der <i>CmContainer</i> vom Entwickler über den Firm Access Counter vorher entsprechend programmiert worden sein. Der Firm Access Counter (FAC) liegt auf der Firm Item Ebene eines <i>CmContainers</i> . Über diesen Zähler ist es möglich zu kontrollieren, ob eine Firm Item Ebene für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535. Er kann auf jeden anderen belieben Wert programmiert werden. Das Sperren der Hardware funktioniert nur, wenn der Firm Item Access Counter auf einen Wert <> 0, beziehungsweise <> -1 gesetzt ist.
128	Abhängig von den Einstellungen können alle Lizenzen eines Software-Herstellers bei einer Hacker-Attacke gesperrt werden. Der Besitzer / Anwender des gesperrten <i>CmContainers</i> muss zwecks Aufhebung der Sperre mit dem Software-Hersteller in Kontakt treten. Ob und wie oft die Aufhebung bewilligt wird, hängt von der Verfahrensweise des Software-Herstellers ab.
256	Die Hardware-Sperre wird nur bei einem gültigen Firm Access Counter durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>). Das Herunterzählen des Firm Access Counter um den Wert 1 wird durchgeführt (nur in Kombination mit ADC5 und <i>CmContainer</i>).

Parameter

-CAG<1>

gilt für



aktiviert Anti-Debugging-Mechanismen (Anti-Debugging-Checks, ADC).
<1> umfasst die Optionen [0, 7]



Zur Verwendung von mehr als nur eines Anti-Debugging-Mechanismus (ADC) können die Ebenen bis zu einem Maximum von 7 mit „oder“ verknüpft werden.
Die Standardeinstellung für <1> ist 7.

Option	Beschreibung
0	kein Anti-Debugging-Mechanismus wird verwendet. Standardeinstellung, wenn -cag nicht angegeben wird.

Parameter **-CAG<1>**

Option	Beschreibung
1	überprüft auf Erkennung des JVMPPI (Java Virtual Machine Profiler Interface). Mit JVMPPI kann man die Java Virtual Machine manipulieren, damit diese Nachrichten an nativen Code schickt. Besonders das Ereignis JVMPPI_EVENT_CLASS_LOAD_HOOK kann dazu verwendet werden, den unverfälschten Byte Code der aktuell geladenen Klasse abzufangen. Das Aktivieren dieser Option verhindert dieses Vorgehen.
2	überprüft auf Manipulationen durch Callback-Funktionen. D.h. der Zugriff auf Objekte anderer Klassen wird geprüft.
4	überprüft auf die Java Virtual Machine für Java 6 (1.6).

Parameter **-CAL**gilt für 

Beschränkt die automatische Verschlüsselung auf spezifizierte Bereiche .

Parameter **-CAM**gilt für 

fügt dem Systemmenü das Menü-Element 'Control' und 'About' hinzu.

Parameter **-CAR<t>, <m>**gilt für 

fügt der automatischen Verschlüsselung eine Überprüfung während der Laufzeit hinzu (runtime check) zu.

Die Aktivierung erfolgt alle <t> Sekunden. Die Standardeinstellung beträgt 300 Sekunden (5 Minuten).

<m> gibt an, wie oft der Anwender eine fehlgeschlagene Überprüfung ignorieren kann.

Parameter **-CAS<p>**gilt für 

gibt den prozentualen Umfang [0 . . . 100] an, zu dem der Quelltext der zu schützenden Anwendung verschlüsselt wird.

Die Standardeinstellung beträgt 75 Prozent.

Parameter **-CAT(t)(,u)**gilt für 

Bei jedem Applikationsstart wird versucht eine zertifizierte Zeit (Certified Time) zu holen (Zeitzertifikat setzen). Dann startet die Applikation, unabhängig davon, ob die Zeit geholt werden

Parameter **-CAT(t)(,u)**

konnte oder nicht und schreibt sie in den *CmContainer*.

Danach startet die Anwendung, wenn die Zeitdifferenz zwischen der zertifizierten Zeit und der Systemzeit des PC nicht größer ist als in <t> Stunden angegeben.

<u> in Stunden gibt die erlaubte Zeitspanne an, innerhalb der sich die Differenz zwischen der zertifizierten Zeit und der Systemzeit bewegen darf, ohne dass eine Aktualisierung der zertifizierten Zeit erfolgt (nur *CodeMeter®*).



<t> muss größer oder gleich <u> sein.

Parameter **-CAV**

gilt für



fügt der automatisch verschlüsselten Anwendung eine Virusüberprüfung hinzu.

Parameter **-CAZ**

gilt für



speichert die Verschlüsselungszeit (*CmContainer System Time*) in der geschützten Anwendung. Die Anwendung läuft dann nur, wenn die PC-Zeit nach dem Zeitpunkt der Verschlüsselung liegt.



Erfordert mindestens *CodeMeter® 4.10*.

Option **-CC[[A[a:s][E],[H],[I],[M],[O],[Q],[R],[S],[X]]]**

setzt Kompatibilitätsparameter.

Parameter **-CCA**

gilt für



definiert das Zielsystem/Subsystem in Kombination mit der Option -ccx für .NET Executables inklusive Debugging verschlüsselter Anwendungen.

<a> enthält die Zielplattform [1,2]. 1: x86 / Intel 32 Bit
2: ArmV4i

<s> enthält das Subsystem [9] 9: Windows CE System

Parameter **-CCB**

gilt für



wenn die Option nicht gesetzt ist, wird die .reloc-Sektion zur Erhöhung der Sicherheit in ein proprietäres Format übersetzt, das von einem (Windows-)Loader nicht erkannt wird.

Parameter **-CCE**gilt für   

definiert, dass das PE nicht vergrößert wird.

Parameter **-CCH**gilt für   

verhindert sämtliches globales Hooking in einer geschützten Anwendung.

Parameter **-CCI**gilt für   erlaubt die geschützte Anwendung so zu verwenden, dass sich durch den hinzugefügten Schutz die eventuell bestehende Ladereihenfolge von DLLs nicht verändert wird. Dadurch ist die ehemalige Option "**-ccm**" nicht mehr notwendig.**Parameter** **-CCM**gilt für   gibt an, dass die geschützte Anwendung die Bibliothek **wibucrt32/64.dll** laden soll, um Probleme mit der Ladereihenfolge durch die Bibliothek **msvcr*.dll** zu beheben.**Parameter** **-cco**gilt für   

aktiviert einen besonderen Umgang mit ActiveX / OCX images.

Parameter **-ccq**gilt für   

räumt Lizenzen der geschützten Anwendung nicht beim Aufruf von WM_QUIT, sondern erst beim Aufruf von ExitProcess().

Parameter **-CCR**gilt für   

deaktiviert die Umbenennung von Sektionen.

Parameter **-CCS**gilt für     gibt an, dass alle Lizenzen aus dem gleichen *CmContainer/WibuBox* vom gleichen Rechner kom-

Parameter **-ccs**

men müssen wie die der ersten gefunden Lizenz.

Parameter **-ccx**

gilt für



gibt an, dass auch sogenannte mixed-mode Assemblies geschützt werden können. Damit können .NET Assemblies verschlüsselt werden, die sich nicht mit dem AxProtector .NET verschlüsseln lassen. Es können neben Win32 auch Win64/x64 mixed-mode Assemblies mit dem nativen AxProtector verschlüsselt werden.

Die Bibliothek `wbcor32/64.dll` wird benötigt, damit die geschützte Assembly läuft.

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

gilt für



definiert Optionen beim Laden geschützter Shared Objects

Parameter **-CCD**

gilt für



definiert Kennzeichner für `dlopen` beim Laden von shared objects. Die Kennzeichner können mit Linux-Konstanten ver-odert werden:

- RTLD_LAZY 0x00001
- RTLD_NOW 0x00002
- RTLD_NOLOAD 0x00004
- RTLD_DEEPBIND 0x00008
- RTLD_GLOBAL 0x00100 (wenn nicht gesetzt, dann gilt RTLD_LOCAL)

Parameter **-cck**

gilt für



entlädt das Shared Object nicht explizit.

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

gilt für



verschlüsselt eine Datei 1:1 und fügt einen Header mit Verschlüsselungsinformationen hinzu. Diese Dateien können von einer automatisiert verschlüsselten Anwendung automatisch entschlüsselt werden.

Parameter **-CDC**

gilt für



übernimmt die Dateinamen-Erweiterung aus der *.wbc-Datei.

Parameter**-CDH**

gilt für



gibt an, dass der Zugriff auf die Lizenz gehalten wird, wenn der Player die Datei schließt. Damit wird ein Handle offen gelassen. Die Option gilt pro Datei.

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

gilt für



gibt das benutzte Lizenzierungssystem an:

CA benutzt *CmActLicense*

CM benutzt *CmDongle*

WK benutzt *WibuKey*.

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

gilt für



gibt den Firm Code (x) an mit dem die Anwendung verschlüsselt sein muss, wenn sie die verschlüsselte Datei öffnen soll.

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

gilt für



gibt den Product Code (y) an mit dem die Anwendung geschützt werden muss. Der Firm Code muss vorher gesetzt sein.

Es kann mehr als ein Firm Code - Product Code Paar gesetzt sein.

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

gilt für



verschlüsselt explizit definierte Quelltext-Bereiche innerhalb der ausführbaren Dateien, um sie mit *IxProtector* zu nutzen.

Parameter**-CIH**

gilt für



setzt fest, dass WupiXXX-Funktionen nicht dynamisch in den *IxProtector*-Ablauf eingreifen (hooking).

Parameter**-CIN**

gilt für



setzt fest, dass im Fehlerfall keine Fehlermeldungen angezeigt werden.

Parameter **-CID**

gilt für



verschlüsselt explizit definierte Quelltext-Bereiche innerhalb der ausführbaren Dateien, um sie mit *lxProtector* zu nutzen. WUPI wird jetzt auch unter Mac und Linux unterstützt (Option **-cid**). Hier wird aktuell mit einer dynamisch geladenen Variante gearbeitet.

Option **-CI**

gilt für



aktiviert die Verschlüsselung von expliziten Teilen des Quelltextes (Klassen / Methoden) innerhalb der ausführbaren Datei, um sie mit *lxProtector* zu verschlüsseln.

Welche Klassen / Methoden verschlüsselt werden wird durch das Setzen verschiedener Annotationen gesteuert (für Details dieser Option, die mit AxProtector Version 9.0 eingeführt wurde, siehe [Java-spezifische Einstellungen](#)^[316]).

Option **-CK<n>**

gilt für



puffert den RID-Schlüssel der Anwendung für <n> Sekunden in den Cache-Speicher.
<n> kann Werte zwischen 0 und 255 annehmen.

Option **-CO(n)**

gilt für



Diese Option ist gültig für den Projekttyp *AxProtector .NET*.

setzt fest, welche Elemente obfuskriert werden (ab AxProtector Version 8.40).

<n> umfasst die Optionen [0 , 15]

Obfuscierung ersetzt les- und nachvollziehbare Element-Namen durch maschinengenerierte Benennungen, verschleiert Programm-Informationen und schützt vor einem Reverse Engineering.



Zur Verwendung von mehr als nur einer Obfuscierungsoption können die Optionen bis zu einem Maximum von 15 mit „oder“ verknüpft werden.

Die Standardeinstellung für <n> beträgt 0.

Option Beschreibung

- 0 keine Elemente werden obfuskriert.
- 1 Private Elemente werden obfuskriert.
- 2 Interne Elemente werden obfuskriert.
- 4 Geschützte Elemente obfuskriert.
- 8 Public Elemente werden obfuskriert.

Beginnend mit AxProtector Version 8.40 gibt es die Möglichkeit, die Obfuscierung durch Attribute zu beeinflussen.

Option -CO(n)

Hierfür wird das Obfuscation-Attribut aus dem Namespace `System.Reflection` benutzt.

Das Attribut kann Klassen, Methoden, Feldern und Properties zugewiesen werden.

Folgende `Named Parameter` für das Attribut sind erlaubt:

Parameter	Werte	Beschreibung
<code>Exclude</code>	<code>true / false</code> Der Standardwert ist <code>true</code> .	nimmt das Element von der Obfuscierung aus
<code>ApplyToMembers</code>	<code>true / false</code> Der Standardwert ist <code>true</code> .	Einstellungen gelten für alle Member, wenn das Attribut einer Klasse zugewiesen wurde
<code>StripAfterObfuscation</code>	<code>true / false</code> Der Standardwert ist <code>true</code> .	Obfuscierungsattribut wird beim Obfuscieren entfernt
<code>Feature</code>		wird ignoriert

Die Obfuscierungsattribute werden immer ausgewertet.

Option -CPA

gilt für 

deaktiviert die Verschlüsselung von Eigenschafts-Accessoren.

Option -CMD<n>

gilt für 

aktiviert das Wiederverschlüsseln von Methoden nach Nichtbenutzung.

<n> erlaubt die Angaben von Sekunden.

Option -CML<n>

gilt für 

verschlüsselt nur Methoden mit einer Mindestgröße von <n> Bytes.

<n> hat den Standard-Wert von 10. Bei Angabe eines Wertes von 0 wird die Option deaktiviert.

Option -EC

gilt für 

verschlüsselt Klassen-Konstruktoren im .NET (MSIL) Code.

Option -CP<>

gilt für 

installiert einen Aufräummechanismus, der alle angelegten Dateien und Registry-Einträge beim Beenden einer Anwendung löscht, wenn diese von einem *CmContainer* gestartet wurde.

<1> bewirkt, dass alle gelöschten Einträge in eine Log-Datei geschrieben werden, die im Verzeichnis der geschützten Anwendung liegt..

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

gilt für     

setzt zusätzliche Überprüfungen während des Ver- und Entschlüsselungsprozesses fest.

Parameter **-EA**

gilt für     

aktiviert die Überprüfung der Activation Time (*CodeMeter®*).

- C prüft, falls die Product Item Option Activation Time vorhanden ist.
- I ignoriert die Product Item Option Activation Time (*CodeMeter®*).
- R erfordert die Product Item Option Activation Time.

Parameter **-EE**

gilt für     

aktiviert die Überprüfung der Expiration Time.

- C prüft, falls die Product Item Option Expiration Time vorhanden ist.
- I ignoriert die Product Item Option Expiration Time (nur *CodeMeter®*).
- R erfordert die Product Item Option Expiration Time.

Parameter **-EF**

gilt für     

aktiviert die Verringerung des Firm Access Counter (nur *CodeMeter®*).

Parameter **-EM**

gilt für     

aktiviert die Überprüfung des Wartungszeitraumes (*Maintenance Period*).

- C prüft, falls die Product Item Option Wartungszeitraum (*Maintenance Period*) vorhanden ist.
- I ignoriert die Product Item Option Wartungszeitraum (*Maintenance Period*) (nur *CodeMeter®*).
- R erfordert die Product Item Option Wartungszeitraum (*Maintenance Period*).

Parameter **-ET**

gilt für     

Parameter

-ET

erzwingt die Aktualisierung der Certified Time nach dem Einschalten.



Diese Option setzt eine aktivierte Expiration Time voraus.

Parameter

-EU

gilt für



aktiviert die Überprüfung des Unit Counter und das Herunterzählen um das Dekrement <n>.

S prüft und vermindert nur beim Start der geschützten Anwendung.

R prüft und vermindert bei jeder Überprüfung zur Laufzeit.

Die Option R beinhaltet die Option S.

Für die Optionen R und S sind die folgenden Optionen verfügbar:

C prüft ob die PIO existiert (Standard-Einstellung).

R(R) erfordert die Product Item Option Unit Counter.

<n> definiert die Zahl, um die heruntergezählt wird. Die Standardeinstellung ist 0.

I ignoriert die Product Item Option Unit Counter (nur CodeMeter®).

z.B. -eurr2 aktiviert einen erforderlichen Unit Counter bei jeder Überprüfung während der Laufzeit und zählt diesen dabei um die Zahl 2 herunter.

Option

-RIDx[,y]

gilt für



gibt die Anzahl der RID Varianten (x) und Fallen (y) an.

Option

-RIDIx[.y]

gilt für



gibt die Anzahl der RID-Varianten (x) und Trap-Varianten (y) bei der Verwendung von *lxProtector* (WUPI).

Option

-G[o,!][: "Marker",.]

gilt für



schließt den angegebenen Bereich aus der Verschlüsselung aus.

<o> legt die Ausnahme am Beginn des Bereiches fest.

<1>

legt die Länge des auszuschließenden Bereiches fest (nur).

"Marker" identifiziert eine Textmarkierung innerhalb des Quelltextes, der den Anfang des Bereiches anzeigen, der von der Verschlüsselung ausgenommen werden soll.

Option **-FW**

gilt für

setzt beim Verschlüsseln die minimale Firmware-Version.

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

gilt für

gibt die Schwellenwerte für Warnungen an.

Parameter **-WC[t]**

gilt für

setzt den Schwellenwert <t> in Stunden für die Certified Time.

Parameter **-WE[t]**

gilt für

setzt den Schwellenwert <t> in Tagen für die Expiration Time.

Parameter **-WP[t]**

gilt für

aktiviert eine Warnung, wenn die Usage Period noch nicht aktiviert ist.

Parameter **-WU[c]**

gilt für

setzt den Schwellenwert <c> in Einheiten für den Unit Counter.

Option **-SILVERLIGHT(3|4)**

gilt für

gibt an, dass eine Silverlight DLL verschlüsselt wird (nur *CodeMeter®*).

- 3 die zu schützende DLL ist für Silverlight 3 kompiliert (Standard-Einstellung).
- 4 die zu schützende DLL ist für Silverlight 4 kompiliert.

Option **-XAP:Dateiname**

gilt für

weitert die zu schützende Silverlight DLL über die zugehörige XAP-Datei aus, verschlüsselt und ersetzt die alte Datei durch eine Kopie der XAP-Datei (nur *CodeMeter®*).

Der Datei muss mit der Erweiterung angegeben werden.

Ist nur verwendbar zusammen mit Option **-SILVERLIGHT(3|4)**

7.7.4 Einstellungen mit Bezug zur Laufzeit

Option **-I**

gilt für    

gibt an, dass die Fehlerbehandlung für Plugin DLL-Dateien benutzt wird.

Diese Option funktioniert ausschließlich mit DLL-Dateien. Wenn das Plugin dynamisch geladen wird und kein Lizenzierungssystem verbunden ist, schließt das Plugin nicht die gesamte Anwendung und zeigt auch keine Fehlermeldung an.

Option **-L:xx**

gilt für    

gibt die Sprache der benutzerdefinierten Meldungstexte an.

cn: setzt Chinesisch

de: setzt Deutsch

fr: setzt Französisch

jp: setzt Japanisch

us: setzt Englisch (Standardeinstellung)

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

gilt für     

gibt den Ausgabetext für Meldungstexte der geschützten Anwendung aus

"msg" enthält die Zeichenkette für das gewünschte Ereignis.

Zeilenumbrüche, Tabulatoren, Hochkommas, etc. können im Ausgabetext angegeben werden, indem "\n", "\r", "\t", "\v" in der Zeichenkette an der gewünschten Stelle gesetzt werden

Parameter **-MA**

gilt für     

definiert den Anwendungsnamen, der an den Server übermittelt und dann in *CodeMeter WebAdmin* angezeigt wird. Es ist kein Standard Anwendungsname gesetzt. Wenn die Option nicht gesetzt ist, wird der intern verwendete Name der ausführenden Datei gesetzt.

Parameter **-MC**

gilt für     

enthält den Text, der im Eintrag 'Über' des System-Menüs angezeigt wird.

Parameter **-ME**

gilt für     

enthält den Text, der bei einem aufgetretenen Fehler angezeigt wird.

Parameter	-MI
gilt für	    
	enthält den benutzerdefinierten Fehlertext, der angezeigt wird, wenn die benötigte Runtime/Treiber des Lizenzierungssystems nicht installiert ist. Als Fehlercode wird hier der bereits existierende WUPI Fehlercode <code>wibu::UpiErrorLicenseModuleNotLoaded</code> an die User Message DLL übergeben.
Parameter	-MS
gilt für	    
	enthält den Text, der beim Start der Anwendung angezeigt wird.
Parameter	-MT
gilt für	    
	enthält den Text, der beim Auslaufen einer Expiration Time angezeigt wird.
Parameter	-MU
gilt für	    
	enthält den Text, angezeigt wird, wenn ein Unit Counter den Wert 0 erreicht.
Parameter	-MWC
gilt für	    
	enthält den Text, der angezeigt wird, wenn bei aktivierter Option <code>-wc<t></code> die Certified Time zur System Time eine zu große Differenz aufweist.
Parameter	-MWP
gilt für	    
	enthält den Warntext, der beim Start einer Anwendung ausgegeben werden, wenn eine vorhandene Usage Period noch nicht aktiviert wurde.
Parameter	-MWT
gilt für	    
	enthält den Text, der angezeigt wird, wenn demnächst das Ende der Expiration Time oder Usage Period erreichen sollte.
Parameter	-MWU
gilt für	    

Parameter | -MWU

enthält den Text, der angezeigt wird, wenn demnächst der Unit Counter den Wert 0 erreichen sollte.

Option | -U[:DateiName]

gilt für   

ruft die vom Benutzer implementierte Nachrichten-DLL auf wenn vorhanden.

Die Notation folgt der Regel `UserMsgXX.dll`, wobei XX dem Länderkürzel entspricht, z.B. De, Us, etc..

Wenn zusätzlich auch DateiName angegeben wird, folgt die Notation der Nachrichten-DLL der Regel `DateiNameXX.dll`, wobei xx die Länderkürzel Us, Sa, Cn, Dk, Nl, Fr, De, Gr, It, Hu, Jp, Ko, Br, Es,

Se, Tw besitzen kann (nur Projekttyp  .

gilt für 

Die Klasse muss eine Unterklasse des `com.wibu.xpm.MessageHandler` sein. Z.B. `com.wibu.xpmSwingMessageHandler` als Standard Message-Handler des Runtime-Paketes.

Option | -UM[:Dateiname]

gilt für 

ruft die benutzerdefinierte Meldungs-Assembly UserMsg auf, falls diese vorhanden ist.

Wenn [:Dateiname] mit angegeben wird, so trägt die implementierte Message-Assembly den Namen <Dateiname>.dll.



Bitte geben Sie den Namen ohne die *.dll Dateierweiterung ein!.

Option | -UI

gilt für 

implementiert die Message-Assembly inline, die über eine *.ini Datei konfiguriert ist.

Option | -ANF

gilt für 

spezifiziert die Fehlermeldung, wenn die Assembly nicht gefunden wird.

Standardeinstellung: The assembly "#requiredassembly#" could not be found.

Option | -PROBING:<Name>

gilt für 

gibt die Pfade an, unter denen die Assemblies zu finden sind.

Eingabe getrennt durch ';', oder Angabe des Namens einer `app.config` Datei.

Option | -SNK[F,N]:<Name>

gilt für



gibt den Strong Name-Schlüssel für die Assembly an und verwendet diesen zur Signierung der Assembly.

f signiert die Assembly mit dem Schlüsselpaar, das in der Datei <Name> definiert ist.

n signiert die Assembly mit dem Schlüsselpaar, das im Schlüssel-Container <Name> definiert ist.

Option

-TRAP[1:n]

gilt für



fügt Hacker-Fallen in die verschlüsselte Assembly ein.

fügt ungefähr n% Methoden in die verschlüsselte Assembly, die beim nachfolgenden Entschlüsseln den CmContainer sperren.

Die Standardeinstellung für n beträgt 10.

Option

-PRIO[0..31]<S>

gilt für



setzt die Prozesspriorität während des Image-Starts.

0 Der Wert 0 setzt keine Priorität. Der Wert 8 vergibt normale Priorität

S Gibt an, dass die Priorität nach dem Startvorgang nicht wieder hergestellt wird.

Option

-O[:Dateiname]

gilt für



setzt den Pfad und den Namen der verschlüsselten Zielfile.

7.7.5 Java-spezifische Einstellungen

Option

-ja: "params"

gilt für



gibt die Argumente an, die zur Laufzeit an die Main Class gegeben werden.

Option

-jb:<number>

gilt für



aktiviert oder deaktiviert ein besonderes Fehlerbehandlungsverfahren.



Kontaktieren Sie den Wibu-Systems Support für mehr Informationen.

Option

-jcl:<ClassLoader>

gilt für



gibt einen alternativen WIBU ClassLoader an.

Option `-jcl:<ClassLoader>`

Derzeit sind die folgenden ClassLoader verfügbar:

- ClassLoader ClassLoader abgeleitet von `java.net.URLClassLoader`
- DelegateClassLoader ClassLoader abgeleitet von `java.lang.ClassLoader`

Option `-jd:vmin[-vmax]`

gilt für



gibt die Minimal- (und Maximal-) Java-Version an, die benutzt wird.

Die Version muss dem Format entsprechen, wie in der Systemeigenschaft `java.version` angegeben. Die letzte Zahl kann ausgelassen werden.

-jd:1.4-1.5.0_04 lässt die Runtime Versionen von 1.4 bis Java 5 Update 0 Maintenance 4 zu.

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

gilt für



versteckt oder nennt verschlüsselte Klassen um.

- a benennt alle Klassen nach dem Muster '`<MyClass>.class.wibu`' um.
- e benennt nur die verschlüsselten Klassennamen nach dem Muster '`<MyClass>.class.wibu`' um.

Dies ist die Standardeinstellung.

- n benennt keine der verschlüsselten Klassen um.

Option `-jm:<Main-class>`

gilt für



gibt die startende Main Class an.

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

gilt für



aktiviert den nativen Klassen-Ladevorgang.

Dieses Vorgehen macht einen Eingriff in den Quellcode der Anwendung erforderlich.

- p verwendet JVMPI (Java Virtual Machine Profiler Interface).
- s verwendet das Java 6 Modul (wird noch nicht unterstützt).
- t verwendet JVMTI (Java Virtual Machine Tool Interface).

Option `-jl[w|b]:<list>`

gilt für 

gibt an welche Klassen verschlüsselt werden.

w Whitelist: alle Klassen, die dieser Liste entsprechen werden verschlüsselt.

b Blacklist: alle Klassen, die dieser Liste entsprechen werden nicht verschlüsselt.

<list> enthält komplett Klassen- oder Paket-Namen, oder Teile der Namen (Fragmente).
Die Trennung erfolgt mit ':'. Beispiel: '-jlw:com.wibu.:de.wibu.MainClass'

Option `-jo[[a:<jars>],[lsf],[e:[e]]:<list>`

gilt für 

gibt Ausgabeoptionen an.

a: <jars> fügt die angegebenen *.jar-Dateien der Ausgabe hinzu.

 -joa:CodeMeter.jar,WibuKey.jar fügt die Inhalte dieser *.jar-Dateien der Ausgabe hinzu, die mit -o angegeben wurde.

l: listet die Lizenzinformationen einer verschlüsselten *.jar-Datei auf.

s: teilt die Ausgabe in zwei verschiedene *.jar-Dateien.

Die WIBU Runtime Klassen und die Quell-*.jar-Dateien werden nicht zusammengeführt.

Diese Option wird für Servlets empfohlen, oder wenn Sie verschiedene *.jar-Dateien in einem Projekt verschlüsseln, um Platz zu sparen.

 Die erstellte WibuXpm4JRuntime.jar Datei muss dem Klassenpfad manuell hinzugefügt werden.

f: teilt die Ausgabe in drei verschiedene *.jar-Dateien.

Dies ist eine Erweiterung der Option -jos und erstellt eine Datei mit dem Namen WibuXpm4JO<outputfile>.jar.

e: [e] gibt an welche Dateien aus der Ausgabe ausgeschlossen werden.

[e]: bezeichnet die auszuschließende Datei, z.B. com/wibu/xpm/encrypted.

Option `-jvm:`

gilt für 

berücksichtigt die ausgewählte Option zur Virtual Machine.

Einige Java Laufzeiteinstellungen benötigen ein unterschiedliches internes Handling in AxProtector Java.

Die folgenden Option sind verfügbar:

<server>: Die Java Virtual Machine wird mit dem Server gestartet.

Option `-jx`

gilt für 

beendet die Anwendung durch Aufruf des System.exit() nachdem die 'Main-Class' Hauptmethode

Option -jx

einen Wert zurückgibt.

IxProtector Java mit AxProtector Version 9.0**Methodenverschlüsselung**

Mit Version 9.0 wird die Option eingeführt, mit *IxProtector* einzelne Methoden zu verschlüsseln. Dazu steht die Kommandozeilen-Option `-ci` zur Verfügung.

Bitte beachten Sie, dass sobald die Option `-ci` gesetzt ist, folgendes gilt:

- Die Optionen `-jn:t` und `-jh:n` sind standardmäßig aktiviert und dürfen nicht mehr angegeben werden.
- JVMPI wird nicht mehr unterstützt (nur noch JVMTI)
- Das Umbenennen von Klassen in (`.class.wibu`) ist nicht möglich
- Verschlüsseln einzelner Klassen (statt Jar-Files) ist nicht möglich
- Classloader (außer SystemClassLoader / ToolsSysCI) werden nicht unterstützt.

Über die zusätzliche Verwendung von Annotationen im Code kann weiterhin eingestellt werden, welche Klassen / Methoden verschlüsselt werden.

Es gelten die folgenden Einstellungen:

Annotation	Klasse	Methode
keine (Standardeinstellung)	Klasse ist nicht geschützt	Methode ist nicht geschützt.
<code>@Protected</code>	<p>Klasse ist geschützt (entspricht <code>@Protected(licenseList=0)</code>)</p> <p>Optionale Parameter: <u>licenseList</u></p> <ul style="list-style-type: none"> • <code>@Protected(licenseList=1)</code> verschlüsselt die Klasse mit Lizenzliste 1 (oder jedem anderen angegebenen Index-Eintrag 2, 3 usw.). <p><u>scope</u></p> <ul style="list-style-type: none"> • <code>@Protected(scope = {Class})</code> gibt an, dass nur diese Klasse verschlüsselt wird. • <code>@Protected(scope = {Method})</code> gibt an, dass die Verschlüsselung nur für die Methoden durchgeführt wird. Damit wird die Verschlüsselung aller Methoden mit einer Annotation erreicht). • <code>@Protected(scope = {Class, Method})</code> gibt an, dass die Verschlüsselung für die Klasse und alle Methoden 	<p>Methode ist geschützt (entspricht <code>@Protected(licenseList=0)</code>)</p> <p>Optionaler Parameter: <u>licenseList</u></p> <ul style="list-style-type: none"> • <code>@Protected(licenseList=1)</code> verschlüsselt die Klasse mit Lizenzliste 1 (oder jedem anderen angegebenen Index-Eintrag 2, 3 usw.).

Annotation	Klasse	Methode
	<p>außer Konstruktoren durchgeführt wird.</p> <p>Die scope- und licenseList-Optionen können miteinander kombiniert werden.</p>	
@Unprotected	Klasse ist nicht geschützt.	Methode ist nicht geschützt (Standardeinstellung, die aber mit dem Setzen der scope-Option für alle Methoden überschrieben werden kann).
@EntryPoint	Einsprungspunkt für alle Methoden	Einsprungspunkt für diese Methode. Eine Klasse darf nicht verschlüsselt werden.

Setzen von Parametern in einer XML-Datei

Zusätzlich zu den WBC-Dateien wird nun ebenfalls ein XML-Format unterstützt. Die Parameter für die automatische Verschlüsselung der ausführbaren Datei kann über die Option [-@xml](#)³¹⁹ eingebunden werden. Untenstehend eine Beispieldatei:

```
<?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>
            <FirmCode>10</FirmCode>
            <ProductCode>13</ProductCode>
            <FeatureCode>1</FeatureCode>
            <SubSystem>WanLocal</SubSystem>
        </License>
    </Wupi>
</AxProtectorJava>
```

```

        <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>
```

Erzeugen von maschinenlesbaren Class-Dateien

Im Fall des gesetzten Parameter `-ci` für die Methodenverschlüsselung (`/xProtector`) gibt es für externe Anwendungen, die Annotationen analysieren, z.B. Tomcat ab Version 7, die Option `-jff:[c|w]`, die es erlaubt das Verschlüsselungsergebnis entweder als verschlüsselte (unlesbare) Class-Dateien oder als wieder gültige Class-Dateien auszugeben.

Gültige Class-Dateien bestehen dann aus den Methodenrümpfen und Feldern mit Annotationen der ursprünglichen Klassen. Der verschlüsselte Bytecode wird hierbei in der Konstantensektion eingebettet.

Option `-jff:[c|w]`

gilt für 

definiert das Verschlüsselungsergebnis von Methoden.



Das Setzen des Parameters `-ci` für die Methodenverschlüsselung ist erforderlich.

w

erzeugt verschlüsselte (unlesbare) Class-Dateien (Standard-Einstellung).

c

erzeugt als Ausgabe wieder gültige Class-Dateien.

Angeben von zur Verschlüsselung benötigter Bibliotheken

Ab sofort werden alle in der zu verschlüsselnden Jar-Datei befindlichen extern referenzierten Jar-Dateien bei der Verschlüsselung benötigt. Das heißt, dass die im Manifest als Classpath angegebenen Klassen auch dort zur Verfügung stehen müssen. Die neue Option `-jcp` erlaubt das Bekanntgeben weiterer externer Bibliotheken.

Option -jcp:<Zusätzliche Jar-Dateien>

gilt für

gibt AxProtector externe Bibliotheken zusätzlich zu den im Classpath angegebenen bekannt.

-jcp: javaee-api-7.0.jar;someapi.jar

7.7.6 Bedienungseinstellungen

Option -!

gilt für

erzeugt eine Kommandodatei (*.wbc) .

Option -V

gilt für

setzt den ausführlichen Modus (verbose).

Im Fall von ist der Parameter -vn.

Option -#[Datei]

gilt für

druckt die Protokollierung in die angegebene Datei neben der automatischen Ausgabe in das AxProtector.log.[//Dokumente und Einstellungen\user].

Option -EXTRACT

gilt für

druckt den Inhalt der Assembly aus (Eingeben von -EXTRACT? Für mehr Informationen).

Option -? oder -h

gilt für

zeigt die Optionen im Kommandozeilen-Modus an.

Option -@cmds.wbc

gilt für

gibt eine *.wbc Datei an, die die Parameter für die automatische Verschlüsselung einer ausführbaren Datei enthält.

8 Individueller Softwareschutz mit *IxProtector*, WUPI und CodeMeter Kern-API

Zusätzlich zum automatischen Softwareschutz mit *AxProtector*, bei dem Sie nicht in den Quelltext der Anwendung eingreifen, die Sie schützen möchten, bietet *CodeMeter®* auch verschiedene Möglichkeiten, Softwareschutz individuell in Ihre Anwendung zu integrieren und die Sicherheit zu erhöhen.

IxProtector

Mit *IxProtector* steht Ihnen eine Schutztechnologie zur Verfügung, die es Ihnen während der Entwicklung Ihrer Software erlaubt, einzelne Bereiche im Quelltext zu definieren und zu schützen, die dann während der Laufzeit der Anwendung mit unterschiedlichen Lizenzinträgen verknüpft sind.

Softwareschutz-API WUPI

Die Schnittstelle, die Sie benötigen, um mit *IxProtector* geschützte Bereiche während der Laufzeit zu entschlüsseln, steht Ihnen in Form von WUPI (*WIBU Universal Protection Interface*) zur Verfügung. Dieses schlanke, nur wenige, aber elementare Funktionen umfassende Softwareschutz-API ist universell für viele Programmiersprachen einsetzbar.

Kern-API

Bestehen dann noch zusätzliche Anforderungen, wie etwa in den Bereichen Ver- und Entschlüsselung von Daten jeglicher Art, Personalisierung, oder dem Auslesen weiterer Daten, bietet Ihnen das [CodeMeter Kern-API](#)³³³ als die Schnittstelle, auf der alle anderen APIs und Schutzmechanismen aufbauen, umfangreiche Funktionen. Über den interaktiven [CodeMeter API Guide](#)³³⁷ erhalten Sie schnell den passenden Quelltext für die Integration in Ihre Software.

Wibu-Systems **empfiehlt** die kombinierte Verwendung von automatischem und individuellem Softwareschutz zur Erhöhung der Sicherheit.



Außerdem werden die Sicherheitsmechanismen von *AxProtector* wie *IxProtector* ständig weiterentwickelt und verbessert. Sie brauchen Ihre Anwendung nicht neu zu kompilieren, sondern lediglich neu mit *AxProtector* bzw. *IxProtector* zu verschlüsseln.

Leichte Kombination: automatischer und individueller Softwareschutz

Die Kombination aus automatischem und individuellem Softwareschutz gestaltet sich leicht. Zum einen ist *IxProtector* in *AxProtector* integriert, d.h. Sie können beide Schutztechnologien zusammen verwenden. Zum zweiten sind die Übergänge zwischen den einzelnen Schutzebenen über die Verwendung gleicher Zugriffe auf Lizenzinträge (Handles) gewährleistet. WUPI belegt beispielsweise den gleichen Lizenzintrag wie *AxProtector* und über einen Aufruf von [WupiGetHandle](#)³³² lesen Sie diesen Eintrag aus, um ihn im *CodeMeter Kern-API* weiterzuverwenden.

8.1 *IxProtector* und das Softwareschutz-API - WUPI

Die Schutztechnologie *IxProtector* ermöglicht es Ihnen, dass Sie während der Anwendungsentwicklung einzelne Bereiche (Module, Funktionen) im Quelltext definieren, die Sie danach verschlüsseln, und über indexbasierte Platzhalter mit Lizenzinträgen zur Laufzeit verknüpfen. Das *CodeMeter Softwareschutz-API WUPI* (*WIBU Universal Protection Interface*) steht Ihnen dazu als Schnittstelle zur Verfügung.

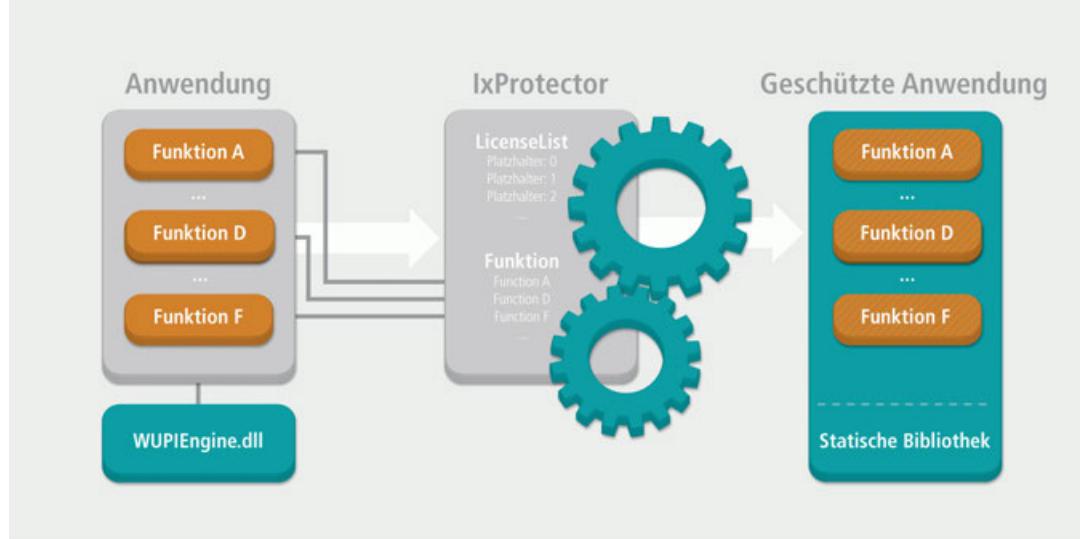


Abbildung 107: *IxProtector - Softwareschutz-API – WUPI*

Das Zusammenspiel von *IxProtector* und WUPI eignet sich für die folgenden Einsatzgebiete:

- Schützen und Freischalten einzelner Module in einer ausführbaren Datei, d.h. modularer Softwareschutz über definierte Funktions- und Lizenzlisten,
- Integrieren eigener Lizenzabfragen an frei definierbaren Stellen und damit zu ausgewählten Zeitpunkten,
- Verschlüsseln von Code-Fragmenten zur Erhöhung des Schutzgrades,
- Implementieren von Pay-per-Use Funktionalitäten, d.h. Herunterzählen eines Zählers bei einer bestimmten Aktion,
- Festlegen, wann welche Anti-Debug-Maßnahmen durch *AxProtector* durchgeführt werden,
- Gleichzeitiges Einsetzen für alle Lizenzierungssysteme (*WibuKey*, *CmDongle* und *CmActLicense*), wobei ein nachträgliches Ändern der Zuordnung stets möglich ist.
- Zugreifen auf die Lizenz, die von *AxProtector* belegt wurde zur Weiterverwendung im *CodeMeter Kern-API*,
- Lesen von Passwort-geschützten Daten während der Laufzeit der geschützten Anwendung, die zuvor in einem Hidden Data Feld abgelegt wurden.

Mit WUPI verwirklichen Sie:

- einfach durchzuführenden Schutz, der für viele Programmiersprachen verfügbar ist und ohne Neukompilierung einmal in die gleiche ausführbare Datei implementiert wird,
- ständig aktualisierten Schutz ohne Änderungen im einmal geschriebenen Quelltext durch kontinuierliche sicherheitstechnische Überarbeitung und Verbesserung der Funktionalitäten in aktualisierten *AxProtector*-Versionen.

8.2 WUPI Funktionen

Insgesamt stehen Ihnen die folgenden Funktionen des schlanken und effektiven *CodeMeter Software-schutz-APIs* WUPI zur Verfügung.



Mit der Ausnahme von ***WupiEncryptCode()*** und ***WupiDecryptCode()***, die sich auf Funktionslisten in *IxProtector* beziehen, haben alle anderen Funktionen einen Bezug zu Lizenzlisten.

Zugriffs-API: Belegen und Freigeben von Lizzenzen

WupiAllocateLicense()	Mit dieser Funktion kann eine Lizenz (<i>LicenseList</i>) auf dem gewählten Lizenzierungssystem belegt werden.
	Rückgabewert
	TRUE (1), wenn die Funktion erfolgreich durchgeführt wurde, ansonsten FALSE (0) wenn ein Fehler aufgetreten ist.
WupiFreeLicense()	Mit dieser Funktion wird eine Lizenz (<i>LicenseList</i>) auf dem gewählten Lizenzierungssystem freigegeben.
	Rückgabewert
	TRUE (1), wenn die Funktion erfolgreich durchgeführt wurde, ansonsten FALSE (0) wenn ein Fehler aufgetreten ist.
WupiGetHandle()	Diese Funktion gibt den aktuellen nativen Handle der Lizenz (<i>LicenseList</i>) zurück.
	Rückgabewert
	Es wird der aktuelle Nativ-Handle der Lizenz zurückgegeben. Falls ein Fehler aufgetreten ist, wird ein Wert von 0 zurückgegeben.

Ver- und Entschlüsselungs-API

WupiEncryptCode()	Mit dieser Funktion wird eine Funktion (<i>Function</i>) verschlüsselt.
	Rückgabewert
	TRUE (1), wenn die Funktion erfolgreich verschlüsselt wurde, ansonsten FALSE (0) wenn ein Fehler aufgetreten ist.
WupiDecryptCode()	Mit dieser Funktion wird eine Funktion (<i>Function</i>) entschlüsselt.
	Rückgabewert
	TRUE (1), wenn die Funktion erfolgreich entschlüsselt wurde, ansonsten FALSE (0) wenn ein Fehler aufgetreten ist.

Sicherheits-API

WupiCheckDebugger()	Mit dieser Funktion (<i>LicenseList</i> , <i>Level</i>) wird überprüft, ob die geschützte Anwendung im Kontext eines Debuggers läuft bzw. oder ein Debugger auf dem System läuft bzw. oder ob ein Kernel Debugger installiert ist.
	Rückgabewert
	TRUE (1), wenn die Funktion eine Debugger-Attacke gefunden hat, ansonsten FALSE (0).

Bitte beachten Sie, dass sich diese Sicherheitsfunktion nur für *AxProtector*-geschützte Anwendungen verwenden lässt; nicht aber für *IxProtector*-geschützte Anwendungen.

WupiCheckLicense()	Mit dieser Funktion wird eine Lizenz (LicenseList) auf dem gewählten Lizenzierungssystem auf eine sichere Art und Weise überprüft.
	Rückgabewert
	TRUE (1), wenn die Funktion erfolgreich durchgeführt wurde, ansonsten FALSE (0) wenn ein Fehler aufgetreten ist.
WupiDecreaseUnitCounter()	Mit dieser Funktion (LicenseList, Units) wird ein Unit Counter in der angegebenen Lizenz um die angegebene Anzahl von Einheiten heruntergezählt
	Rückgabewert
	TRUE (1), wenn der Unit Counter erfolgreich reduziert wurde, ansonsten FALSE (0).

Abfragen von Informationen

WupiQueryInfo()	Diese Funktion gibt Informationen zu einem Eintrag (LicenseList) bzw. zu einem CmContainer zurück.
	Rückgabewert
	Ist der gewünschte Wert vorhanden, wird dieser zurückgegeben. Wenn ein Fehler aufgetreten ist bzw. die gewünschte Information nicht vorhanden ist, wird -1 zurückgegeben und ein passender Fehler-Code ist gesetzt.

Lesen und Schreiben von Daten

Mit dem *CodeMeter Softwareschutz-API* WUPI besteht die Option, während der Laufzeit einer geschützten Anwendung zuvor auf dem *CmContainer* abgelegte Daten zu lesen, um z.B. hinterlegte Daten für die Programmfunctionalität zu nutzen. Zum Lesen dieser vorher einprogrammierten Daten stehen Ihnen die WUPI-Funktionen **WupiReadData** oder **WupiReadDataInteger** zur Verfügung. Zum Schreiben von Daten stehen Ihnen die WupiFunktionen **WupiWriteData** oder **WupiWriteDataInteger** zur Verfügung.

Bei *CodeMeter®* sind die eigentlichen Daten im Hidden Data-Feld gespeichert, die Sie z.B. über *CmBoxPgm* einprogrammiert haben.

Die Daten werden über indizierte Einträge (Typen) gespeichert. Insgesamt stehen dem Lizenzgeber 128 Hidden Data-Einträge zur Verfügung (0-127).

Die Standard-Eintragslänge beträgt 242 Bytes je Eintrag, die geringer ist als die maximale Eintragslänge von 256 Bytes. Die Verwendung dieser Standard-Eintragslänge optimiert die Verwendung der Hardware-Ressourcen im *CmContainer*. Das Lesen der Daten erfolgt automatisch über Einträge hinweg, d.h. wenn ein Eintrag mit der maximalen Eintragslänge gefüllt ist, wird automatisch beim nächsten Eintrag weitergelesen.

Bei 128 Hidden Data-Einträgen und der Standard-Eintragslänge können 30 . 976 Bytes und bei einer maximalen Eintragslänge 32 . 768 Bytes gelesen werden.

Damit Daten aus einem Hidden Data-Feld aus einem *CmContainer* gelesen werden können, ist das Setzen eines Zugangs-Codes erforderlich, des Hidden Data Access Codes (HDAC). Dieser HDAC kann einem automatisch erstelltem Ableitungswert entsprechen. Dieser errechnete Ableitungswert setzt sich aus verschiedenen Parametern zusammen, wie z.B. Firm Code, Product Code, etc.



Wibu-Systems empfiehlt die Verwendung dieses Ableitungswertes.

Wenn die Daten mit dem *Programmier-API (HIP)* in den *CmContainer* geschrieben werden, greifen spezielle Speichermethoden. In diesem Fall können zur Verschlüsselung die Standard-Einstellungen des

AxProtector verwendet werden.



Wenn die Daten nicht mit dem *Programmier-API (HIP)* in den *CmContainer* geschrieben werden, können Sie nicht den automatischen Ableitungswert als HDAC verwenden. Sie müssen dann die notwendigen AxProtector-Einstellungen manuell über die *.wbc- Datei setzen.

Der Lizenzdefinitionsbereich der *.wbc-Datei hat dann folgendes Aussehen:

```
[License CM1]
Type=CodeMeter

UserData=read ; ← erforderlich, aktiviert den Daten-Lesemodus
FirstHiddenData=13 ; ← optional, Standardwert entspricht 0
HiddenDataAccessCode=42 ; ← optional, Standardwert entspricht dem Ableitungswert als HDAC
DataBlockSize=240 ; ← optional, Standardwert entspricht 242
```

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

Diese Funktion liest Rohdaten aus dem *CmContainer* wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.

Die Funktion kann für alle Programmiersprachen verwendet werden, die mit Pointern arbeiten, d.h. mit speziellen Variablen, die Speicheradressen enthalten.

Für die anderen Programmiersprachen wird die Funktion [WupiReadDataInteger](#)³²⁴ angeboten.

Parameter	Beschreibung
iLicenseList	bezeichnet die Nummer des Lizenzlisten-Indexes.
iOffset	enthält in Anzahl von Bytes den Datenversatz vom Beginn des Datenblocks.
pvData	enthält das zu füllende Daten-Array.
cbData	enthält die Anzahl von Bytes von cbData.

Rückgabewert

Der Rückgabewert besteht in der Anzahl der Bytes, die in *pvData* gespeichert sind.

Hat der Rückgabewert einen Wert von 0 rufen Sie [WupiGetLastError](#)³²⁵ auf, um detailliertere Informationen zu bekommen.

WupiReadDataInteger(int iLicenseList, int iOffset);

Diese Funktion liest Rohdaten aus dem *CmContainer*, wenn diese Daten vorher an einer festgelegten Stelle gespeichert wurden.

Die Daten werden 2Bytes-weise gelesen.

Die Funktion kann für alle Programmiersprachen verwendet werden.

Für andere Programmiersprachen, die mit Pointern arbeiten, d.h. mit speziellen Variablen, die Speicheradressen enthalten, empfiehlt Wibu-Systems die Funktion [WupiReadData](#)³²⁴.

Parameter	Beschreibung
iLicenseList	bezeichnet die Nummer des Lizenzlisten-Indexes.
iOffset	enthält den Datenversatz vom Beginn des Datenblocks in Anzahl von Bytes.

Rückgabewert

Der Rückgabewert hat eine Größe von 4 Bytes. Er ist aufgeteilt in 2 obere Bytes, die Status-Flags für die Fehler- und Nachrichten-Behandlung enthalten, und 2 untere Bytes, in denen sich die Daten befinden.

WupiReadDataInteger(int iLicenseList, int iOffset);

Die oberen 2 Bytes können z.B. die folgenden Werte enthalten: <code>#define WupiRDError (0x80000000)</code> <code>#define WupiRDMoreDataAvail (0x40000000)</code>

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

Diese Funktion schreibt Rohdaten in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde. Die Funktion kann für alle Programmiersprachen verwendet werden, die mit Pointern arbeiten, d.h. mit speziellen Variablen, die Speicheradressen enthalten. Für die anderen Programmiersprachen wird die Funktion WupiWriteDataInteger ³²⁵ angeboten.
--

Parameter	Beschreibung
iLicenseList	bezeichnet die Nummer des Lizenzlisten-Indexes.
iOffset	enthält in Anzahl von Bytes den Datenversatz vom Beginn des Datenblocks.
pvData	enthält das zu füllende Daten-Array.
cbData	enthält die Anzahl von Bytes von cbData.

Rückgabewert

Die Funktion gibt FALSE (0) zurück, wenn ein Fehler auftritt; ansonsten TRUE (1). Hat der Rückgabewert einen Wert von 0 rufen Sie WupiGetLastError ³²⁵ auf, um detailliertere Informationen zu bekommen.
--

WupiWriteDataInteger(int iLicenseList, int iOffset, int iData);

Diese Funktion schreibt Rohdaten in einen <i>CmContainer</i> , wenn dieser vorher zum Beschreiben vorbereitet wurde. Die Funktion kann für alle Programmiersprachen verwendet werden. Für andere Programmiersprachen, die mit Pointern arbeiten, d.h. mit speziellen Variablen, die Speicheradressen enthalten, <u>empfiehlt</u> Wibu-Systems die Funktion WupiWriteData ³²⁵ .

Parameter	Beschreibung
iLicenseList	bezeichnet die Nummer des Lizenzlisten-Indexes.
iOffset	enthält den Datenversatz vom Beginn des Datenblocks in Anzahl von Bytes.
ini iData	enthält die zu schreibenden Daten.

Rückgabewert

Die Funktion gibt FALSE (0) zurück, wenn ein Fehler auftritt; ansonsten TRUE (1). Hat der Rückgabewert einen Wert von 0 rufen Sie WupiGetLastError ³²⁵ auf, um detailliertere Informationen zu bekommen.
--

Fehler API

WupiGetLastError()

Mit dieser Funktion wird der aktuell gesetzte Fehler-Code des aktuell gesetzten Lizenztyps (LicenseList) zurückgegeben.

Rückgabewert

<code>wibu::UpiError.NoError (0) --> Es ist kein Fehler aufgetreten.</code> <code>wibu::UpiErrorNoDefaultLicense (-1)</code> <code>--> Es ist keine Default-Lizenz gesetzt, d.h. die Anwendung ist nicht noch zusätzlich automatisch verschlüsselt.</code> <code>wibu::UpiErrorLicenseNotFound (-2)</code>

WupiGetLastError()

```
--> Der angegebene Index auf eine Lizenz konnte nicht gefunden werden.
wibu::UpiErrorFunctionNotFound (-3)
--> Der angegebene Index auf eine Funktion konnte nicht gefunden werden.
wibu::UpiErrorRuntimeTooOld (-4)
--> Die installierten Treiber des benutzten Lizenzierungssystems sind zu alt.
wibu::UpiErrorDebuggerDetected (-5)
--> Ein Debugger-Attacke wurde erkannt.
```

Tabelle 6: WUPI-Funktionen – Überblick

8.2.1 Individueller Softwareschutz mit WUPI: ein Beispiel Indexbasierte Platzhalter

Indexbasierte Platzhalter

Über das *CodeMeter Softwareschutz-API* WUPI werden im Programmablauf Ihrer Anwendung über indexbasierte Platzhalter Software-Schutzmechanismen oder Lizenzabfragen mit Teilen des Quelltextes verknüpft. Im Folgenden zeigen Auszüge aus der Beispielanwendung "Second Sample" wie modularer Softwareschutz über WUPI implementiert wird.



Sie finden das ausführliche Beispiel nach Installation des *CodeMeter® SDK* für die jeweils aktuellen Programmiersprachen im Verzeichnis "%\Users\Public\Documents\WIBU-SYSTEMS\Software Protection".

Alternativ finden Sie die Beispiele über den Navigationseintrag "**Start | Alle Programme | CodeMeter | Samples**" oder über [CodeMeter Start Center](#)⁶⁷.

Die Ausgangsüberlegung des Beispiels ist, Anwendern eine kopiergeschützte Anwendung zu überlassen, zu deren Nutzung passende Einträge in *CmContainern* benötigt werden. Für die unterschiedlichen Module benötigt der Anwender passende zusätzliche Einträge.

Die Umsetzung erfolgt in fünf Schritten:

1. [Definition der Module](#)³²⁶,
2. [Erstellen von indexbasierten Lizenz- und Funktionslisten](#)³²⁷,
3. [Programmieren der Lizenzeinträge](#)³³⁰,
4. [Einfügen in den Quelltext](#)³³¹,
5. [Verschlüsseln der Anwendung](#)³³³.

8.2.1.1 Definition der Module

Der Funktionsumfang eines einfachen Texteditor des "Second Sample"-Beispiels ist klar modular aufgebaut. Neben der "Speichern"-Funktion als Teil der allgemeinen Lizenz gibt es die Funktion "Change Font", für die eine separate Lizenz erforderlich ist.

8.2.1.2 Platzhalter in IxProtector Lizenz- und Funktionslisten

Die Informationen der untenstehenden Tabelle reichen aus, um für die spätere Verknüpfung zwischen IxProtector und dem Aufruf von WUPI-Funktionen im Quelltext die numerischen Platzhalter anzulegen.

Modul	Firm Code, Product Code, Feature Code	Funktionsname
Basislizenz	10:201000:1	Save
Font-Änderung	10:201001:1	ChangeFont

Tabelle 7: Second Sample – Übersicht

Zur Anlage der Platzhalter gehen Sie wie folgt vor:

i Die AxProtector-Projektdateien finden Sie für die jeweils aktuellen Programmiersprachen ebenfalls im Verzeichnis "%\Users\Public\Documents\WIBU-SYSTEMS\Software Protection"

- Aktivieren Sie zunächst IxProtector in AxProtector über die "Erweiterte Optionen"-Seite.

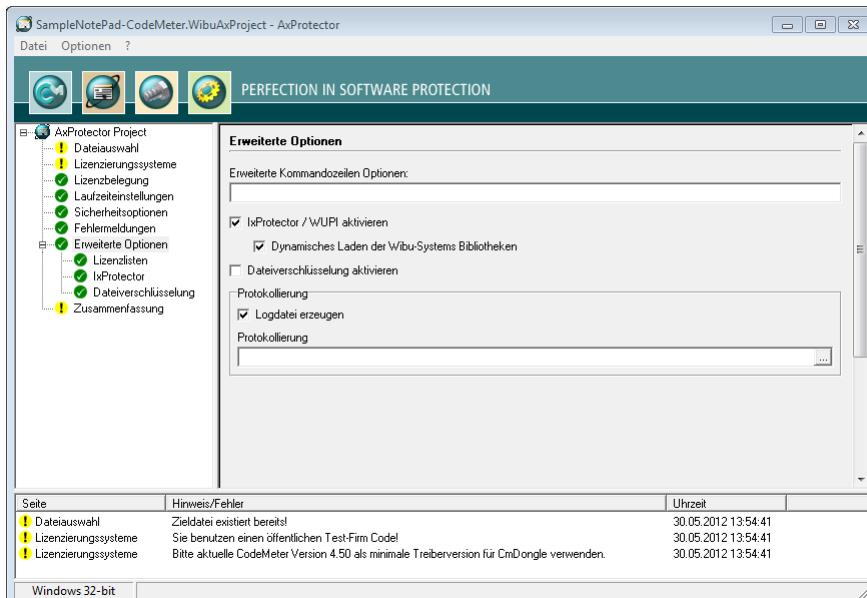


Abbildung 164: IxProtector innerhalb von AxProtector aktivieren

i Mit dieser Option sucht IxProtector die betreffenden Quelltext-Bereiche heraus und verschlüsselt diese bevor AxProtector einen Schutzumschlag um die gesamte Anwendung legt.
Wenn Sie IxProtector ohne AxProtector verwenden möchten, so wählen Sie den Projekttyp "[Nur IxProtector](#)"²¹⁸.
Wibu-Systems empfiehlt IxProtector innerhalb von AxProtector zu nutzen, falls keine besonderen Gründe dagegen sprechen.

- Navigieren Sie auf die "Lizenzlisten"-Seite.

Lizenzlisten

Liste der Lizenzlisten:

ID	Beschreibung	Elemente	Element Details
0	(Standardlizenz)	{1 Element}	{CmDongle 10 201000 1 Lokal - Netzwerk Normal user limit 4.30 1.18 0 none };
1	Font	{1 Element}	{CmDongle 10 201001 1 Lokal - Netzwerk Normal user limit 4.30 1.18 0 none };

Hinzufügen Bearbeiten Löschen

Hilfe < Zurück Weiter >

Abbildung 165: *lxProtector* - Lizenzliste

Lizenzlisten erlauben Ihnen Lizenzen mit unterschiedlichen Lizenzelementen (Lizenzierungssystem, Firm Code, Product Code etc.) zu einzelnen Einträgen zusammenzufassen. Ein Eintrag kann dabei mehrere Lizenzelemente umfassen.



Die Einträge in den Lizenzlisten können alle Wibu-Systems Lizenzierungssysteme (*WibuKey*, *CmDongle* und *CmActLicense*) enthalten. Eine Zuordnung zu einzelnen Lizenzierungssystem ist nachträglich ohne Änderung des Quelltextes änderbar. Lediglich die geänderte Lizenzinformation muss in die Verschlüsselung einfließen.

Der Lizenzlisten-Eintrag von 0 beschreibt hier die Lizenz, auf die sich *AxProtector* bezieht.

3. Markieren Sie den Lizenzlisten-Eintrag mit der ID 1 und klicken Sie auf die "Bearbeiten"-Schaltfläche.

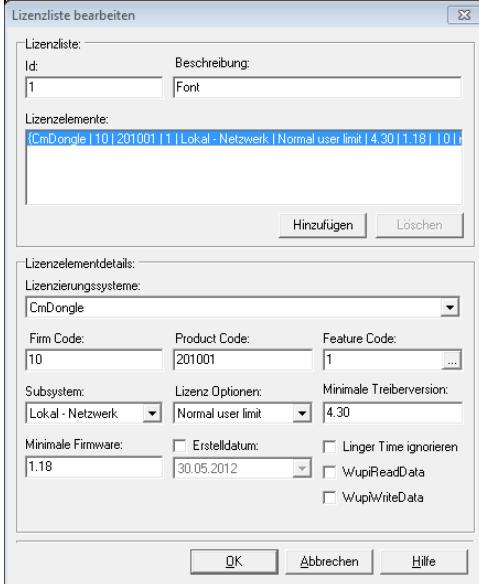


Abbildung 166: *IxProtector* – Lizenzlisten-Eintrag

Für das "Second Sample"-Beispiel wird Ihnen hier für die Change Font-Lizenz der numerischen Platzhalter (die ID=1) angeboten. Die notwendigen Daten entnehmen Sie der [Übersichtstabelle](#)³²⁷, d.h. Firm Code 10 und Product Code 201001 mit einem Bit-Wert (Feature Code) der Feature Map von 1.

In der "ID"-Spalte steht nun der numerische Platzhalter, der über WUPI Lizenz-Befehle angesprochen wird.

4. Navigieren Sie auf die "**IxProtector**"-Seite, um sich die Funktionsliste anzeigen zu lassen. Die hier angebotenen *IxProtector* Optionen legen die Funktionen fest, die geschützt werden sollen und bekommen einer der oben definierten Lizenzlisten-Einträge zu gewiesen.
5. Klicken Sie auf die "**Bearbeiten**"-Schaltfläche.

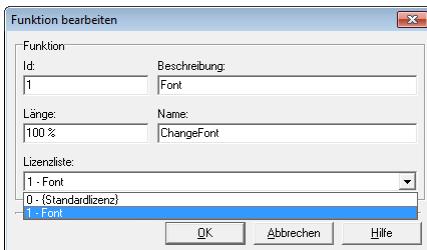


Abbildung 167: *IxProtector* – Funktionslisten-Eintrag

Für das "Second Sample"-Beispiel wird Ihnen hier für die ChangeFont-Funktion der numerischen Platzhalter (die ID=1) angeboten. Übertragen Sie die den Funktionsnamen aus der obigen [Übersichtstabelle](#)³²⁷.



Wichtig ist, dass die Bezeichnung der Funktion im "**Name**"-Feld exakt dem Namen entspricht, der später im Quelltext über diesen indexbasierten Platzhalter angesprochen wird. Überladene Funktionen sind nicht möglich.

Geben Sie außerdem noch die Länge des zu verschlüsselnden Bereichs der Funktion an. Die Länge kann in Prozent (0 . . . 100%) angegeben werden. Hierzu geben Sie das Prozentzeichen mit an. Alternativ dazu ist auch die Angabe in Bytes möglich.



Geben Sie kein Prozentzeichen hinter der Zahl an, so wird diese als Anzahl in Bytes interpretiert

Schließlich wählen Sie die Lizenzliste aus, zu der die Funktion zugeordnet werden soll.

Damit sind alle notwendigen Einträge in *IxProtector* vorgenommen und alle numerischen Platzhalter gesetzt.

8.2.1.3 Programmierung des CmContainer

Nach Schützen der Anwendung "Second Sample" mit *IxProtector* in der *AxProtector*-Oberfläche müssen Sie nun die Lizenzinträge in den *CmContainer* übertragen. Dazu nutzen Sie entweder [CodeMeter License Editor](#)³⁴⁶, [CmBoxPgm](#)³⁵⁵ oder [CodeMeter License Central](#)³⁸⁴.

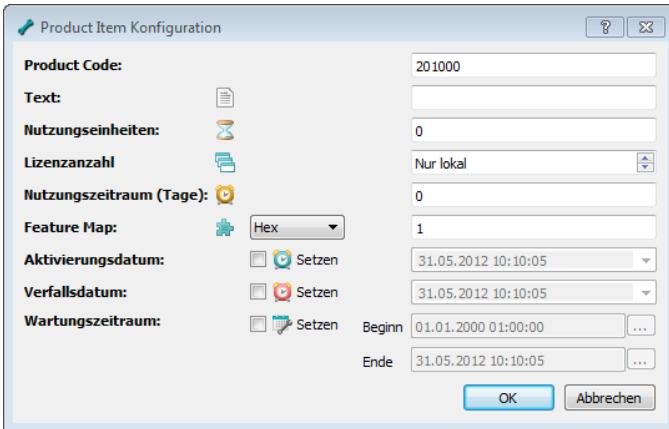
Zu programmieren ist:

- ein Product Item mit dem Product Code 201000 für den Lizenzcontainer mit dem Test-Firm Code 10 und einem Feature Code-Wert von 1 für die Feature Map.
- ein Product Item mit dem Product Code 201001 für den Lizenzcontainer mit dem Test-Firm Code 10 und einem Feature Code-Wert von 1 für die Feature Map.

Programmieren mit *CodeMeter License Editor*

In *CodeMeter License Editor* führen Sie dazu die folgenden Schritte durch:

- Markieren der Lizenzcontainer-Ebene des Test-Firm Codes 10 und Anlegen des Product Items mit dem Product Code 201000 bzw. 201001 über "**Hinzufügen**"-Element entweder über die entsprechende Schaltfläche, oder das Kontext-Menü.



2. Ausfüllen der **Product Code**, **Text**, **Unit Counter** und **Feature Map** Felder nach den Vorgaben.
3. Betätigen der "Ausführen"-Schaltfläche, um diesen Lizenzeintrag in den angeschlossenen CmDongle zu programmieren.

Programmieren mit CmBoxPgm

In CmBoxPgm führen Sie dazu die folgenden Programmierschritte durch:

1. Anlegen eines Product Items mit dem Product Code 201000 im Lizenzcontainer mit dem Test-Firm Code 10 und einem Feature Code-Wert von 1 für die Feature Map.
`CmBoxPgm.exe /f10 /p201000 /pfm1 /ca`
2. Anlegen eines Product Items mit dem Product Code 201001 im Lizenzcontainer mit dem Test-Firm Code 10 und einem Feature Code-Wert von 1 für die Feature Map.
`CmBoxPgm.exe /f10 /p201001 /pfm1 /ca`

8.2.1.4 Einfügen in den Quelltext

Anschließend fügen Sie die WUPI Funktionen dort in den Quelltext ein, wo die Software-Schutzmechanismen oder Lizenzabfragen durchgeführt werden sollen.

Über die vorher in IxProtector angelegten Lizenz- und Funktionslisten beziehen sich die WUPI Funktionen nun auf die dort angelegten Platzhalter.

Während der Entwicklung müssen Sie zunächst eine Dummy-DLL integrieren, die die WUPI Funktionsaufrufe enthält.

Im Fall von Windows ist dies die je nach Betriebssystem (32- oder 64-Bit) die `wupiEngine32.dll` bzw. `wupiEngine64.dll`, im Fall von .NET Anwendungen die `wupiEngineNet.dll`.

Diese Dateien befinden sich im Verzeichnis "`%Program Files%\WIBU-SYSTEMS\AxProtector\DevKit\lib`".

Im Folgenden zeigen Quelltext-Ausschnitte wie einige WUPI Funktionen für das "Second Sample"-Beispiel umgesetzt wurden.



Die Quelltextdatei für "Second Sample" in der Programmiersprache C++ (aber auch die der anderen Sprachen) finden Sie im Verzeichnis "%\Users\Public\Documents\WIBU-SYSTEMS\Software Protection". Alle folgenden Beispelauszüge stammen aus dieser Implementierungsdateien.

WupiQueryInfo

In der Datei SampleNotePad.cpp des Second Sample (C++) wird ***WupiQueryInfo*** z.B. bei Start der Anwendung durchgeführt.

```
CTextEditApp::CTextEditApp()
{
    // Construction code, initialization in InitInstance
    // Checks if the software is encrypted
    if (WupiQueryInfo(0, WupiQIFirmCode) == 0)
    {
        MessageBox(NULL, TEXT("Software is not encrypted correctly! \nDon't ship this version."),
                  TEXT("SampleNotePad - INTERNAL version"), MB_ICONERROR);
    }
}

//CTextEditApp()
```

WupiEncryptCode* und *WupiDecryptCode

In der Datei CMainFrame.cpp des Second Sample (C++) werden ***WupiDecryptCode*** und ***WupiEncryptCode*** beim Aufruf von OnViewFont() aufgerufen.

```
/// <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);
    }
}

//OnViewFont()
```

8.2.1.5 Verschlüsselung mit AxProtector

Nach dem Kompilieren von "Second Sample" verschlüsseln Sie mit *AxProtector* und aktivieren dabei *IxProtector*. *IxProtector* ersetzt nun die Platzhalter durch die Einträge in der Lizenzliste oder Funktionsliste.



IxProtector ist in *AxProtector* integriert und kann alternativ über einen "IxProtector" Projekttyp allein, oder zusätzlich zu *AxProtector* genutzt werden. Bei der Integration von *IxProtector* in *AxProtector* sucht *IxProtector* die betreffenden Quelltext-Bereiche heraus und verschlüsselt sie bevor *AxProtector* die gesamte Anwendung verschlüsselt. Aber selbst im Fall eines "IxProtector" Projekttyps ist eine erhöhte Sicherheit des Schutzes gegeben, da die Dummy-DLL bei der Verwendung von *IxProtector* durch statischen Code ersetzt wird. Diese DLL wird später bei der Ausführung der Anwendung nicht mehr benötigt.

8.3 Das CodeMeter Kern-API

Mit dem *CodeMeter Kern-API* bietet Wibu-Systems eine mächtige Schnittstelle zur Kommunikation mit *CmContainern* zur Laufzeit von *CodeMeter Lizenzserver*. Alle anderen APIs und Schutzmechanismen (*AxProtector*, *IxProtector*, *Softwareschutz-API WUPI*) setzen letzten Endes auf *Kern-API* Funktionen auf. Daher eignet sich diese Schnittstelle zum Einsatz ergänzend zu den anderen Schutzmöglichkeiten durch *AxProtector* und *IxProtector*. Die Übergänge sind dabei recht einfach.

Ein Eintrag - verschiedene Schutzschichten

Der Eintrag, den das *Softwareschutz-API WUPI* zur Laufzeit verwendet, ist über *AxProtector* belegt. Mit der WUPI Funktion [*WupiGetHandle*](#)²² innerhalb von *IxProtector* lesen Sie diesen Eintrag aus, und verwenden ihn weiter im *CodeMeter Kern-API*.

Einsatz-Szenarien

Mögliche zusätzliche Einsatzszenarien umfassen:

- Auslesen weiterer Daten aus dem *CmContainer*, z.B. Anzeigen und Mitschicken von benutzerspezifischen Lizenzinformation (COLI) beim Auslösen einer Support-Anfrage (***CmGetInfo*** über ***WupiGetHandle***).
- Ver- und Entschlüsselung von Daten jeder Art innerhalb von Anwendungen, z.B. Verschlüsseln über ***CmCrypt*** oder ***CmCrypt2*** mit unterschiedlichen Sicherheitsfeatures (Encryption Code Options) für veränderliche Daten innerhalb einer Anwendung, d.h. sensible Daten werden dann bei verschiedenen Kunden unterschiedlich verschlüsselt.
- Verwendung eines *CmDongles* zur Authentifizierung, z.B. Signieren von Daten und damit Führen des Nachweises, dass von verschiedenen Benutzern übermittelte Daten auch tatsächlich von diesen Benutzern geschickt wurden.
- Aktualisieren von Lizenzinformationen über Erstellung von Kontext-Dateien (***CmSetRemoteContext***) und deren Aktualisierung durch Update-Dateien (***CmSetRemoteUpdate***) um z.B. pay-per-use-Informationen zu erhalten.

Das sind nur einige der zusätzlichen Möglichkeiten, die das *CodeMeter Kern-API* bietet. Bei Fragen steht Ihnen der Wibu-Systems Kundensupport jederzeit zur Verfügung.

8.3.1 Funktionsbereiche

Die Funktionen des *CodeMeter Kern-API* lassen sich in verschiedene Bereiche zusammenfassen. Der überwiegende Teil der Funktionen findet sich im [CodeMeter API Guide](#)³³⁷ wieder. Die Funktionen werden hier nur kurz skizziert. Für eine detailliertere Beschreibung der Funktionen, zu Syntax und Parameter siehe *CodeMeter Core API Help* (erreichbar als Kontexthilfe F1 im *CodeMeter API Guide*, oder über den "Start | Alle Programme | CodeMeter | Documentation" Systemmenü-Eintrag).

8.3.1.1 Zugriffs API

Dieser Bereich beinhaltet alle Funktionen, die Zugriffe auf einen *CmContainer* ermöglichen.

Befehl	Beschreibung
CmAccess	führt einen Zugriff auf ein Subsystem, einen <i>CmContainer</i> , ein Firm Item, oder auf einen Produkteintrag (Product Item) in einem gegebenen Subsystem durch.
CmAccess2	führt einen Zugriff wie CmAccess durch, aber stellt erweiterte Funktionen zur Verfügung (verfügbar seit <i>CodeMeter® Version 3.30</i>). <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Verwenden Sie CmAccess2 um einen erweiterten Funktionsumfang zu nutzen. </div>
CmRelease	schließt ein mit CmAccess oder CmAccess2 geöffnetes Handle. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Verwenden Sie die CmRelease auf Einträge, die Sie über WUPI mit dem Befehl WupiGetHandle adressiert haben. </div>

8.3.1.2 Authentifizierungs API

Dieser Bereich beinhaltet alle Funktionen, die für die Durchführung von Authentifizierungsvorgängen benötigt werden.

Befehl	Beschreibung
CmCalculateDigest	berechnet einen 32 Byte Hash-Wert einer eingegebenen Eingabefolge zur Verwendung in einer Authentifizierung. Es wird der Algorithmus SHA-256 eingesetzt.
CmCalculateSignature	berechnet eine ECDSA (Elliptic Curve Digital Signature Algorithm) Signatur mit dem angegebenen Hash-Wert im <i>CmContainer</i> .
CmGetPublicKey	liest den öffentlichen Schlüssel aus einem <i>CmContainer</i> .
CmValidateSignature	überprüft eine ECDSA (Elliptic Curve Digital Signature Algorithm) Signatur mit einem gegebenen öffentlichen Schlüssel.

8.3.1.3 Enabling API

Dieser Bereich beinhaltet Funktionen, die für das [Enabling](#)⁴⁰⁶ (geregelte Aktivieren und Deaktivieren eines kompletten *CmContainers*, Firm Item- oder Lizenzeinträgen) benötigt werden.

Elementare Enabling-Funktionen können einfach durch **CmGetInfo()** und **CmProgram()** umgesetzt werden. Ein Einsatz der nachfolgenden Funktionen ist daher nur in seltenen Fällen notwendig.

Befehl	Beschreibung
CmEnablingWriteApplicationKey	führt Operationen für den Enabling Access Code durch.
CmEnablingGetApplicationContext	liest die Inhalte der Anwendung aus, die aktiviert oder deaktiviert werden soll.
CmEnablingGetChallenge	liest die Session ID aus dem <i>CmContainer</i> , der aktiviert oder deaktiviert werden soll.
CmEnablingSendResponse	aktiviert einen <i>CmContainer</i> oder einen <i>CmContainer</i> -Eintrag.
CmEnablingWithdrawAccessRights	deaktiviert einen <i>CmContainer</i> oder einen <i>CmContainer</i> Eintrag.

8.3.1.4 Verschlüsselungs API

Dieser Bereich beinhaltet alle Funktionen, die für Vorgänge der Ver- und Entschlüsselung von Daten benötigt werden.

Befehl	Beschreibung
CmCrypt, CmCrypt2	ver- bzw. entschlüsselt Daten direkt oder indirekt über einen <i>CmContainer</i> .
CmCryptEcies	verschlüsselt eine gegebene Byte-Folge mit dem ECIES (Elliptic Curve Integrated Encryption Scheme) Algorithmus.
CmCrypt Sim	ver- bzw. entschlüsselt Daten direkt oder indirekt über den Firm Security Box Eintrag des gewünschten Firm Codes.
CmCalucalatePioCoreKey	berechnet den Core-Key zum Entschlüsseln der PIO Hidden Data. Diese Operation benötigt eine Firm Security Box.
CmGetSecureData	liest mit dem Product Item Option Encryption Key (PIOEK) verschlüsselte Secure Data aus dem <i>CmContainer</i> aus.
CmDecryptPioData	entschlüsselt eine ausgelesene Hidden Data Folge mit dem Product Item Decrypt- on Key (PIODK).
CmGetPioDataKey	berechnet den Schlüssel, der zum Entschlüsseln von Hidden Data benötigt wird.

8.3.1.5 Fehlermanagement API

Dieser Bereich beinhaltet Funktionen, die für die Arbeit mit Fehlermeldungen benötigt werden.

Befehl	Beschreibung
CmConvertString	wandelt die Eingabe in eine angegebene Buchstabenfolge um.
CmGetLastErrorCode	fragt den letzten Fehlercode ab.
CmGetLastErrorText	fragt den letzten Fehler-Text ab.
CmGetLastErrorText2	umfasst erweiterte Funktionen im Vergleich zu CmGetLastErrorText .
CmSetLastErrorText	setzt einen Fehlercode in eine intern verwendete globale Fehlercode-Variable.

8.3.1.6 Management API

Dieser Bereich beinhaltet alle Funktionen, mit denen die für Ver- und Entschlüsselungsvorgänge von Daten benötigt werden.

Befehl	Beschreibung
CmCheckEvents	Wartet bis ein ausgewähltes (lokales) Ereignis eintritt und gibt dieses Ergebnis zurück.
CmGetBoxes	ermittelt alle verbundenen <i>CmContainer</i> am angegebenen Anschluss.
CmGetBoxContents	liest alle Einträge eines <i>CmContainers</i> .
CmGetInfo	fragt Daten aus dem <i>CmContainer</i> ab. Unterschiedlich verwendete Anfrage-Parameter ergeben verschiedene Ergebnisse.
CmGetServers	durchsucht das lokale Netzwerk nach laufenden <i>CodeMeter License Server</i> , bei denen ein <i>CmContainer</i> angeschlossen ist.
CmGetVersion	Ermittelt die Version des betreffenden <i>CodeMeter® Modules</i> .

8.3.1.7 Programming API

Dieser Bereich beinhaltet Funktionen, mit denen *CmContainer* programmiert werden können.

 i	Die Funktionen dieses Bereiches sind mittlerweile auf Herstellerseite vom <i>Programmier-API [High Level Application Programming Interface (HIP)]</i> abgelöst worden. Ein Einsatz dieser Funktionen ist daher nur in seltenen Fällen notwendig.
--	--

Befehl	Beschreibung
CmReserveFirmItem	reserviert ein temporäres Firm Item in einem <i>CmContainer</i> für nachfolgende Firm Item und Product Item Operationen.
CmCreateProductItemOption	bereitet eine Sicherheitssequenz für das Hinzufügen oder die Aktualisierung einer Product Item Option vor.
CmCreateSequence	berechnet eine Signatur um einen <i>CmContainer</i> Eintrag zu programmieren.
CmProgram	programmiert verschiedene Einträge in einen <i>CmContainer</i> .
CmValidateEntry	überprüft eine angegebene Sequenz.

8.3.1.8 Remote Update API

Dieser Bereich beinhaltet alle Funktionen, die für die Fernprogrammierung von Lizenzanforderungs- und Lizenzaktualisierungsdateien benötigt werden

(* .WibuCmRac und *.WibuCmRaU-Dateien).

Befehl	Beschreibung
CmGetRemoteContext	speichert die Inhalte eines <i>CmContainers</i> in eine verschlüsselte und komprimierte Remote Kontext-Datei (Lizenzanforderungsdatei) (*.WibuCmRac-Datei).
CmSetRemoteContext2	umfasst erweiterte Funktionen im Vergleich zu CmGetRemoteContext .
CmSetRemoteUpdate	programmiert einen <i>CmContainer</i> mit einer angegebenen Remote Activation Update-Datei (Lizenzaktualisierungsdatei) (*.WibuCmRaU-Datei). Die Datei

Befehl	Beschreibung
	enthält alle notwendigen Informationen um den <i>CmContainer</i> zu programmieren.
<i>CmSetRemoteUpdate2</i>	umfasst erweiterte Funktionen im Vergleich zu <i>CmSetRemoteUpdate</i> .
<i>CmListRemoteUpdate</i>	analysiert eine Remote Activation Update-Datei (Lizenzaktualisierungsdatei) (*.WibuCmRaU-Datei) und bestimmt die Seriennummern aller in der Datei referenzierten <i>CmContainer</i> .
<i>CmListRemoteUpdate2</i>	umfasst erweiterte Funktionen im Vergleich zu <i>CmListRemoteUpdate</i> .

 Die erweiterten Funktionen mit dem Suffix 2 ermöglichen beispielsweise die Verwendung eines Puffers anstelle einer Datei, oder Encoding-Einstellungen für übergebene Dateinamen.

8.3.1.9 Zeit Management API

Dieser Bereich beinhaltet die Funktion, die für die Verwendung einer aktuell zertifizierten Zeit benötigt wird (zur Synchronisierung der verschiedenen Zeiten auf einem *CmContainer* siehe [hier](#)⁴¹⁷).

Befehl	Beschreibung
<i>CmSetCertifiedTimeUpdate</i>	bezieht vom Zeitserver (Certified Time Creation Server, CTCS) eine aktuell zertifizierte Zeit und speichert diese im <i>CmContainer</i> .

8.3.2 CodeMeter API Guide

CodeMeter API Guide ist ein interaktives Programm zur Erstellung von Quelltext-Fragmenten. Sie erstellen und testen API Funktionen mit allen dazugehörigen Parametern und notwendigen Strukturen für Ihre Programmiersprache. Unterstützte Programmiersprachen umfassen derzeit C, C++, C#, CB6, VB.NET, Delphi, und Java.

Die erstellten Quellcode-Fragmente können Sie einfach über die Zwischenablage in den Quellcode einer Anwendung übernehmen.

8.3.2.1 Aufbau und Navigation

Am besten öffnen Sie *CodeMeter API Guide* über *CodeMeter Start Center* oder alternativ über das System-Menü "Start | Alle Programme | CodeMeter | Tools".

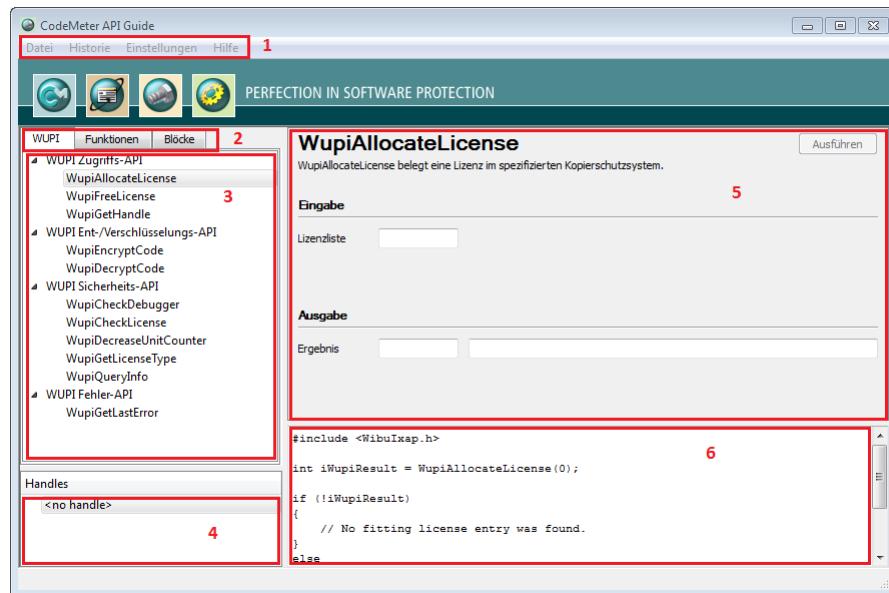


Abbildung 168: *CodeMeter API Guide* - Startbildschirm

Die *CodeMeter API Guide*-Benutzeroberfläche teilt sich in sechs separate Bereiche auf:

- Menüleiste (1)
- Karteireiter zum Wechsel zwischen WUPI, Kern-API und Blöcken (2)
- Baumansichtsfenster Funktionsaufrufe (3)
- Handle-Anzeigefenster (4)
- Interaktiv-Bereich: Eingabe- und Ausgabefeld (5)
- Quellcode-Bereich (6)

8.3.2.2 Menüleiste

Das Datei-Menü

Element	Beschreibung
Exportieren des generierten Codes	Das Wählen dieses Eintrages ermöglicht Ihnen, den generierten Code in eine separate Datei zu speichern.
Beenden	Das Wählen dieses Eintrages beendet <i>CodeMeter API Guide</i> .

Das Historie-Menü



Die Tasten-Kombination <STRG><H> öffnet jederzeit das Historie-Fenster

CodeMeter API Guide bietet Ihnen über dieses Menü die Option den Verlauf Ihrer API-Aufrufe abzuspeichern, um sie danach wiederzuverwenden.

Element	Beschreibung
Laden	Lädt die *.WibuCmAPI-Datei inklusive Code in das Historie-Fenster.
Speichern	Speichert den Verlauf der API-Aufrufe in eine *.WibuCmAPI-Datei, die Sie frei benennen und an gewünschter Stelle abspeichern.
Anzeigen	Zeigt den Verlauf Ihrer API-Aufrufe inklusive erzeugtem Code im Historie-Fenster an.

```

History der API-Aufrufe
History der API-Aufrufe

API-Funktion Rückgabewert Fehlertext
CmAccess 554 Es ist kein Fehler aufgetreten, Fehler 0.
CmAccess 555 Es ist kein Fehler aufgetreten, Fehler 0.
CmAccess 556 Es ist kein Fehler aufgetreten, Fehler 0.
CmCrypt 0 Es ist kein Fehler aufgetreten, Fehler 0.
CmCrypt 0 Es ist kein Fehler aufgetreten, Fehler 0.
CmCrypt 0 Es ist kein Fehler aufgetreten, Fehler 0.
CmCrypt 0 Es ist kein Fehler aufgetreten, Fehler 0.

Erzeugter Code
CBLOCKINFO cmBoxInfo;
memset(&cmBoxInfo, 0, sizeof(cmBoxInfo));
int res = CmSetInfo(hcmse1, CM_GET_BOXINFO, &cmBoxInfo,
sizeof(cmBoxInfo));

if(0 != res)
{
    // The real number of bytes in pDest is returned.
    // If pDest does not contain enough memory,
    // the number of needed bytes will be returned.
}

Laden Speichern Schließen

```

Abbildung 169: *CodeMeter API Guide* – Historie-Fenster

Das Einstellungen-Menü

Element	Beschreibung
Sprache	Das Wählen dieses Eintrages erlaubt Ihnen die Sprache der Oberfläche einzustellen. Sie können zwischen den Sprachen Deutsch, Englisch und Chinesisch wählen.
Programmiersprachen	Das Wählen dieses Eintrages ermöglicht Ihnen, die Programmiersprache Ihres Softwareprojektes auszuwählen. Ihnen stehen die folgenden Programmiersprachen zur Verfügung: C++, C, C#, VB.NET, VB6, Java, Delphi.

Das Hilfe Menü

Element	Beschreibung
Kontexthilfe F1	Das Wählen dieses Eintrages öffnet die kontextsensitive <i>CodeMeter API Guide</i> Hilfe. Sie erhalten Informationen über den ausgesuchten Befehl ebenfalls über Drücken der [F1]-Taste.
Info	Das Wählen dieses Eintrages öffnet ein separates Fenster mit <i>CodeMeter API Guide</i> Versionsinformationen.

8.3.2.3 Karteireiter

CodeMeter API Guide bietet Ihnen den Bereich "**Karteireiter**" an, mit denen Sie zwischen API-Aufrufen für das WUPI- und Kern-API sowie kompletten Funktionsblöcken wählen können.

Element	Beschreibung																
WUPI	Die Funktionen des Softwareschutz-API oder WUPI (WIBU Universal Protection Interface) sind im Karteireiter übersichtlich nach einzelnen Funktionsbereichen angeordnet.																
Funktionen	Den überwiegenden Teil der <i>CodeMeter Kern-API</i> Funktionen finden Sie in diesem Karteireiter.																
Blöcke	Neben einzelnen API-Befehlen finden Sie im <i>CodeMeter API Guide</i> auch komplettete Funktionsblöcke. Diese Funktionsblöcke zeigen das Lesen und Schreiben von Daten in einen <i>CmContainer</i> , die Durchführung verschiedener Verschlüsselungen sowie das Einschalten der <i>CmStick</i> -LEDs. <table border="1"> <thead> <tr> <th>Block</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>Lesen von Daten</td><td>Auslesen der Product Item Options Text, User Data, Protected Data und Protected Data.</td></tr> <tr> <td>Schreiben von Daten</td><td>Schreiben von Daten in ein Product Item. Es werden nur Operationen unterstützt die keine Firm Security Box (FSB) benötigen. Die meisten Schreiboperationen sind auf den Firm Code 0 beschränkt.</td></tr> <tr> <td>ECDSA Signatur</td><td>Verwendung des ECDSA (Elliptic Curve Digital Signature Algorithmus) zur Signierung von Daten.</td></tr> <tr> <td>Signatur-Verifikation</td><td>Verifizierung von signierten Daten.</td></tr> <tr> <td>Symmetrische Verschlüsselung</td><td>Symmetrische Ver- und Entschlüsselung binärer Daten.</td></tr> <tr> <td>Asymmetrische Verschlüsselung</td><td>Asymmetrische Verschlüsselung binärer Daten.</td></tr> <tr> <td>Asymmetrische Entschlüsselung</td><td>Asymmetrische Entschlüsselung binärer Daten mit dem <i>CmContainer</i>.</td></tr> </tbody> </table>	Block	Beschreibung	Lesen von Daten	Auslesen der Product Item Options Text, User Data, Protected Data und Protected Data.	Schreiben von Daten	Schreiben von Daten in ein Product Item. Es werden nur Operationen unterstützt die keine Firm Security Box (FSB) benötigen. Die meisten Schreiboperationen sind auf den Firm Code 0 beschränkt.	ECDSA Signatur	Verwendung des ECDSA (Elliptic Curve Digital Signature Algorithmus) zur Signierung von Daten.	Signatur-Verifikation	Verifizierung von signierten Daten.	Symmetrische Verschlüsselung	Symmetrische Ver- und Entschlüsselung binärer Daten.	Asymmetrische Verschlüsselung	Asymmetrische Verschlüsselung binärer Daten.	Asymmetrische Entschlüsselung	Asymmetrische Entschlüsselung binärer Daten mit dem <i>CmContainer</i> .
Block	Beschreibung																
Lesen von Daten	Auslesen der Product Item Options Text, User Data, Protected Data und Protected Data.																
Schreiben von Daten	Schreiben von Daten in ein Product Item. Es werden nur Operationen unterstützt die keine Firm Security Box (FSB) benötigen. Die meisten Schreiboperationen sind auf den Firm Code 0 beschränkt.																
ECDSA Signatur	Verwendung des ECDSA (Elliptic Curve Digital Signature Algorithmus) zur Signierung von Daten.																
Signatur-Verifikation	Verifizierung von signierten Daten.																
Symmetrische Verschlüsselung	Symmetrische Ver- und Entschlüsselung binärer Daten.																
Asymmetrische Verschlüsselung	Asymmetrische Verschlüsselung binärer Daten.																
Asymmetrische Entschlüsselung	Asymmetrische Entschlüsselung binärer Daten mit dem <i>CmContainer</i> .																

8.3.2.4 Baumansichtsfenster

CodeMeter API Guide bietet Ihnen zur klaren und übersichtlichen Darstellung der einzelnen API-Aufrufe einem Karteireiter eine navigierbare Baumstruktur an. Die Aufrufe sind abhängig von der Auswahl des Karteireiters thematisch in Bereiche gegliedert. Die einzelnen Wurzelknoten lassen sich bequem über die Auf- bzw. Zusammenklappen Bedienelemente (und) erweiterten oder zusammenfassen.

8.3.2.5 Handle-Anzeigefenster

In diesem Bereich zeigt Ihnen *CodeMeter API Guide* vorhandene Handles an. Ein Handle identifiziert und verweist auf ein bestimmtes Objekt, d.h. einen Eintrag im Kommunikationsprozess zwischen *CmContainer* und der *Kern-API* Schnittstelle.

Objekte mit Eintragsbezug können hier Product Items, Firm Items, *CmContainer* oder Subsysteme sein. Der durchgeführte Aufruf der API-Funktion bezieht sich dann auf den hier angezeigten / ausgewählten Handle.

8.3.2.6 Interaktiv-Bereich

Der interaktive Eingabe-Bereich erlaubt Ihnen Einträge zu Parametern und Strukturen der zuvor ausgewählten API-Funktionen vorzunehmen. In manchen Fällen öffnen sich hierzu separate Fenster zur genaueren Angabe. Die Eingaben werden für den Quellcode-Bereich übernommen.

Über die "Ausführen" Schaltfläche starten Sie den Funktionsaufruf. Der Ausgabe-Bereich zeigt Ihnen danach die Resultate der Funktionsaufrufe an, d.h. ob ein Fehler aufgetreten ist oder nicht und beispielsweise ein Verschlüsselungsergebnis.

8.3.2.7 Quellcode-Bereich

Im Quellcode-Bereich wird der Quellcode automatisch entsprechend der Angaben im Eingabe-Bereich angepasst. Sie können hier den Quellcode markieren und in ein eigenes Softwareprojekt kopieren.



Alternativ können Sie den angepassten Quellcode über den "**Datei | Exportieren des generierten Codes**" Menü-Eintrag in eine separate Datei exportieren, oder über den "**Historie | Speichern**" Menü-Eintrag den Ablauf der Funktionsaufrufe als Datei abspeichern.

8.3.3 Beispielanwendungen: CmDemo, CmCalculator, WupiCalculator

Das CodeMeter Development Kit wird mit Beispielanwendungen in verschiedenen Programmiersprachen (C++, C#, VB6, VB.NET, Delphi und Java) ausgeliefert, die Ihnen den Einstieg und den Umgang mit CodeMeter®-Funktionen erleichtern.



Sie finden die Beispiele "CmDemo" und "CmCalculator" nach Installation des CodeMeter® SDK für die jeweiligen Programmiersprachen im Verzeichnis "%Users\Public\Documents\WIBU-SYSTEMS"

Das Beispiel "WupiCalculator" finden in den jeweils aktuellen Programmiersprachen im Verzeichnis

"%\Programm Files%\WIBU SYSTEMS\AxProtector\DevKit\Samples\IxProtector\...\WupiCalculatorIndex".

Alternativ finden Sie die Beispiele über den Navigationseintrag "**Start | Alle Programme | CodeMeter | Samples**" oder über [CodeMeter Start Center](#)⁶⁷.

8.3.3.1 CmDemo

Die Beispielanwendung "CmDemo" ist ein Projekt, das die Implementierung der am häufigsten benötigten *Kern-API*-Funktionen demonstriert. Nach der Installation finden Sie die Datei *CmDemo.exe* standardmäßig im angegebenen Verzeichnis.

Die Funktionen inklusive Quelltext finden Sie in den jeweiligen Programmierdateien der entsprechenden Sprache im gleichen Verzeichnis.

Das Beispiel in C++ steht auch als Kommandozeilenvariante mit Projektdateien für MacOS X bzw. makefile für Linux zu Verfügung.

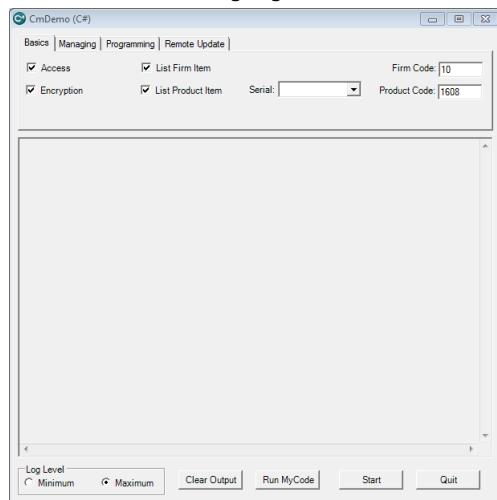


Abbildung 170: CmDemo - Überblick

Zur besseren Übersicht sind die API Funktionen in thematisch geordnete Karteikarten zusammengefasst.

Element	Beschreibung
Basics	Diese Seite zeigt den Zugriff auf Einträge und das Lesen von Einträgen, ebenso die Verschlüsselungseigenschaften von CodeMeter®. Dieser Abschnitt enthält den Code, der in den meisten CodeMeter® Implementierungen benötigt wird.
Managing	Diese Seite demonstriert das vollständige Auslesen eines <i>CmContainers</i> , das Ermitteln von internen Informationen des <i>CmContainers</i> wie z.B. die Version des CodeMeter Lizenzservers und die Version der Hardware. Der Zugriff auf die LEDs, CodeMeter® im Netzwerk und die Fehlerbehandlung werden hier ebenfalls gezeigt.
Programming	Hier werden das Programmieren und das Löschen der verschiedenen Eintragstypen dargestellt. Der Code in diesem Abschnitt kann bei der Entwicklung einer eigenen Programmieranwendung für <i>CmContainer</i> verwendet werden. <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <i>Beachten Sie hier, dass dazu eine Firm Security Box an den Rechner angegeschlossen sein muss.</i> <div style="display: flex; align-items: center; gap: 10px;"> Die Funktionen dieses Bereiches sind mittlerweile auf Herstellerseite vom Programmier-API [High Level Application Programming Interface (HIP)] abgelöst worden. Dazu gibt es eigene Beispiele. </div> </div>
Remote Update	Diese Seite demonstriert den <i>CodeMeter Field Activation Service</i> (CmFAS), d.h. die Fern-Umprogrammierung eines <i>CmContainers</i> ohne den <i>CmContainer</i> versenden zu müssen.

Zusätzliche Schaltflächen umfassen:

Element	Beschreibung
Log Level	Über die "Minimum" und "Maximum"-Option stellen Sie die Protokollierungstiefe des Darstellungsfensters ein.
Run MyCode	Die "Run MyCode"-Schaltfläche startet durch das verbundene Ereignis den Code, der in der separaten "MyCode"-Unterfunktion untergebracht ist.  Fügen Sie selbstverfassten, oder aus anderen Teilen von "CmDemo" kopierten Code in die Funktion "MyCode" ein. Sie können diesen Code nach erfolgreichem Kompilieren in der Oberfläche einfach ausprobieren.
Start	Mit dieser Schaltfläche starten Sie die ausgewählten Funktionalitäten des aktuell ausgewählten Reiters. In Abhängigkeit der eingestellten Protokollierungstiefe bekommen Sie die Informationen im Darstellungsfenster angezeigt.
Quit	Mit dieser Schaltfläche beenden Sie "CmDemo".
Clear Output	Löscht die Inhalte im Darstellungsfenster.

8.3.3.2 CmCalculator

Die Beispielanwendung "CmCalculator" ist ein Projekt, das die Verwendung einiger zentraler CodeMeter Kern-API Funktionen und Strukturen anhand eines einfachen Taschenrechner-Beispiels zeigt.

8.3.3.3 WupiCalculator

Die Beispielanwendung "WupiCalculator" zeigt wie modularer Softwareschutz in Kombination mit einem "pay-per-use"-Lizenzmodell über WUPI realisiert werden kann. Zur Verwendung des Beispiels im Zusammenhang mit IxProtector.

9 Programmierung von CmContainern und Lizenzierungsverwaltung

Nachdem Sie eine Anwendung geschützt haben, stehen Ihnen für die Programmierung der auszuliefernden *CmContainer* verschiedene Möglichkeiten offen.

Bei *CodeMeter®* spielt es übrigens keine Rolle, welchen Schritt Sie für die Abbildung Ihrer Lizenzstrategie zuerst vornehmen.

Ob Sie zunächst Ihre Lizenzmodelle bereits beim Verschlüsseln in *AxProtector* und *IxProtector* abbilden und danach die *CmContainer* programmieren, oder ob Sie zuerst die Lizenzinformationen in die *CmContainer* programmieren, und dann erst mit *AxProtector* oder *IxProtector* verschlüsseln - beides ist möglich.

Im Fall des *CodeMeter Kern-API* haben Sie diese Möglichkeit auch, indem Sie das dafür notwendige "Handle" nicht zur Laufzeit der Anwendung benutzen, sondern die WUPI Funktion **WupiGetHandle** innerhalb von *IxProtector* nutzen, den Eintrag auslesen, und ihn für Ihre Zwecke im *Kern-API* weiternutzen.

Die Programmierung der Lizenzinformationen (Firm Code, Product Code und Product Item Options) in *CmContainer* kann grundsätzlich auf drei Wegen erfolgen:

- **lokal:** Das Programmieren von lokalen *CmContainer* mit einer lokal angeschlossenen Firm Security Box (FSB).
- **dateibasiert:** Das Umprogrammieren einer Lizenzanforderungsdatei (*.WibuCmRaC-Datei), die vom Lizenznehmer an den Lizenzgeber geschickt wird, in eine Lizenzaktualisierungsdatei (*.WibuCmRaU-Datei), die der Lizenzgeber anschließend in den *CmContainer* einspielt.
- **protokollbasiert (SOAP):** Das Programmieren und Verwalten von Lizenzanforderungs- und Lizenzaktualisierungsdateien (*.WibuCmRaC und *.WibuCmRaU-Dateien) erfolgt über das internetfähige Netzwerkprotokoll SOAP (Simple Object Access Protokoll) unter Verwendung von [CodeMeter License Central](#)³⁸⁴.

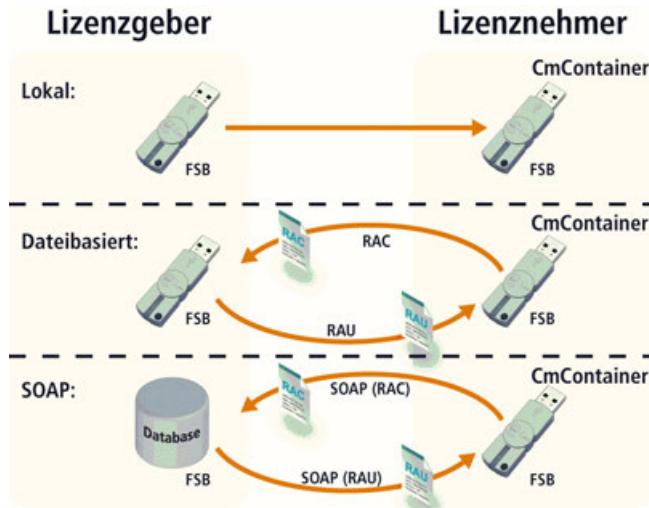


Abbildung 171: Optionen für die CmContainer Programmierung

Für diese drei unterschiedlichen Wege stellt *CodeMeter®* verschiedene Werkzeuge zur Verfügung:

- [CmBoxPgm](#)³⁵⁵: Kommandozeilen-Werkzeug zur Batch-Programmierung von CmContainer in der Produktion.
- [CodeMeter License Editor](#)³⁴⁶: Werkzeug mit graphischer Oberfläche zur Programmierung von CmDongles zum lokalen Testen von Lizenzierungsstrategien.
- [CodeMeter License Central](#)³⁴⁴: Datenbank-basiertes Werkzeug zum Erstellen, Verwalten und Ausliefern von Lizenen in einer Desktop und Internet Version über SOAP.

Die Werkzeuge CmBoxPgm, CodeMeter License Editor und CodeMeter License Central nutzen Sie zur dateibasierten [Fernprogrammierung](#)³⁵⁰, dem CodeMeter Field Activation Service (CmFAS).

Die meisten Anwendungen setzen auf dem *CodeMeter® Programmier-API* (HIP - High Level Programming Interface) auf. Diese klassenorientierte Schnittstelle lässt Sie auf alles zugreifen, was zur Programmierung und Organisation von Lizenzinträgen in einem CmContainer benötigt wird, und erlaubt ein weitgehendes Customizing.

Das *Programmier-API* gibt es für viele Programmiersprachen. Über Hilfsprogramme wurden jeweils passende Schnittstellen z.B. für Delphi, Visual Basic, .NET und Java generiert. Sie erfahren mehr über das *Programmier-API* unter dem Systemmenü-Eintrag "**Start | Alle Programme | CodeMeter | Documentation | Programming-API**".

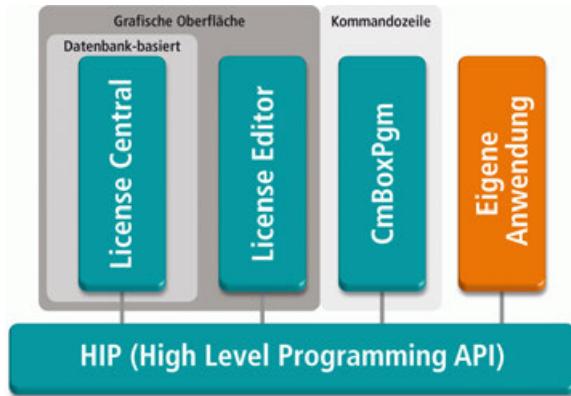


Abbildung 172: Werkzeuge zur *CmContainer* Programmierung

9.1 CodeMeter License Editor

CodeMeter License Editor ist eine Anwendung, mit der Sie Lizenzen und deren Bestandteile (Firm Item, Product Item und Product Item Options) in *CmDongles* anlegen, bearbeiten und löschen. *CodeMeter License Editor* unterstützt neben der Programmierung von lokal mit dem PC verbundenen *CmDongles* auch die dateibasierte Fernprogrammierung⁶⁶² (*CodeMeter Field Activation Service*, *CmFAS*).

 Der Einsatz von *CodeMeter License Editor* empfiehlt sich besonders, wenn Sie nur eine kleine Anzahl von *CmDongles* im Einsatz haben, z.B. während der Entwicklung oder des Tests von Lizenzierungsstrategien.

Sie erreichen das *CodeMeter License Editor* entweder über das *CodeMeter Start Center*⁶⁶ oder über das "Start | Alle Programme | CodeMeter | Tools" Systemmenü.

9.1.1 Oberfläche und Navigation

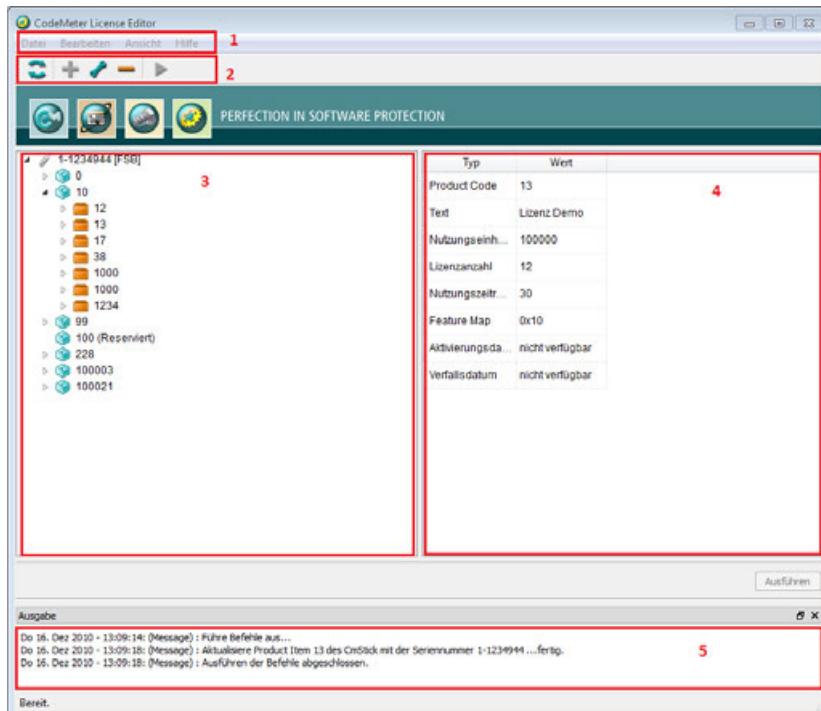


Abbildung 173: CodeMeter License Editor - Oberfläche

Die *CodeMeter License Editor*-Benutzeroberfläche teilt sich in fünf separate Bereiche auf:

- Menüleiste ³⁴⁷ (1)
- Symbolleiste ³⁴⁸ (2)
- Baumansichtsfenster ³⁴⁹ (3)
- Darstellungsfenster ³⁵⁰ (4)
- Ausgabefenster ³⁵⁰ (5)

9.1.1.1 Menüleiste

Das Datei-Menü

Element	Beschreibung
Remote Programmiermodus	Lädt über *.WibuCmRaC oder *.WibuCmRaM Dateien Lizenzinformationen in <i>CmDongles</i> in <i>CodeMeter License Editor</i> .
Direkter Programmiermodus	Lädt Lizenzinformationen in <i>CmDongles</i> direkt in <i>CodeMeter License Editor</i> . Dieser Menü-Eintrag entspricht dem Ausführen Befehl.

Element	Beschreibung
Beenden	 Beendet <i>CodeMeter License Editor</i> . Um <i>CodeMeter License Editor</i> mit Hilfe der Tastatur zu beenden, drücken Sie die <ALT+F4> Tastenkombination. Alternativ können Sie das Fenster über das  Bedienelement schließen. Vor dem Verlassen werden Sie aufgefordert vorgenommene Änderungen abzuspeichern.

Das Bearbeiten-Menü

Element	Beschreibung
Item hinzufügen	 Fügt ein neues Item hinzu. Um ein Item mit Hilfe der Tastatur hinzuzufügen, drücken Sie die <CTRL+A> Tastenkombination.
Item ändern	 Öffnet einen Dialog zum Bearbeiten eines Items. Um ein Item mit Hilfe der Tastatur zu ändern, drücken Sie die <CTRL+M> Tastenkombination.
Item löschen	 Löscht ein Item. Um ein Item mit Hilfe der Tastatur zu löschen, drücken Sie die <CTRL+D> Tastenkombination.
Ausführen	 Speichert die Änderungen an den Lizenzen in den <i>CmDongle</i> . Um Änderungen an den Lizenzen mit Hilfe der Tastatur zu speichern, drücken Sie die <CTRL+X> Tastenkombination.
Aktualisieren	 Aktualisiert die Ansicht der Lizenzen im <i>CmDongle</i> . Um mit Hilfe der Tastatur die Inhalte der <i>CmDongles</i> einzulesen, drücken Sie die <CTRL+R> Tastenkombination.

Das Ansicht-Menü

Element	Beschreibung
Status	Hier können Sie das Ausgabefenster, das sie über das Bedienelement  komplett ausgeblendet haben, wieder einblenden.

Das Hilfe-Menü

Element	Beschreibung
Hilfe	Das Wählen dieses Eintrages öffnet die Online-Hilfe zu <i>CodeMeter License Editor</i> .
Über	Das Wählen dieses Eintrages öffnet ein Fenster, das Sie über die verwendete <i>CodeMeter License Editor</i> Version informiert.

9.1.1.2 SymbolleisteAbbildung 174: *CodeMeter License Editor* - Symbolleiste

Die frei verschiebbare *CodeMeter License Editor* Symbolleiste besteht aus einem Satz von Symbolen, die gängige Aktionen darstellen. Um auf eine dieser Aktionen zuzugreifen, klicken Sie auf das Symbol für diese Aktion in der Symbolleiste.

9.1.1.3 Baumansichtsfenster

Dieses Fenster zeigt Ihnen die Inhalte der an Ihrem PC angeschlossenen *CmDongles* an.

Über die Bedienelemente klappen Sie die Wurzelknoten einzelner *CmDongles*, Lizenzebenen (Firm Items) und Lizenzeinträge (Product Items) auf bzw. zu.

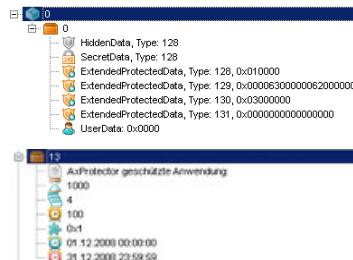


Abbildung 175: *CodeMeter License Editor* – Baumansichtsfenster

Die nachfolgende Abbildung zeigt eine Übersicht der verwendeten Symbole und ihre Bedeutung.

Symbol	Objekt
	<i>CmDongle</i>
	Firm Item (Lizenzcontainer)
	Product Item (Lizenzeinträge)
	Product Item Options
	Text
	Nutzungseinheiten (Unit Counter)
	Lizenzanzahl (License Quantity)
	Usage Period (Nutzungszeitraum)
	Feature Map
	Aktivierungsdatum (Activation Time)
	Verfallsdatum (Expiration Time)
	Wartungszeitraum (Maintenance Period)
	Hidden Data
	Secret Data
	Protected Data

Symbol	Objekt
	Extented Protected Data
	User Data

Tabelle 8: *CodeMeter License Editor* - Eintragssymbole

9.1.1.4 Darstellungsfenster

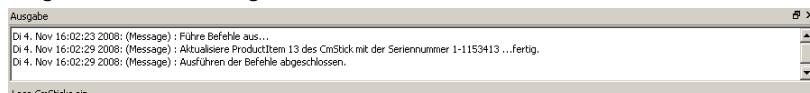
Das Darstellungsfenster zeigt in Übersichten die Details zu den im Baumansichtsfenster gewählten Objekten (*CmDongles*, Firm Item, Product Item).

Typ	Wert
Product Code	201000
Text	Basis
Nutzungseinheiten	200
Lizenzzahl	16
Nutzungszeitraum	30
Feature Map	0x1
Aktivierungsdatum	22.03.2011 13...
Verfallsdatum	23.03.2012 13...
Wartungszeitraum	Beginn: 01.01....

Abbildung 176: *CodeMeter License Editor* - Darstellungsfenster

9.1.1.5 Ausgabefenster

Das Ausgabefenster informiert Sie über ausgeführte Aktionen in *CodeMeter License Editor* und gibt mögliche Fehlermeldungen aus.

Abbildung 177: *CodeMeter License Editor* - Ausgabefenster

Über das Bedienelement können Sie das Ausgabefenster von seinem zugewiesenen Ort lösen und an einen von Ihnen bevorzugten Platz auf dem Desktop schieben. Dies kann die Übersichtlichkeit erhöhen.

Über das Bedienelement können Sie das Ausgabefenster auch komplett ausblenden. Sie blenden es wieder ein über den "Ansicht | Status" Menü-Eintrag

9.1.2 Arbeiten mit CodeMeter License Editor

Der folgende Teil zeigt Ihnen, wie Sie mit CodeMeter License Editor arbeiten.

9.1.2.1 Starten CodeMeter License Editor

Sie erreichen CodeMeter License Editor entweder über [CodeMeter Start Center](#)⁶⁶ oder über den "Start | Alle Programme | CodeMeter | Tools" Systemmenü-Eintrag.

9.1.2.2 Anzeige von angeschlossenen CmDongles

Zur Anzeige von Inhalten angeschlossener CmDongles stehen Ihnen zwei Optionen zur Verfügung. Sie lesen Lizenzdetails aus den CmDongles entweder über die Funktion **Aktualisierung** ein, oder Sie laden im **Remote Programmiermodus** eine *.WibuCmRac oder *.WibuCmRaM Datei, die die verschlüsselten Lizenzdetails enthält.

9.1.2.2.1 Aktualisieren der Anzeige

Über den "Bearbeiten | Aktualisieren" Menü-Eintrag bzw. über das  Symbol lesen Sie die Lizenzdetails aller an Ihrem PC angeschlossenen CmDongles erneut ein.

9.1.2.2.2 Remote Programmiermodus

Über den "Datei | Remote Programmiermodus" Menü-Eintrag laden Sie die entsprechenden *.Wi-buCmRac oder *.WibuCmRaM Dateien, die die verschlüsselten Lizenzdetails enthalten, die Sie im Folgenden bearbeiten möchten.

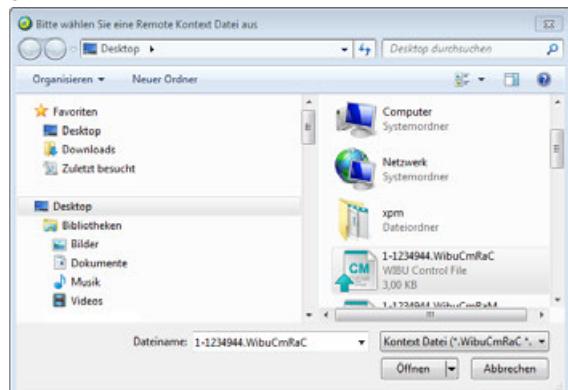


Abbildung 178: *CodeMeter License Editor – Remote Kontext Datei*

9.1.2.3 Anlegen und Ändern eines Firm Items

Wählen Sie den "Item hinzufügen" Eintrag bzw. "Item ändern" auf der Navigationsebene eines **CmDongles** über:

- das bzw. Symbol im Kontextmenü (rechte Maustaste) oder in der Symbolleiste
- den gleichnamigen **Bearbeiten** Menü-Eintrag.

Der nachfolgende Dialog erlaubt Ihnen die Eingabe bzw. das Ändern von Daten zur Konfiguration eines Firm Items.

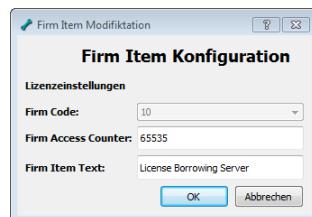


Abbildung 179: CodeMeter License Editor – Firm Item anlegen und ändern

Element	Beschreibung
Firm Code	Hier geben Sie den Firm Code ein bzw. wählen ihn aus einer Liste verfügbarer Firm Codes aus.
Firm Access Counter	Hier setzen Sie einen Zahlenwert für das Firm Item. Dieser Wert wird um 1 heruntergezählt, wenn bei einer Verschlüsselungs- oder Entschlüsselungsoperation eine spezielle Option gesetzt ist. Der Standardwert ist auf 65535 gesetzt. Wenn dieser Zähler 0 ist, ist dieses Firm Item für Verschlüsselungs- und Entschlüsselungsoperationen <u>gesperrt</u> .
Firm Item Text	Hier geben Sie den Text ein, der das Firm Item näher beschreibt.

9.1.2.4 Löschen eines Firm Items

Wählen Sie den "Item löschen" Eintrag auf der Navigationsebene eines Firm Items über:

- das Symbol im Kontextmenü (rechte Maustaste) oder in der Symbolleiste
- den gleichnamigen **Bearbeiten** Menü-Eintrag.

Der nachfolgende Dialog fordert Sie auf das Löschen des Objektes zu bestätigen.

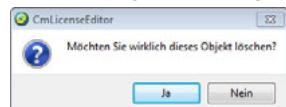


Abbildung 180: CodeMeter License Editor – Firm Item löschen

	Je nach Lizenztyp Ihres Firm Codes kann es sein, dass das Löschen eines Firm Items mit Ihrer FSB nicht möglich ist. Dies verhindert versehentliches Löschen. Diese Möglichkeit kann jederzeit nachträglich kostenlos freigeschaltet werden.
--	---

9.1.2.5 Anlegen und Ändern eines Product Items

Wählen Sie den "Item hinzufügen" Eintrag bzw. "Item ändern" auf der Navigationsebene eines Firm Items über:

- das bzw. Symbol im Kontextmenü (rechte Maustaste) oder in der Symbolleiste
- den gleichnamigen **Bearbeiten** Menü-Eintrag.

Der nachfolgende Dialog erlaubt Ihnen die Eingabe bzw. das Ändern von Daten zur Konfiguration eines Product Items.

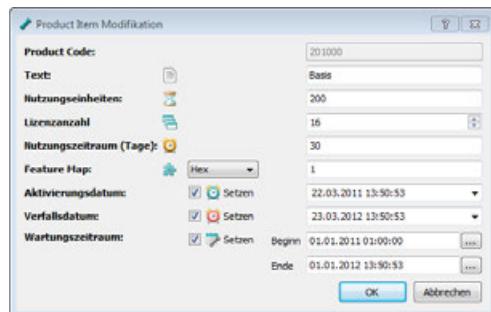


Abbildung 181: CodeMeter License Editor – Product Item anlegen

Neben dem Product Code Feld stehen Ihnen acht weitere Product Item Options zur Verfügung, mit denen Sie die Lizenz konfigurieren können.

Element	Beschreibung
Text	Geben Sie hier einen Text ein, der den Product Code, das eigentliche Produkt, näher beschreibt.
Unit Counter	Geben Sie hier die Zahl ein, von der beginnend ein Herunterzählen eines Begrenzungszählers stattfinden soll.
License Quantity	Hier geben Sie die Anzahl der Lizzen ein, die Sie für das Produkt programmieren möchten. Die Standardeinstellung (pure local license) setzt eine lokale Einzelplatzlizenz. Wie die Lizenzbelegung innerhalb eines Netzwerkes geregelt wird, haben Sie bereits in AxProtector festgelegt.
Usage Period	Geben Sie hier die Anzahl der Tage an, die die Lizenzdauer schreibt und für die Lizenz Gültigkeit besitzen soll.
Feature Map	Geben Sie hier die gewünschte Kombination der freizuschaltenden Features ein. Dies können Module, Funktionen oder verschiedene Versionen sein. Über ein Auswahlfeld ist die Eingabe sowohl im Format Binär (Bin) als auch Decimal (Dec) möglich.
Aktivierungsdatum (Activation Time)	Markieren Sie das "Setzen" Auswahlkästchen, um die PIO zu aktivieren. Geben Sie dann in das Datumsfeld das Aktivierungsdatum ein, ab wann die Lizenz gültig sein soll.

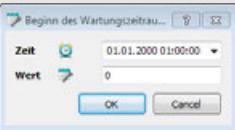
Element	Beschreibung
	<p> Beachten Sie, dass nach Erreichen des Aktivierungsdatums über Internet eine zertifizierte Zeit geholt werden muss.</p>
Verfallsdatum (Expiration Time) 	Markieren Sie das "Setzen" Auswahlkästchen, um die PIO zu setzen. Geben Sie dann in das Datumsfeld das Ablaufdatum, bis wann die Lizenz gültig sein soll.
Wartungszeitraum (Maintenance Period) 	<p>Markieren Sie das "Setzen" Auswahlkästchen, um die PIO zu aktivieren. Geben Sie dann in die Datumsfelder den Beginn und das Ende des Wartungszeitraumes ein, innerhalb dessen die Lizenz gültig sein soll.</p> <p> Erfordert CodeMeter® Firmware 1.18 oder höher.</p> 

Abbildung 182: *CodeMeter License Editor* – Wartungszeitraum anlegen

In beiden Feldern können entweder Zeiten eingegeben werden oder ganzzahlige Zeitwerte in der für CodeMeter®-üblichen Formatierung (Sekunden seit 1.1.2000). Dies deckt die derzeit bei CodeMeter üblichen Zeithorizonte bis maximal Februar 2136 ab. Die Eingabe erfolgt direkt oder über ein Kalender-Element, das sich über das links befindliche Pfeil-Symbol öffnet. Betätigen Sie die "OK" Schaltfläche, um die Eingaben zu speichern.

9.1.2.6 Löschen eines Product Items

Wählen Sie den "**Item löschen**" Eintrag auf der Navigationsebene eines Product Items über:

- das  Symbol im Kontextmenü (rechte Maustaste) oder in der Symbolleiste
- den gleichnamigen **Bearbeiten** Menü-Eintrag

Der nachfolgende Dialog fordert Sie auf das Löschen des Objektes zu bestätigen.

Abbildung 183: *CodeMeter License Editor* – Product Item löschen

9.1.2.7 Ausführen der Programmierung

Betätigen Sie die Schaltfläche "**Ausführen**" oberhalb des Ausgabefensters. Damit speichern Sie die vorgenommenen Änderungen an Firm und Product Items ab und übertragen die Lizenzdetailinformationen in die angeschlossenen *CmDongles*.

Alternativ programmieren Sie den *CmDongle* über den "**Bearbeiten | Ausführen**" Menü-Eintrag bzw.

über das Symbol .

9.2 CmBoxPgm

Neben der Programmierung von *CmDongle* mit [CodeMeter License Editor](#)³⁴⁶ und [CodeMeter License Central](#)³⁴⁷ bietet CodeMeter® auch die Möglichkeit, *CmContainer* lokal über die Konsole (Kommandozeile) zu programmieren.



Die lokale Programmierung von *CmContainer*n benötigt CodeMeter®-Transaktionen. Daher wird Ihr FSB Unit Counter jedes Mal herabgesetzt, wenn Sie einen *CmContainer* lokal programmieren.

Vorteil der Konsole

Die Kommandozeilen-Programmierung hat besonders den Vorteil, dass Sie Skripte und Batch-Dateien verwenden können. Mit Hilfe der vielfältigen Parameter können Sie in einem Durchgang die Programmierung von Abläufen auf mehrere *CmContainer* anwenden.

Einsatzgebiete

Solche Vorteile sind insbesondere in den Bereichen der Massenproduktion und der Automatisierung von Tests unverzichtbar.

Aufruf

Öffnen der *CmBoxPgm*-Kommandozeile über: "**Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt**". *CmBoxPgm* öffnet sich dann im Benutzerverzeichnispfad.

9.2.1 Syntax der Kommandozeile

Optionsblöcke

Die allgemeine Syntax in *CmBoxPgm* folgt dem Muster von sogenannten Optionsblöcken. Einzelne Programmierungen oder Auflistungen von Kommandos inklusive der Zielangaben und Optionen werden dabei zusammengefasst.

Das Muster eines Optionsblockes ist wie folgt:

```
<Zielbeschreibung> <zieldspezifische Optionen> <Vorgang>
```

Zielbeschreibung

Der Anfangsteil eines Optionsblockes beinhaltet die erforderlichen Informationen über das Ziel eines Vorgangs. Solche Ziele können sein:

- einzelne *CmContainer* oder eine Auswahl an *CmContainer*n,
- einzelne Firm Items,
- Product Items,
- Enabling Blöcke.

Die Syntax der Zielbeschreibung entspricht der hierarchischen Struktur eines *CmContainers* und ist vom Allgemeinen zum Speziellen geordnet.

Die Adressierung eines Product Item beginnt mit der Angabe des betroffenen *CmContainer*s oder der Auswahl an *CmContainer*n, geht weiter über die Angabe des Firm Codes, des Firm Items unter dem das Product Item liegt, und endet mit der Angabe des Product Items.

Der tipptechnische Aufwand lässt sich reduzieren, da Teile einer Zielangabe nicht wiederholt werden müssen, die bereits in einem vorangegangen Block angegeben wurden. Wird eine Reihe von Product Items dem selben Firm Item hinzugefügt, so ist es ausreichend das Firm Item einmalig zu Anfang der Kommandosequenz der Product Items anzugeben.

Zielspezifische Optionen

Der Mittelteil eines Optionsblockes umfasst zielgebende Optionen. Je nach Vorgang kann oder sollte dieser Abschnitt leer bleiben.

Vorgang

Der abschließende Teil beinhaltet die Angabe des Vorgangs, der durchgeführt werden soll.



Die Angabe des abschließenden Teils ist zwingend!

Die wichtigsten Vorgänge entsprechen den grundlegenden Befehlen und umfassen das Hinzufügen, Aktualisieren oder Löschen von Firm Items, Product Items oder Enabling Blöcken. Auch können Inhalte ausgewählter Items oder kompletter *CmContainer* in der Konsole aufgeführt werden.

Für die in der Programmierung verwendeten Zeitbezüge gelten die folgenden Zeitzonen.

Abkürzung	Beschreibung
CET	Central European Time, Mitteleuropäische Zeit (MEZ)
CST	Central Standard Time
EET	Eastern European Time
EST	Eastern Standard Time
MST	Mountain Standard Time
PST	Pacific Standard Time
UTC	Universal Time Coordinated, koordinierte Weltzeit

Tabelle 9: Zeitzonen in *CmBoxPgm*



Die Monatsangabe erfolgt nach dem Muster: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec.

9.2.2 Verwendung

Sie finden die Anwendung in Form der ausführbaren Datei `cmbboxpgm.exe` standardmäßig im Verzeichnis "%\Program Files%\CodeMeter\DevKit\bin". Für andere Betriebssysteme finden Sie *CmBoxPgm* an den gewohnten Stellen.



In den nachfolgenden Beschreibungen kann den Befehlen auch ein '-' statt des '/' vorangestellt werden.

9.2.3 Grundlegende Befehle

Dieser Abschnitt beschreibt die grundlegenden Befehlsoptionen der Anwendung.

Ein grundlegender Befehl schließt immer eine Befehlssequenz ab, die ein Firm Item, Product Item, oder einen Enabling Block zum Ziel hat.

Es stehen die folgenden Optionen zur Verfügung.

/ca - Add (Hinzufügen)

Hinzufügen eines neuen Eintrages in den *CmContainer* (Firm Item oder Product Item).

/cau - Add/Update (Hinzufügen/aktualisieren)

Aktualisieren eines existierenden Eintrages im *CmContainer* (Firm Item oder Product Item), oder Hinzufügen eines neuen Eintrages, wenn dieser noch nicht existiert.

/cu - Update (aktualisieren)

Aktualisieren eines existierenden Eintrages im *CmContainer* (Firm Item oder Product Item).

Beim Anlegen eines Firm Item gilt ab der Version 4.50 das Folgende:

Wird beim Anlegen eines Firm Item nicht der Text des Firm Item explizit gesetzt, so wird der für diesen Firm Code definierte Text aus der Datei *CmFirm.wbc* verwendet. Standardmäßig ist das Text-Attribut für *CmDongle* auf "Text=Test Kit Firm Code" und für *CmActLicense* auf "Text=CmAct Testkit" gesetzt.

Wollen Sie diesen Text ändern, so haben Sie zwei Optionen: entweder einen Firm Item Text [explizit setzen](#), oder den Text in der *CmFirm.wbc*-Datei editieren. Sie finden die Datei *CmFirm.wbc* im Verzeichnis "C:\ProgramData\CodeMeter\DevKit".

/cd - Delete (Löschen)

Löschen eines existierenden Eintrages im *CmContainer* (Firm Item, Product Item oder Product Item Optionen).

/cdx - Delete if possible (Löschen wenn möglich)

Löschen eines Eintrages aus dem *CmContainer* wenn verfügbar (Firm Item oder Product Item).

/1 - List (Auflisten)

Auflisten der Inhalte von ausgewählten *CmContainer*, Firm Items, Product Items oder Product Item Optionen.

Programmierbeispiele

CmBoxPgm /1

listet den Inhalt des ersten *CmContainers* auf, der keine Firm Security Box ist. Entspricht einem **CmBoxPgm /qn1 /1**

CmBoxPgm /qb1 /1

listet den Inhalt des *CmContainers* mit dem Index 1 auf.

Dabei spielt es keine Rolle, ob der *CmContainer* einen Firm Security Box Eintrag enthält oder nicht. Entspricht einem **CmBoxPgm /qn1:f /1**

CmBoxPgm.exe /qs1-1234 /1

listet den Inhalt des *CmContainers* mit der Seriennummer 1-1234 auf.

CmBoxPgm.exe /qn2,4:f /1

listet den Inhalt der *CmContainers* im Indexbereich von 2 bis 4 auf einschließlich der Firm Security Boxes.

9.2.4 CmContainer Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf einen *CmContainer* beziehen.

CmContainer können Sie anwählen:

- **einzeln**: entweder über die Seriennummer (/qs) anwählen, oder über den Index (/qb),
- **als Auswahl**: über den Index (/qb).

 Achten Sie bei der Adressierung immer darauf, ob der zu programmierende *CmContainer* eine Firm Security Box (FSB) ist.

Es stehen die folgenden Optionen zur Verfügung.

Befehl	/qb - Box index
	Bestimmen des <i>CmContainers</i> , der programmiert werden soll. Eingabe eines Dezimalwertes als einstelliger Index.
	 Darf nicht zusammen mit den Optionen /qn oder /qs verwendet werden.
Syntax	/qb<Index>
Befehl	/qnx[,y] [:f] - Box Index Range
	Bestimmen des <i>CmContainers</i> , der programmiert werden soll innerhalb eines Indexbereiches x zu y. Der höhere Index y ist optional. Wird dieser Indexwert nicht angegebenen, werden alle <i>CmContainer</i> mit einem gleichen oder höheren Index als y ausgewählt.
	 In der Standardeinstellung werden Firm Security Boxes (FSB) ausgelassen. Das Setzen von [:f] überschreibt diese Standardeinstellung. Darf nicht zusammen mit den Optionen /qb oder /qs verwendet werden.
Syntax	/qn[<Index des ersten CmContainers>,]<Index des letzten CmContainers>[:f]
Befehl	/qs[m-]s - Serial Number
	Bestimmen des <i>CmContainers</i> , der programmiert werden soll, anhand der festgelegten Maske m und Seriennummer s. Zur eindeutigen Auswahl des <i>CmContainers</i> sollten immer beide Parameter angegeben werden.
	 Darf nicht zusammen mit den Optionen /qb oder /qn verwendet werden..
Syntax	/qs<Seriennummer>
	/qs1-12345 oder /qs2-12345 Die Maske bezieht sich auf die verwendete CodeMeter® Chip-Version.
Befehl	/pwd - Box password
	Ändern des Passwortes des <i>CmContainers</i> .
Syntax	/pwd "<altes Passwort>"="<neues Passwort>"
Befehl	/r - Recursive Removal (Rekursive Löschen)
	Entfernen jedes Eintrages, dessen Löschung durch eine passende Lizenz in diesem <i>CmContainer</i> mit einer angeschlossenen Firm Security Box abgedeckt ist.
	 Dieser Befehl funktioniert nur mit Lizenzmodellen, die auf Transaktionen beruhen (Hinzufügen und Aktualisieren Product Items – PaPu), da das Firm Item gelöscht wird. Bei umfangreichen Transaktion pro <i>CmContainer</i> (Aktualisierung Firm Item – Fa) ist dies aus Sicherheitsgründen nicht möglich.
Syntax	/r
Befehl	/rau - Remote Activation Update
	Ausführen der in der angegebenen Remote Activation Update-Datei enthaltenen Programmierschritte im Ziel- <i>CmContainer</i> sofern diese anwendbar sind.
Syntax	/rau: "<RaU-Datei>"

Programmierbeispiele

CmBoxPgm	listet den Inhalt des ersten <i>CmContainers</i> auf, der keine Firm Security Box ist. Entspricht einem CmBoxPgm /qn1 /1
CmBoxPgm /qb1 /1	listet den Inhalt des <i>CmContainers</i> mit dem Index 1 auf. Dabei spielt es keine Rolle, ob der <i>CmContainer</i> einen Firm Security Box-Eintrag enthält oder nicht. Entspricht einem CmBoxPgm /qn1:f /1
CmBoxPgm /qs1-1234 /1	listet den Inhalt des <i>CmContainers</i> mit der Seriennummer 1-1234 auf.
CmBoxPgm /qn2,4:f /1	listet den Inhalt der <i>CmContainers</i> im Indexbereich von 2 bis 4 auf einschließlich der Firm Security Boxes.

9.2.5 Firm Item Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf Firm Items beziehen.

Firm Item-Befehle bauen sich wie folgt auf:

f<Firm Code> [<Firm Item Optionen>] <grundlegender Befehl>

Es stehen die folgenden Optionen zur Verfügung.

Befehl	/f - Firm Code
	Bestimmen des zu verwendenden Firm Codes. Eingabe eines Dezimalwertes ohne Vorzeichen.
Syntax	/f<Wert>
Befehl	/fac - Firm Access Counter (FAC)
	Setzen des Firm Access Counter auf den angegebenen Wert. Eingabe eines Dezimalwertes ohne Vorzeichen oder eines Hexadezimalwertes mit dem Präfix 0x.
	 Die Standardeinstellung ist 0xfffff.
Syntax	/fac<Wert>
Befehl	/fpta - Firm Precise Time, absolut
	Setzen der Firm Precise Time auf den angegebenen absoluten Wert. Eingabe eines Tagesdatums gefolgt von einer Zeitangabe und der Zeitzone.
	 Ohne Angabe einer Zeitzone, wird die Zeitzone des Systems verwendet.
Syntax	/fpta<JJJJ><Monat><TT>[,<SS>:<MM>:<SS>[PST MST CST EST UTC CET EET]]
	/fpta2006Dec31,23:59:59UTC
Befehl	/fptr - Firm Precise Time, relativ
	Hinzufügen der angegebenen Zahl an Tagen zum aktuellen Wert der Firm Precise Time. Eingabe als ganzzahliger Wert größer oder gleich Null.
	 Ist dieses Firm Item noch nicht angelegt, so wird die Systemzeit plus die angegebene Zahl als Firm Precise Time benutzt. Zum Beispiel: /fptr1 ist gleichbedeutend zu 1 Tag von jetzt beginnend.
Syntax	/fptr<Anzahl der Tage>

Befehl	/ft - Firm Item Text
	Setzen des Firm Item Texts. Eingabe als Text (bis zu 256 Zeichen) beginnend mit Doppelpunkt und den Text von zwei Anführungszeichen umgeben.
Syntax	/ft : "<Text>"
Befehl	/fuc - Firm Update Counter
	Setzen des Firm Update Counter auf den angegebenen Wert. Eingabe als Dezimalwert ohne Vorzeichen.
	 Dieser Zähler wird beim Programmieren von Einträgen automatisch „hochgesetzt“.  Dieser Wert wird automatisch vergeben und sollte normalerweise nicht explizit gesetzt werden.
Syntax	/fuc<wert>

Programmierbeispiele	
CmBoxPgm /qn1,4 /f206 /ft : "My Company" /ca	fügt den <i>CmContainer</i> innerhalb des Indexbereiches von 1 bis 4 ein neues Firm Item mit dem Firm Code 206 hinzu. Die Firm Security Boxes werden ausgeschlossen. Der Firm Item Text entspricht der Abgabe im String "My Company". Update Counter und Access Counter werden auf die Standardwerte eingestellt.
CmBoxPgm /qb2 /f206 /fuc42 /fac0x1066 /cu	aktualisiert das Firm Item mit dem Firm Code 206 im zweiten <i>CmContainer</i> . Der Firm Item Update Counter wird auf den Wert 42 gesetzt und der Firm Item Access Counter auf den Wert 0x1066.
CmBoxPgm /qs1-1234 /1 /f206 /cu /1	listet die Inhalte des <i>CmContainers</i> auf, aktualisiert das Firm Item mit dem Firm Code 206 und listet anschließend erneut die Inhalte auf.
CmBoxPgm /f206 /cd	löscht das Firm Item mit dem Firm Code 206.

9.2.6 Product Item Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf Product Items bzw. Product Item Options (PIO) beziehen.



Die notwendige Voraussetzung zur Programmierung von Product Items und PIO ist ein bereits existierendes Firm Item.

Product Item-Befehle bauen sich wie folgt auf:

```
/f<Firm Code> [...] /p<Product Code>[...] [<PIO Optionen>] <grundlegender Befehl>
```

TVB (Trailing Validation Block)

Mit Ausnahme der Options Text und User Data haben Sie die Möglichkeit für alle anderen Product Item Options zusätzlich vor der Ausführung von Programmiersequenzen eine zusätzliche Überprüfung durchzuführen.

Dabei können Sie über sogenannte Trailing Validation Blocks [TVB] Abhängigkeiten für die einzelnen Programmiersequenzen bestimmen. In Abhängigkeit von gesetzten Daten (d), Seriennummern (s) oder Update-Begrenzungszählern (u) werden Befehle, die an den *CmContainer* gesendet werden, nur dann ausgeführt, wenn Sie den angegebenen Kriterien entsprechen. Z.B. eine Programmierung erfolgt nur bei einer angegebenen Seriennummer, oder bei einer angegebenen Zahl erlaubter Aktualisierungen.

Als Standardeinstellung werden alle TVB gesetzt, so dass die Programmiersequenzen maximal variieren und eine Programmierung nur genau in den gewünschten *CmContainer* in dem vorgegebenen Zustand möglich ist.

Befehl	/p - Product Code	
	Bestimmen des zu verwendenden Product Code. Eingabe eines Dezimalwertes ohne Vorzeichen.	
 Optional können als zusätzliche Auswahlparameter der Feature Code oder eine Item-Referenz angegeben werden. Die Eingabe der Item-Referenz muss in eckigen Klammern erfolgen. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).		
Syntax	/p<wert> [=<neuer wert>] [<Feature Code> <reference>] [, <TVB dep.>]	
	Auswahl des ersten Product Items mit Product Code 13	/p13
	Auswahl des Product Item mit Product Code 13, Feature Map = 0x00000001	/p13:0x00000001
	Auswahl des Product Item mit Product Code 13, Product Item Reference = 16	/p13:[16]
Befehl	/pat - Activation Time	
	Hinzufügen, Aktualisieren oder Löschen der PIO Activation Time eines Product Items. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).	
Syntax	/pat- [a<date> r<offset>] [, <TVB dep.>]	
	 Löschen der PIO (-) oder Setzen setzt einer (a)bsoluten oder einer (r)elativen Activation Time.	
Befehl	/pata - Activation Time, absolut	
	Setzen der Activation Time auf den angegebenen absoluten Wert. Es werden ausschließlich Zeitangaben akzeptiert, die vor dem 1. Januar 2100 00:00:00 UTC liegen. Eingabe eines Datums optional gefolgt von einer Zeitangabe und einer Zeitzone.	
	 Ohne Angabe einer Zeitzone, wird die Zeitzone des Systems verwendet.	
	Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern,	

Befehl	/pata - Activation Time, absolut
	u=(u)pdate Counter, oder none).
Syntax	/pata<JJJJ><Monat><TT>[,<HH>:<MM>:<SS>[PST MST CST EST UTC CET EET] [,<TVB dep.>] oder nach ISO-8601: /pata<JJJJ>-<Monat>-<DD>[D<HH>:<MM>:<ss>[Z][±hh:mm oder ±hhmm] [±hh][,<TVB dep.>]
	Setzt die Activation Time auf den 31. Dezember 2008, 1 Sekunde vor Mitternacht, Koordinierte Weltzeit /pata2008Dec31,23:59:59UTC
Befehl	/patr - Activation Time, relativ
	Hinzufügen der angegebenen Anzahl von Tagen zur aktuellen Activation Time. Eingabe als ganzzahliger Wert größer oder gleich Null.
	 Ist dieses Firm Item noch nicht angelegt, so wird die Systemzeit plus die angegebene Zahl als Activation Time benutzt. Zum Beispiel: /patr1 ist gleichbedeutend zu 1 Tag von jetzt beginnend.
	Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).
Syntax	/patr<days>[,<TVB dep.>]
Befehl	/pcoli - Customer Owned License Information (COLI)
	Hinzufügen, Aktualisieren oder Löschen der PIO Customer Owned License Information eines Product Items. Eingabe als Text (bis zu 256 Zeichen) beginnend mit Doppelpunkt und den Text von zwei Anführungszeichen umgeben. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).
Syntax	/pcoli- /pcoli:<text>[,<TVB dep.>]  Löschen der PIO (-) oder Setzen des angegebenen Textes.
Befehl	/ped - Extended Protected Data
	Hinzufügen, Aktualisieren oder Löschen der PIO Extended Protected Data eines Product Items. Eingabe des Feldindexes [0-127] und eine Folge von Hexadezimalzeichen (bis zu 256 Bytes) mit dem Präfix 0x Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).
Syntax	/ped<extended type>- [:0x<hex data>][,<TVB dep.>]  Löschen der PIO (-) oder Setzen des angegebenen Textes. Die Angabe als Hex-Zahl muss immer geradzahlig sein. D.h. 0x1 ist ungültig, nicht aber 0x01.
	/ped0:0x75BCD15 fügt dem Feld 0 den Dezimalwert 123456789 hinzu. /ped2- Löscht das Feld 2
Befehl	/pet - Expiration Time
	Hinzufügen, Aktualisieren oder Löschen der PIO Expiration Time eines Product Items. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).

Befehl	/pet - Expiration Time	
Syntax	/pet- [a<date> r<offset>] [,<TVB dep.>]  Löschen der PIO (-) oder Setzen der angegebenen Expiration Time.	
Befehl	/peta - Expiration Time, absolut	
	Setzen der Expiration Time auf den angegebenen absoluten Wert. Es werden ausschließlich Zeitangaben akzeptiert, die vor dem 1. Januar 2100 00:00:00 UTC liegen. Eingabe eines Tagesdatums gefolgt von einer Zeitangabe und der Zeitzone.  Ohne Angabe einer Zeitzone, wird die Zeitzone des Systems verwendet.	
	Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).	
Syntax	/peta<yyyy><month><dd>[,<hh>:<mm>:<ss>[PST MST CST EST UTC CET EET] [,<TVB dep.>] oder nach ISO-8601: /peta<yyyy>-<mm>-<dd>[T<hh>:<mm>:<ss>[Z] [±hh:mm oder ±hhmm] [±hh] [,<TVB dep.>]	
	Setzt die Expiration Time auf den 31. Dezember 2009, 1 Sekunde vor Mitternacht, Koordinierte Weltzeit	
Befehl	/petr - Expiration Time, relativ	
	Hinzufügen der angegebenen Anzahl von Tagen zur aktuellen ExpirationTime. Eingabe als ganzzahliger Wert größer oder gleich Null.  Ist dieses Firm Item noch nicht angelegt, so wird die Systemzeit plus die angegebene Zahl als Expiration Time benutzt. Zum Beispiel: /petr1 ist gleichbedeutend zu 1 Tag von jetzt beginnend.	
	Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).	
Syntax	/petr<days>[,<TVB dep.>]	
	Verlängern der Expiration Time um 30 Tage.	
Befehl	/pfm - Feature Map	
	Hinzufügen, Aktualisieren oder Löschen der PIO Feature Map eines Product Items. Eingabe als Dezimalzahl ohne Vorzeichen, oder als Folge von Hexadezimalzeichen mit dem Präfix 0x. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).	
Syntax	/pfm- [<value>] [,<TVB dep.>]  Löschen der PIO (-).	
Befehl	/phd - Hidden Data	
	Hinzufügen, Aktualisieren oder Löschen der PIO Hidden Data eines Product Items. Eingabe des Feldindexes [0-127] und Eingabe einer ID für einen erweiterten PIO-Typ. Entweder als Access Code, oder als Datenbereich.	

Befehl	/phd - Hidden Data	
	Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten , s=(s)eriennummern , u=(u)pdate Counter , oder none).	
Syntax	<p>Füllt das Hidden Data PIO mit benutzerdefinierten Daten. <code>/phd<ext. type>, [<acc. code>] [:0x<hex data>] [, <TVB dep.>]</code> Füllt das Hidden Data PIO mit <count> Bytes an Zufallsdaten. <code>/phd<ext. type>, <acc. code>[:r<count>] [, <TVB dep.>]</code> Löscht die PIO (-) <code>/phd<ext. type>-</code></p>	
	<p><code>/phd15:0x1122334455</code> füllt das Feld 15 der Hidden Data PIO mit benutzerdefinierten Daten. Die Angabe als Hex-Zahl muss immer geradzahlig sein. D.h. <code>0x1</code> ist ungültig, nicht aber <code>0x01</code></p> <p><code>/phd16,<acc. code>:r32</code> füllt das Feld 16 der Hidden Data PIO mit 32 Bytes an Zufallsdaten. Der Access Code <code><acc. code></code> kann aus einer Text-Eingabe oder 16 Bytes in Hex bestehen.</p>	
Befehl	/plq - License Quantity	
	Hinzufügen, Aktualisieren oder Löschen der PIO License Quantity eines Product Items. Eingabe als Dezimalwert ohne Vorzeichen. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten , s=(s)eriennummern , u=(u)pdate Counter , oder none).	
Syntax	<code>/plq- [<counter>] [, <TVB dep.>] :w</code> Bei Angabe des Parameters <code>w</code> ist eine Lizenz auch innerhalb eines Wide Area Network (Weitverkehrsnetz), WAN, verwendbar. Zum Programmieren ist eine separater Firm Security Box (FSB)- Lizenzeintrag notwendig, den Sie von Wibu-Systems erhalten.	
	Löschen der PIO (-) oder Setzen der angegebenen License Quantity.	
	<code>/plq15</code> fügt der PIO 15 Lizenzen hinzu.	
Befehl	/plt - Linger Time	
	Hinzufügen, Aktualisieren oder Löschen der PIO Linger Time eines Product Items. Eingabe als Dezimalwert ohne Vorzeichen. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten , s=(s)eriennummern , u=(u)pdate Counter , oder none).	
Syntax	<code>/plt- [<seconds>] [, <TVB dep.>]</code>	
	Löschen der PIO (-) oder Setzen der angegebenen Linger Time.	
	<code>/plt15</code> fügt der PIO 15 eine Nachlaufzeit von 15 Sekunden hinzu.	
Befehl	/pmp – Wartungszeitraum (Maintenance Period)	
	Hinzufügen, aktualisieren oder löschen der PIO Wartungszeitraum (Maintenance Period). Eingabe des Beginn- und Endkriteriums des Zeitraumes. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten , s=(s)eriennummern , u=(u)pdate Counter , oder none).	

Befehl	/pmp – Wartungszeitraum (Maintenance Period)
	 Erfordert CodeMeter® Firmware 1.18 oder höher.
Syntax	<pre>/pmp - [d[<start date>],]<end date> i[<start value>],]<end value>[, <TVB dep.>]</pre>  Löschen der PIO (-) oder Setzen des Beginns oder des Endes des angegebenen Wartungszeitraumes (Maintenance Period). Die Eingabe erfolgt über eine Datumsangabe <d>. Alternativ kann das Datum auch ganzzahlig ohne Vorzeichen <i> in Sekunden angegeben werden. Der Startpunkt ist der 2000-01-01 00:00:00.
Befehl	/pmpd – Wartungszeitraum (Maintenance Period)(Datumsangabe)
	Hinzufügen, aktualisieren oder löschen der PIO Wartungszeitraum (Maintenance Period). Eingabe als Start- und Enddatum des Zeitraumes.  Das Startdatum kann ausgelassen werden. In diesem Fall wird das Startdatum gesetzt auf: 2000-01-01, 00:00:00 UTC. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).
	 Erfordert CodeMeter® Firmware 1.18 oder höher.
Syntax	<pre>/pmpd[<start time>,]<end time>[, <TVB dep.>]</pre> <p> /pmpd2011Jul01,00:00:00,2012Jun30,23:59:59 oder /pmpd2011-07-01T00:00:00, 2012-06-30T23:59:59</p> <p>Setzt einen einjährigen Wartungszeitraum (Maintenance Period) ab dem 1. Juli 2011.</p>
Befehl	/pmpi – Wartungszeitraum (Maintenance Period)(Ganzzahl)
	Hinzufügen, aktualisieren oder löschen der PIO Wartungszeitraum (Maintenance Period). Eingabe als Ganzzahl ohne Vorzeichen. Alternativ kann das Datum auch ganzzahlig ohne Vorzeichen <i> in Sekunden angegeben werden. Der Startpunkt ist der 2000-01-01 00:00:00.  Die Startangabe kann ausgelassen werden. In diesem Fall wird die Zahl auf einen Wert von 0 gesetzt. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).
	 Erfordert CodeMeter® Firmware 1.18 oder höher.
Syntax	<pre>/pmpi[<start value>,]<end value>[, <TVB dep.>]</pre> <p> /pmpi394416000</p> <p>Setzt einen Wartungszeitraum (Maintenance Period) bis zum 1. Juli 2012. Die Differenz von 4565 Tagen zum 1.1.2000 in Sekunden beträgt 394416000.</p>
Befehl	/pnwc - Network License Counter
	 Nicht mehr eingesetzte Option. Bitte benutzen Sie stattdessen die Option /plq mit der identischen Syntax.
Befehl	/ppd - Protected Data
	Hinzufügen, Aktualisieren oder Löschen der PIO Protected Data eines Product Items. Eingabe als Folge von Hexadezimalzeichen (bis zu 256 Bytes) mit dem Präfix 0x. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).

Befehl	/ppd - Protected Data
Syntax	<pre>/ppd- [0x<hex data>][,<TVB dep.>]</pre>  Löschen der PIO (-) oder Setzen der angegebenen Protected Data.
Befehl	/psd - Secret Data
Syntax	<p>Hinzufügen, Aktualisieren oder Löschen der PIO Secret Data eines Product Items. Eingabe des Feldindexes [0–127] und Eingabe einer ID für einen erweiterten PIO-Typ. Ein Datenbereich kann angegeben werden. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).</p> <p>Füllt das Secret Data PIO mit Daten. <code>/psd<ext. type>,[<acc. code>][:0x<hex data>][,<TVB dep.>]</code> Füllt das Secret Data PIO mit <count> Bytes an Zufallsdaten. <code>/psd<ext. type>,<acc. code>[:r<count>][,<TVB dep.>]</code> Löscht die PIO (-) <code>/psd<ext. type>-</code></p> <p> /psd- d15:0x1122334455 füllt das Feld 15 der Secret Data PIO mit benutzerdefinierten Daten. Die Angabe als Hex-Zahl muss immer geradzahlig sein. D.h. 0x1 ist ungültig, nicht aber 0x01 <code>/psd16:r32</code> füllt das Feld 16 der Secret Data PIO mit 32 Bytes an Zufallsdaten.</p>
Befehl	/pt - Text
Syntax	<p>Hinzufügen, Aktualisieren oder Löschen der PIO Text eines Product Items. Eingabe als Text (bis zu 256 Zeichen) beginnend mit Doppelpunkt und den Text von zwei Anführungszeichen umgeben.</p> <p><code>/pt -</code> <code>/pt:<text></code></p>  Löschen der PIO (-) oder Setzen des angegebenen Textes.
Befehl	/puc - Unit Counter
Syntax	<p>Hinzufügen, Aktualisieren oder Löschen der PIO Unit Counter eines Product Items. Eingabe als Dezimalwert ohne Vorzeichen. In Abhängigkeit vom gewählten Modus wird der Wert (a)absolut oder (r)elativ interpretiert. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).</p> <p><code>/puc- [a<value> r<value>][,<TVB dep.>]</code></p>  Löschen der PIO (-) oder Setzen eines (a)bsoluten oder (r)elativen Wert des Unit Counter.
Befehl	/puca - Unit Counter, absolut
	<p>Setzen des Unit Counters auf den angegebenen Wert. Eingabe als Dezimalwert ohne Vorzeichen. Bei Firmware-Versionen ab 1.18 ist dieser Wert kleiner oder gleich 4294967295. Bis Firmware 1.18 ist dieser Wert kleiner oder gleich 16777215. Individuelle TVB Abhängigkeiten für diese PIO können bestimmt werden (d=(d)aten, s=(s)eriennummern, u=(u)pdate Counter, oder none).</p>

Befehl	/puca - Unit Counter, absolut
Syntax	/puca<value>[,<TVB dep.>]
 z.B.	/puca25 Setzt den Wert der Unit Counter PIO auf den absoluten Wert von 25.
Befehl	/pucr - Unit Counter, relativ
Syntax	/pucr<signed value>[,<TVB dep.>]
 z.B.	/pucr10 Erhöht den Wert der Unit Counter PIO um einen Wert von 10.
Befehl	/pud - User Data
Syntax	/pud- [0x<hex data>]
 z.B.	 Löschen der PIO (-) oder Setzen der angegebenen User Data.
	/pud0x1122334455 weist der User Data PIO den angegebenen Wert zu. Die Angabe als Hex-Zahl muss immer geradzahlig sein. D.h. 0x1 ist ungültig, nicht aber 0x01.
Befehl	/pup - Usage Period
Syntax	Hinzufügen, Aktualisieren oder Löschen der PIO Usage Period eines Product Items. Eingabe als einen ganzzahligen Wert größer oder gleich 0.
 z.B.	 Erforderlich ist die CodeMeter® Firmware Version 1.11 oder höher.
 z.B.	/pup:30 setzt den Wert der Usage Period PIO auf 30 Tage.
Command	/pupa - Usage Period, absolute
Syntax	Setzt die Länge der Nutzungszeitraums der PIO Usage Period eines Product Items auf eine Zahl von <days>. Eingabe als einen ganzzahligen Wert größer oder gleich 0..
 z.B.	 Erforderlich ist die CodeMeter® Firmware Version 1.11 oder höher.
	/pupa[:]<days>[,<TVB dep.>]
 z.B.	/pupa:30 Löschen der PIO (-).
Command	/pupr - Usage Period, relative
	Verlängert den Nutzungszeitraum der PIO Usage Period eines Product Items um eine Zahl von <days>.

Command	/pupr - Usage Period, relative	
	Wenn die PIO nicht existiert, wird die PIO mit der angegebenen Länge angelegt. Eingabe als einen ganzzahligen Wert größer oder gleich 0.	
	 Erforderlich ist die CodeMeter® Firmware Version 1.11 oder höher.	
Syntax	/pupr [:]<days>[,<TVB dep.>]	
 z.B.	/pup:30 Löschen der PIO (-).	
Command	/pwupidata - WUPI Data	
	Addiert oder löscht eine Sequenz von Hidden Data PIOs, die als WUPI Datenspeicher benutzt werden. Ein neuer WUPI Datenspeicher kann entweder mit dem Inhalt einer Datei gefüllt werden, oder mit einem gesetzten Anfang für das Füllen der Bytes. In diesem Fall muss die Länge der WUPI Daten angegeben werden.	
Syntax	<p>/pwupidata[:e<ext. type>][,b<block size>][,a<acc. code>]:<file></p> <p>/pwupidata:s<size>[,e<ext. type>][,b<block size>][,a<acc. code>][,f<fill byte>]</p> <p>/pwupidata[:s<size>][,e<ext. type>][,b<block size>]-</p>	Nutzt den WUPI Datenspeicher mit dem Dateninhalt aus <file>. Nutzt <size> Bytes des WUPI Datenspeichers initialisiert mit dem Wert <fill byte>. Löscht den WUPI Datenspeicher.

9.2.7 CmActLicense Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf das Lizenzierungssystem *CmActLicense* beziehen.

Die folgenden Optionen werden unterstützt:

Befehl	/lac – <i>CmActLicense</i> Lizenz-Aktivierungscode
	Verwenden Sie diese Option, um den <i>CmActLicense</i> -Aktivierungs-Code für die telefonische Aktivierung zu berechnen. Geben Sie die Installations ID an.
Syntax	/lac:<installation ID>
Befehl	/laf – <i>CmActLicense</i> Lizenz-Aktivierungsdatei
	Verwenden Sie diese Option, um die Pfadinformation zur <i>CmActLicense</i> -Lizenzanforderungsdatei und zur Lizenzaktualisierungsdatei zu bestimmen.
Syntax	/laf:"<request file>\",\"<activation file>\"
Befehl	/lbind – <i>CmActLicense</i> Bindungswert
	Verwenden Sie diese Option, um einen Bindungswert zu setzen im Fall, dass die Bindungsmerkmale des

Befehl	/lbind – CmActLicense Bindungswert
	<p>Endkundenrechners bekannt sind. Erwartet wird die Eingabe einer Sequenz von 32 Bytes in hexadezimaler Notation mit einem vorangestellten 0x.</p> <p> Diese Option wird nur in Kombination mit den Bindungsschemata-Optionen /lfs:cus³⁷⁰ und /laf³⁶⁸ unterstützt. Die Angabe der Lizenzanforderungsdatei in der Option /laf muss ausgelassen werden.</p>
Syntax	/lbind:0x<hex data>
Befehl	/ldf – Anzeige der CmActLicense Lizenzdatei
	<p>Verwenden Sie diese Option, um die Inhalte der CmActLicense-Lizenzdatei anzuzeigen.</p>
Syntax	/ldf:<file>
Befehl	/ldi – Anzeige der CmActLicense-Installations ID
	<p>Verwenden Sie diese Option, um die Informationen über die CmActLicense-Installations ID anzuzeigen.</p>
Syntax	/ldi:<installation ID>
Befehl	/lfs – CmActLicense-Bindungsschemata (License Feature Set)
	<p>Verwenden Sie diese Option, um die CmActLicense-Bindungsschemata festzulegen.</p> <p>CodeMeter® SmartBind CodeMeter® SmartBind optimiert die fortbestehende Gültigkeit von Lizenzen, wenn sich Hardware-Eigenschaften des PCs ändern, an den die Lizenzen gebunden sind.</p> <p> Wibu-Systems empfiehlt diese Option zu setzen.</p> <p>Optional kann in begründeten Einzelfällen für CodeMeter® SmartBind zusätzlich eine Toleranzgrenze gesetzt werden. Sie bestimmt die zulässige Abweichung zwischen der Ausgangskonfiguration des PCs zum Zeitpunkt der Lizenzaktivierung und der aktuellen Konfiguration. Die Toleranzgrenze kann über die Parameter 1 (=tight), 2 (=medium) oder 3 (=loose) auf eng, mittel und weit gesetzt werden.</p> <p> In der Standardeinstellung verwendet CodeMeter® SmartBind die Toleranzgrenze 2. Wollen Sie diese Einstellung ändern, kontaktieren Sie bitte vorher Wibu-Systems Support.</p> <p>CmActLicense unterstützt auch Bindungsschemata, die sich entweder auf feste oder konfigurierbare Hardware-Eigenschaften des PCs beziehen.</p> <p> Wibu-Systems empfiehlt vor Nutzung dieser Optionen Wibu-Systems Support zu kontaktieren.</p> <p>Syntax: <code>/lfs:smart[:<Toleranzgrenze>]</code></p> <p>CmActLicense mit SmartBind für Lizenzen in einer VM (Virtuellen Maschine) Das Verhalten von CmActLicense mit dem Bindungsschema SmartBind ist für Lizenzen in einer VM wie folgt definiert.</p> <ul style="list-style-type: none"> • Wird die VM kopiert, d.h. die "I copied it"-Option ist gewählt worden, dann bricht die Lizenz. • Wird die VM verschoben, d.h. die "I moved it"-Option ist gewählt worden, dann bleibt die Lizenz bei gleichen CPU-Typen intakt. Sind die CPU-Typen hingegen verschieden, bricht die Lizenz ebenfalls außer die Toleranzgrenze wurde

Befehl	/lfs – CmActLicense-Bindungsschemata (License Feature Set)														
	de auf einen Wert von "3" (weit) gesetzt. Feste Hardware-Eigenschaften Für die Hardware-Bindungsschemata stehen vier feste Eigenschaften der Hardware zur Verfügung, die beliebig miteinander kombiniert werden können. Über <count> legen Sie fest, wie viele sich davon ändern oder nicht ändern dürfen.														
	<table border="1"> <thead> <tr> <th>Hardware-Eigenschaft</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>'b'</td><td>(B)IOS</td></tr> <tr> <td>'c'</td><td>(C)PU</td></tr> <tr> <td>'d'</td><td>(d)isk</td></tr> <tr> <td>'n'</td><td>(n)etwork adapter</td></tr> </tbody> </table>	Hardware-Eigenschaft	Beschreibung	'b'	(B)IOS	'c'	(C)PU	'd'	(d)isk	'n'	(n)etwork adapter				
Hardware-Eigenschaft	Beschreibung														
'b'	(B)IOS														
'c'	(C)PU														
'd'	(d)isk														
'n'	(n)etwork adapter														
	Syntax: <code>/lfs:[b][c][d][n][:<count>]</code>														
	Konfigurierbare Hardware-Eigenschaften Für die Hardware-Bindungsschemata stehen weitere konfigurierbare Eigenschaften der Hardware zur Verfügung, die nicht miteinander kombiniert werden können.														
	<table border="1"> <thead> <tr> <th>Bindungsschema</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>'ip'</td><td>(IP) Adresse</td></tr> <tr> <td>'mid'</td><td>(m)aschinen (ID); enthält die Maschinen SID und die Domain SID.</td></tr> <tr> <td>'non'</td><td>(non)e -keine Hardware-Bindung</td></tr> <tr> <td>'ran'</td><td>(ran)dom; nach dem Zufallsprinzip ausgewählte Grundlage für eine Bindung.</td></tr> <tr> <td>'ser'</td><td>Produkt- (Ser)iennummer</td></tr> <tr> <td>'cus'</td><td>(cus)tom plugin, angepasstes Plugin, das den Namen des Plugins als Eingabe erwartet (bis zu 31 Zeichen; gültige Zeichen sind 'A'..'Z', 'a'..'z', '0'..'9', '_').</td></tr> </tbody> </table>	Bindungsschema	Beschreibung	'ip'	(IP) Adresse	'mid'	(m)aschinen (ID); enthält die Maschinen SID und die Domain SID.	'non'	(non)e -keine Hardware-Bindung	'ran'	(ran)dom; nach dem Zufallsprinzip ausgewählte Grundlage für eine Bindung.	'ser'	Produkt- (Ser)iennummer	'cus'	(cus)tom plugin, angepasstes Plugin, das den Namen des Plugins als Eingabe erwartet (bis zu 31 Zeichen; gültige Zeichen sind 'A'..'Z', 'a'..'z', '0'..'9', '_').
Bindungsschema	Beschreibung														
'ip'	(IP) Adresse														
'mid'	(m)aschinen (ID); enthält die Maschinen SID und die Domain SID.														
'non'	(non)e -keine Hardware-Bindung														
'ran'	(ran)dom; nach dem Zufallsprinzip ausgewählte Grundlage für eine Bindung.														
'ser'	Produkt- (Ser)iennummer														
'cus'	(cus)tom plugin, angepasstes Plugin, das den Namen des Plugins als Eingabe erwartet (bis zu 31 Zeichen; gültige Zeichen sind 'A'..'Z', 'a'..'z', '0'..'9', '_').														
	Syntax: <code>/lfs:ip mid non ran ser cus:<plugin name></code>														
Befehl	/lif – CmActLicense-Lizenz-Informationsdatei (dateibasierte Aktivierung)														
	Verwenden Sie diese Option, um den Pfad der CmActLicense-Lizenz-Informationsdatei zu setzen.														
Syntax	<code>/lif: "<Lizenz-Informationsdatei>"</code>														
Befehl	/lip – CmActLicense-Lizenz-Informationsdatei (telefonische Aktivierung)														
	Verwenden Sie diese Option, um den Pfad der CmActLicense-Lizenz-Informationsdatei zu setzen.														
Syntax	<code>/lip: "<Lizenz-Informationsdatei>"</code>														
Befehl	/lmrt – Minimal erforderliche CodeMeter Runtime Version														
	Verwenden Sie diese Option, um die minimal erforderliche Runtime festzulegen, die für die Verwendung von CmActLicense benötigt wird. Als Argument ist eine Versionsnummer erwartet, z.B. '4.50'. Die Mindestversion, die durch CmActLicense unterstützt wird, ist 4.30.														
Syntax	<code>/lmrt:<version>. <subversion></code>														

Befehl	/lopt – CmActLicense-Lizenzoptionen														
	Verwenden Sie diese Option, um die CmActLicense-Lizenzoption zuzutzen.														
	Gültige Lizenzoption-Kennzeichner sind:														
	<table border="1"> <thead> <tr> <th>Kennzeichner</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>'vm'</td> <td>Die CmActLicense -Lizenz kann innerhalb einer (V)irtuellen (M)aschine genutzt werden.</td> </tr> <tr> <td>'reimport'</td> <td>Die CmActLicense-Freischaltdatei kann beliebig oft eingespielt werden..</td> </tr> </tbody> </table>	Kennzeichner	Beschreibung	'vm'	Die CmActLicense -Lizenz kann innerhalb einer (V)irtuellen (M)aschine genutzt werden.	'reimport'	Die CmActLicense-Freischaltdatei kann beliebig oft eingespielt werden..								
Kennzeichner	Beschreibung														
'vm'	Die CmActLicense -Lizenz kann innerhalb einer (V)irtuellen (M)aschine genutzt werden.														
'reimport'	Die CmActLicense-Freischaltdatei kann beliebig oft eingespielt werden..														
Syntax	/lopt:<license option>[,<license option>]														
Befehl	/los – CmActLicense-Lizenz-Ziel-Betriebssystem														
	Verwenden Sie diese Option, um anzugeben auf welchen Betriebssystem(en) die CmActLicense-Lizenz verwendet werden kann.														
	 Das Setzen dieser Option ist zwingend erforderlich.														
	Die folgenden Betriebssysteme werden unterstützt:														
	<table border="1"> <thead> <tr> <th>Betriebssystem</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>'Win'</td> <td>(Win)dows, alle unterstützten Windows-Versionen</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)edded</td> </tr> </tbody> </table>	Betriebssystem	Beschreibung	'Win'	(Win)dows, alle unterstützten Windows-Versionen	'Mac'	(Mac) OS X	'Lin'	(Lin)ux	'Emb'	(Emb)edded				
Betriebssystem	Beschreibung														
'Win'	(Win)dows, alle unterstützten Windows-Versionen														
'Mac'	(Mac) OS X														
'Lin'	(Lin)ux														
'Emb'	(Emb)edded														
	Sie haben auch die Option, spezielle Betriebssystem-Versionen auszuwählen:														
	<table border="1"> <thead> <tr> <th>Betriebssystem</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>'Win2000'</td> <td>Windows 2000</td> </tr> <tr> <td>'WinXp'</td> <td>Windows XP</td> </tr> <tr> <td>'WinServer2003'</td> <td>Windows Server 2003</td> </tr> <tr> <td>'WinVista'</td> <td>Windows Vista</td> </tr> <tr> <td>'WinServer2008'</td> <td>Windows Server 2008</td> </tr> <tr> <td>'Win7'</td> <td>Windows 7</td> </tr> </tbody> </table>	Betriebssystem	Beschreibung	'Win2000'	Windows 2000	'WinXp'	Windows XP	'WinServer2003'	Windows Server 2003	'WinVista'	Windows Vista	'WinServer2008'	Windows Server 2008	'Win7'	Windows 7
Betriebssystem	Beschreibung														
'Win2000'	Windows 2000														
'WinXp'	Windows XP														
'WinServer2003'	Windows Server 2003														
'WinVista'	Windows Vista														
'WinServer2008'	Windows Server 2008														
'Win7'	Windows 7														
Syntax:	/los:<OS version>[,<OS version>]														
Befehl	/lpid – CmActLicense ID (CmAct ID)														
	Verwenden Sie diese Option, um die CmActLicense ID (CmAct ID) zuzutzen. Geben Sie eine Kombination aus vier ASCII Zeichen an, um den 'major part' der ID zu bestimmen. Geben Sie optional zusätzlich ein Zahl ohne Vorzeichen an, die den 'minor part' der ID bestimmt.														
Syntax	/lpid:<major>[-<minor>]														
	<pre>major = 'ABCD'; minor = 'ABCD'; bei Auslassen des 'minor part' => minor = 0 = 123 /lpid:ABCD /lpid:ABCD-123</pre>														
Befehl	/lpn – CmActLicense Name (CmAct Name)														
	Verwenden Sie diese Option, um den CmActLicense Namen (CmActLicense Name) zu bestimmen, der im CodeMeter Kontrollzentrum angezeigt wird.														

Syntax /lpn: "<name>"

Programmierbeispiele

Wie programmiere und aktualisiere ich eine *CmActLicense*-Lizenz?

Die Erstellung und Aktivierung der rechnergebundenen *CmActLicense*-Lizenz umfasst grob die folgenden Einzelschritte:

- Erstellen eines Bindungsschemas
Ein Bindungsschema definiert die Hardware-Merkmale eines Rechners, die zur Bindung herangezogen werden. Hier bietet *CodeMeter* mit *SmartBind*® eine einfache und zugleich sichere Art, Lizenzen eindeutig an Rechner zu binden.
- Importieren eines leeren „virtuellen“ *CmContainers* durch den Lizenznehmer.
- Ermitteln der konkreten Hardware-Merkmale des Rechners über einen digitalen „Fingerabdruck“ und Transfer an den Lizenzgeber über eine Lizenzanforderungsdatei.
- Programmieren von Lizenzen für diesen *CmContainer* durch den Lizenzgeber und Senden einer Lizenzaktualisierungsdatei an den Lizenznehmer.
- Übertragen der Bindungs- und Aktivierungsinformationen über den Import der Lizenzaktualisierungsdatei durch den Lizenznehmer.
- Senden einer Quittung über den Aktivierungsvorgang an den Lizenzgeber.

 Diese Einzelschritte werden in der *CodeMeter License Central* automatisiert. Die folgende Beschreibung bezieht sich auf die Verwendung von *CmBoxPgm*.

1. Erstellen der Lizenzinformationsdatei durch den Lizenzgeber

Der Lizenzgeber erstellt einen leeren Lizenzcontainer (LIF) (*.wbb) mit dem Bindungsschema *SmartBind*²⁸.

- a) Geben Sie den Firm Code an (in diesem Fall 5010, den Evaluierungscode für *CmActLicense*).
- b) Geben Sie einen Product Code 14 an.
- c) Geben Sie den Dateinamen der Lizenzinformationsdatei an (in unserem Beispiel "TemplateDisc.wbb").
- d) Geben Sie einen Namen für die Lizenzinformationsdatei an (in unserem Beispiel "Leerer virtueller *CmContainer*"). Dieser Name erscheint im *CodeMeter Kontrollzentrum* anstelle des Namens des *CmContainers*.
- e) Geben Sie eine Product-ID ein (im Beispiel 0001). Über diese können Sie mit dem *CodeMeter* Kern-API später die passende Lizenzinformationsdatei identifizieren.
- f) Wählen Sie das Bindungsschema *SmartBind* mit der Toleranzgrenze 2 aus.
- g) Geben Sie an, für welche Betriebssysteme diese Lizenzinformationsdatei verwendet werden kann. So können sowohl die Betriebssystemfamilien angeben (Linux, Mac OS und Windows), als auch einzelne spezielle Versionen (z.B. Windows XP). Im Beispiel erstellen wir eine Lizenzinformationsdatei, die unter Windows verwendet werden kann.
Der entsprechende *CmBoxPgm*-Befehl sieht wie folgt aus:
`CmBoxPgm /f5010 /p14 /ca /lif:"TemplateDisc.wbb" /lpn:"Leerer virtueller CmContainer" /lpid:0001 /lfs:smart:2 /los:win`
- h) Die "TemplateDisc.wbb" wird nun an den Kunden ausgeliefert.

2. Erstellen einer Lizenzanforderungsdatei durch den Kunden

- a)** Der Kunde importiert die Lizenzinformationsdatei entweder durch das Ziehen der Datei auf das *CodeMeter Kontrollzentrum* oder über den Menü-Eintrag "**Datei | Lizenz importieren**" und die Lizenz wird im *CodeMeter Kontrollzentrum* angezeigt.
 - b)** Der Kunde erstellt eine Lizenzanforderungsdatei, indem der virtuelle *CmContainer*, der aktiviert werden soll, ausgewählt und danach die Schaltfläche "**Lizenz aktivieren**" gedrückt wird.
 - c)** Die über den *CmFAS-Assistenten* erstellte Lizenzanforderungsdatei schickt der Kunde dem Lizenzgeber zu.
- 3.** Erstellen der Lizenzaktualisierungsdatei durch den Lizenzgeber

Aus der erhaltenen Lizenzanforderungsdatei erzeugen Sie nun die Lizenzaktualisierungsdatei, die die Lizenzen auf den PC des Kunden freischaltet.

Die Optionen sind zum großen Teil identisch mit denen aus der Lizenzinformationsdatei. Zwingend gleich sein müssen Firm Code (/f...), Product-ID (/lpid:...), Bindungsschema (/lfs:...) und Betriebssysteme (/los:...).

Der Firm Item Text (/ft:...) und der Name des virtuellen *CmContainers* (/lpn:...) können geändert werden. Mit /laf:... geben Sie die Lizenzanforderungsdatei an und mit Komma getrennt den Namen der Lizenzaktualisierungsdatei, die Sie erstellen möchten.

Lizenzeinträge und deren Eigenschaften können beliebig hinzugefügt oder geändert werden. Im Beispiel ist für Product Code 14 lediglich der Product Item Text ergänzt und die License Quantity ist auf 1 gesetzt worden. Die License Quantity steht für die Anzahl der gleichzeitig benutzbaren Lizenzen, speziell auch als Floating Lizenz im Netzwerk.

Der entsprechende *CmBoxPgm*-Befehl sieht wie folgt aus:

```
CmBoxPgm /f5010 /ft:"CmActLicense Demo Firm Item" /Cu /p14 /pt:"Meine Erste
CmActLicense Lizenz" /plq1 /ca /laf:"[Name der Lizenzanforderungsdatei].Wi-
buCmRaC", "[Name der Lizenzaktualisierungsdatei].WibuCmRaU" /lpn:"Meine erste Li-
zenz" /lpid:0001 /lfs:smart:2 /los:win
```

Sie könnten aber auch z.B. eine zeitliche Befristung für 30 Tage ab ersten Start einfügen (/pup:30) oder einen Unit Counter mit 10 Einheiten zum herunterzählen (/puca10). Beides fügen Sie optional vor dem /ca ein.

Der entsprechende *CmBoxPgm*-Befehl sieht dann wie folgt aus:

```
CmBoxPgm /f5010 /ft:"CmActLicense Demo Firm Item" /Cu /p14 /pt:"Meine Erste
CmActLicense Lizenz" /pup:30 /puca30 /plq1 /ca /laf:"[Name der Lizenzanforde-
rungsdatei].WibuCmRaC", "[Name der Lizenzaktualisierungsdatei].WibuCmRaU" /
lpn:"Meine erste Lizenz" /lpid:0001 /lfs:smart:2 /los:win
```

Die erstellte Lizenzaktualisierungsdatei wird an den Kunden ausgeliefert.

- 4.** Import und Aktivierung der Lizenz durch den Kunden
- a)** Der Kunde aktiviert die Lizenzaktualisierungsdatei, indem der virtuelle *CmContainer*, der aktiviert werden soll, ausgewählt und danach die Schaltfläche "**Lizenz aktivieren**" gedrückt wird.
 - b)** Der Kunde schickt optional dem Lizenzgeber eine Quittung mit..
- 5.** Aktualisieren von eigenen *CmActLicense*-Lizenzen
- Die Aktualisierung (d.h. das Hinzufügen, Ändern und Löschen von Lizenzeinträgen, bzw. deren Eigenschaften) und die Reaktivierung (die erneute Freischaltung wenn sich die Bindungsinformationen geändert haben) erfolgt auf dem gleichen Weg wie die erste Aktivierung der Lizenz.
- Der Kunde erzeugt eine Lizenzanforderungsdatei. Sie erzeugen daraus eine Lizenzaktualisierungsdatei. Der Kunde spielt diese ein und kann Ihnen wahlweise eine Quittung zurückschicken. Die Kom-

mandozeilen-Optionen sind hier analog zur ersten Aktivierung der Lizenz. Sie können Eigenschaften oder weitere Lizenzeinträge (andere Product Codes) wahlweise hinzufügen.

 Bitte beachten Sie, dass Sie die Lizenzaktualisierungsdatei bei der Aktualisierung einer Lizenz nicht auf dem Rechner erstellen können, an den die Lizenz gebunden sein soll. Diese Einschränkung ist nur beim Testen relevant, da Sie in der Praxis die Lizenz ja für Ihren Kunden und nicht für sich selbst erzeugen.

Wie programmiere ich eine Trial-Lizenz?

Zum Erstellen einer *CmActLicense*  gehen Sie wie folgt vor:

1. Öffnen der *CmBoxPgm*-Kommandozeile über: "**Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt**".
CmBoxPgm öffnet sich im Benutzerverzeichnispfad.
2. Eingeben der folgenden Kommandozeile. Bitte beachten Sie, dass Sie keine Trennstriche und Zeilenumbrüche in die Kommandozeile übernehmen!

```
cmboxpgm /F5010 /ft:"Meine Firma" /cau /p2002 /pup90 /ca /laf:"UpdateTrialLicense.WibuCmRaU" /lpn:"Trial CmActLicense" /lfs:none /los:WIN /lpid:0001
```

Beschreibung

Ein *CmActLicense*-Lizenzcontainer mit einem F(irm Code) 5010 mit dem f(irm Item)t(ext) "MeineFirma" wird aktualisiert (/cau) sowie ein p(roduct Code) 2002 mit einem Nutzungszeitraum (pup) von 90 Tagen hinzugefügt (/ca).

Die zusätzlichen *CmActLicense*-Optionen umfassen eine Lizenzaktivierungsdatei (/laf) "Upda-teTrialLicense.WibuCmRaU" mit dem *CmActLicense*-Namen (/lpn) "Trial CmActLicense", die ein Bindungsschema (/lfs) "None" für WIN(dows)-Betriebssysteme (/los) und eine *CmActLicenseID* (/lpid) 0001 besitzt.

Optional könnte hier noch die Verwendung auf virtuellen Maschinen (/lopt:vm) oder alternativ zum Nutzungszeitraum ein absolutes Ablaufdatum (/peta) gesetzt werden, das kleiner als 90 Tage ist.

Eine Trial-License kann nicht aktualisiert und nur einmal eingespielt werden, d.h. die Option /re-import darf nicht gesetzt werden.

Ergebnis

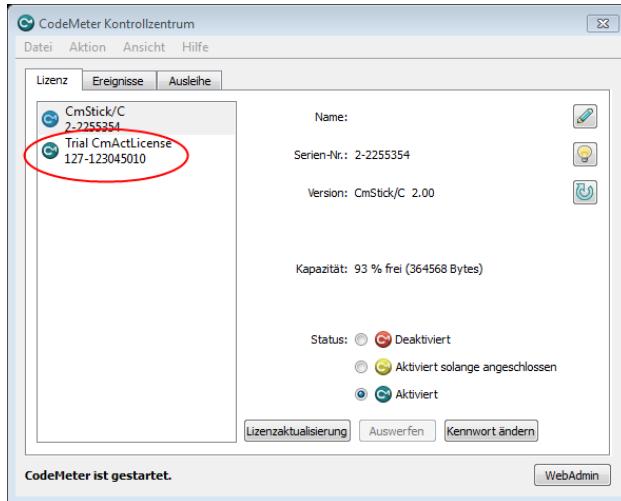
Eine Meldung ähnlich wie die unten stehende über das erfolgreiche Anlegen erscheint und die Lizenzaktivierungsdatei *UpdateTrialLicense.WibuCmRaU* wird im Benutzerverzeichnis (%USER%\\) erstellt.

```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 127-123045006, FC=5010
*** Add Product Item, CmContainer 127-123045006, PC=2002
```

3. Öffnen des *CodeMeter Kontrollzentrums*.
4. Importieren der Lizenzaktivierungsdatei.

Entweder durch das Ziehen der Datei auf das *CodeMeter Kontrollzentrum* oder über den Menü-Eintrag "**Datei | Lizenz importieren**".

Die Lizenz wird im *CodeMeter Kontrollzentrum* angezeigt.



Wie programmiere ich eine Protection Only-Lizenz?

Zum Erstellen einer *CmActLicense* [Protection Only-Lizenz](#)²⁹ gehen Sie wie folgt vor:

1. Öffnen der *CmBoxPgm*-Kommandozeile über: "Start | Alle Programme | CodeMeter | Tools | **CodeMeter Command Prompt**".
CmBoxPgm öffnet sich im Benutzerverzeichnispfad.
2. Eingeben der folgenden Kommandozeile. Bitte beachten Sie, dass Sie keine Trennstriche und Zeilenumbrüche in die Kommandozeile übernehmen!

```
cmbxpgm /F5010 /ft:"MeineFirma" /cau /p2002 /ca /laf:"UpdateProtectionOnlyLicense.WibuCmRaU" /lpn:"Protection Only CmActLicense" /lfs:none /los:WIN /lpid:0001
```

Beschreibung

Ein *CmActLicense*-Lizenzcontainer mit einem F(irm Code) 5010 mit dem f(irm Item)t(ext) "MeineFirma" wird aktualisiert (/cau) sowie ein p(roduct Code) 2002 (/ca) hinzugefügt.

Die zusätzlichen *CmActLicense*-Optionen umfassen eine Lizenzaktivierungsdatei (/laf) "UpdateProtectionOnlyLicense.WibuCmRaU" mit dem *CmActLicense*-Namen (/lpn) "Protection Only CmActLicense", die ein Bindungsschema (/lfs) "None" für WIN(dows)-Betriebssysteme (/los) und eine *CmActLicenseID* (/lpid) 0001 besitzt.

Optional könnte hier noch die Verwendung auf virtuellen Maschinen (/lopt:vm) und ein beliebig häufiges Einspielen der Lizenz erlaubt werden (/reimport).

Ergebnis

Eine Meldung ähnlich wie die unten stehende über das erfolgreiche Anlegen erscheint und die Lizenzaktivierungsdatei *Update-ProtectionOnly-License.WibuCmRaU* wird im Benutzerverzeichnis (%Users%) erstellt.

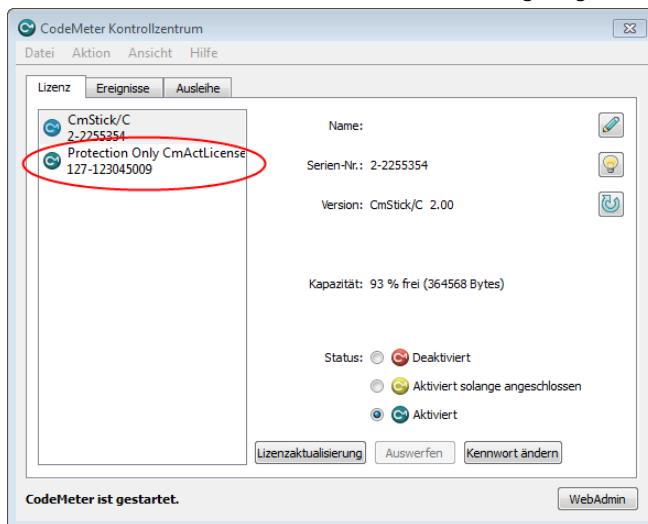
```
*** Create CmActLicense Activation File
*** Update Firm Item, CmContainer 127-123045009, FC=5010
*** Add Product Item, CmContainer 127-123045009, FC=5010, PC=2002
```

3. Öffnen des CodeMeter Kontrollzentrums.

4. Importieren der Lizenzaktivierungsdatei.

Entweder durch das Ziehen der Datei auf das CodeMeter Kontrollzentrum oder über den Menü-Eintrag "Datei | Lizenz importieren".

Die Lizenz wird im CodeMeter Kontrollzentrum angezeigt.



9.2.8 Lizenzausleihe Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf Lizenzausleih-Einträge beziehen. Lizenzausleihe-Befehle bauen sich wie folgt auf:

```
/f<Firm Code> [...] /p<Product Code> [...] [<Ausleih-Optionen>] <grundlegender Befehl>
```

Es stehen die folgenden Optionen zur Verfügung.

Befehl	/bls – Lizenzausleihe Server (License Borrowing Server)
	Programmieren einer ausleihbaren Server-Lizenz.
Syntax	<pre>/bls [: [cm ca],<fc>,<pc>,<fm>,<lqClient>,<duration> [,serverID]]</pre> <p>[cm ca] Lizenzierungssystem der Client-Lizenz (CmDongle, CmActLicense)</p> <p><fc> Firm Code der Client-Lizenz</p> <p><pc> Product Code der Client-Lizenz</p> <p><fm> Feature Map der Client-Lizenz</p> <p><lqClient> Anzahl der maximal ausleihbaren Client-Lizenzen (License Quantity)</p>

Befehl	/bls – Lizenzausleihe Server (License Borrowing Server)										
	<table border="1"> <tr> <td><duration></td><td>Dauer wie lange maximal die Lizenz ausgeliehen werden darf (in Minuten)</td></tr> <tr> <td>[,serverID]</td><td>ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.</td></tr> <tr> <td colspan="2"> Es können bis maximal 4 solcher Einträge pro Server-Lizenz programmiert werden.</td></tr> </table>	<duration>	Dauer wie lange maximal die Lizenz ausgeliehen werden darf (in Minuten)	[,serverID]	ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.	 Es können bis maximal 4 solcher Einträge pro Server-Lizenz programmiert werden.					
<duration>	Dauer wie lange maximal die Lizenz ausgeliehen werden darf (in Minuten)										
[,serverID]	ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.										
 Es können bis maximal 4 solcher Einträge pro Server-Lizenz programmiert werden.											
Befehl	/blc – Lizenzausleihe Client (License Borrowing Client)										
	Programmieren einer ausleihbaren Client-Lizenz.										
Syntax	<table border="1"> <tr> <td>/blc:[cm ca],<fc>,<pc> [,serverID]]</td><td></td></tr> <tr> <td>[cm ca]</td><td>Lizenzierungssystem der Server-Lizenz (<i>CmDongle</i>, <i>CmActLicense</i>)</td></tr> <tr> <td><fc></td><td>Firm Code der Server-Lizenz</td></tr> <tr> <td><pc></td><td>Product Code der Server-Lizenz</td></tr> <tr> <td>[,serverID]</td><td>ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.</td></tr> </table>	/blc:[cm ca],<fc>,<pc> [,serverID]]		[cm ca]	Lizenzierungssystem der Server-Lizenz (<i>CmDongle</i> , <i>CmActLicense</i>)	<fc>	Firm Code der Server-Lizenz	<pc>	Product Code der Server-Lizenz	[,serverID]	ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.
/blc:[cm ca],<fc>,<pc> [,serverID]]											
[cm ca]	Lizenzierungssystem der Server-Lizenz (<i>CmDongle</i> , <i>CmActLicense</i>)										
<fc>	Firm Code der Server-Lizenz										
<pc>	Product Code der Server-Lizenz										
[,serverID]	ServerID des Servers Diese ID ist 8 Byte lang und sollte der Notation: 0x<Heximaldaten> folgen.										

Programmierbeispiel

Ein Software-Hersteller bietet seinem Kunden an, einen Teil seiner insgesamt 50 Lizenzen mobil zu halten, d.h. lizenzierte Anwendungen laufen auch, ohne dass eine Verbindung zum Lizenzserver besteht. Dies lässt sich über das Feature Lizenzausleihe umsetzen. Insgesamt werden 10 Lizenzen für den mobilen Gebrauch vorprogrammiert. Dabei sollen 5 der Lizenzen mit *CmDongles* über zur Nutzung an beliebigen Rechnern ausgegeben werden und 5 *CmActLicenses* fest an bestimmte Laptops gebunden sein. Zu *CmActLicense* siehe das Whitepaper "Skalierbarer Softwareschutz" verfügbar über www.wibu.com.

1. Dazu muss zunächst ein Server-*CmDongle* vorprogrammiert werden.

Für die geschützte Anwendung mit dem Firm Code 10 und Product Code 1000 werden 10 Lizenzen (/plq) für die Ausleihe vorbereitet. Danach werden mit dem Befehl /bls die ausleihbaren Server-Lizenzen vorprogrammiert. Damit werden fünf *CmDongle* und *CmActLicense*-Lizenzen für 8 Stunden (480) als ausleihbar definiert. Für die *CmDongles* mit der Lizenz 10:1000, für *CmActLicenses* mit der Lizenz 5010:1000.

Befehl:

```
CmBoxPgm.exe /qb1 /f10 /ft:"Lizenzausleihs-Server" /cau /p1000 /pt:" Lizenzausleihe mit CmDongle und CmActLicense" /plq10 /
bls:cm,10,1000,0,5,480,0x12345678:ca,5010,1000,0,5,28800,0x12345678 /ca
```

2. Anschließend werden die fünf Client-*CmDongles* vorbereitet.

Für jeden der fünf *CmDongles* wird über den Befehl /blc die geschützte Anwendung mit dem Firm Code 10 und Product Code 1000 als ausleihbare Client-Lizenz verfügbar gemacht.

Befehl:

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

3. Bei der software-basierten Variante *CmActLicense* müssen zunächst die vorbereitenden Lizenzinformationsdateien (LIFs) programmiert werden.

- a) In der Lizenzinformationsdatei *CmAct.wbb* legen Sie das erforderliche Bindungsschema fest (hier z.B. Bindung an die Festplatten-Seriennummer [/1fs:D:1] und das Ziel Betriebssystem [/los:win].

Diese Datei senden Sie dann an Ihren Kunden

Befehl:

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

- b) Die *CmAct.wbb* Datei zieht dann Ihr Kunde per Drag & Drop in das *CodeMeter Kontrollzentrum* und erzeugt über den *CmFAS Assistenten* eine Lizenzanforderungsdatei (hier die "LicenseRequest.WibuCmRaC" Datei, die Ihnen zugesandt wird).

Im nächsten Programmierschritt erstellen Sie dann auf der Grundlage die Lizenzaktualisierungsdatei (hier die Datei "Activation.WibuCmRaU"), die Ihr Kunde dann über den *CmFAS Assistenten* importiert und damit die Lizenz aktiviert.

Gleichzeitig wird über den Befehl /blc die geschützte Anwendung mit dem Firm Code 5010 und Product Code 1000 als ausleihbare Client-Lizenz verfügbar gemacht.

Befehl:

```
CmBoxPgm.exe -f5010 -p1000 -blc:cm,10,1000,0x12345678 -pt:"CmActLicense-Ausleihlizenz Client-Seite" -ca /laf:"LicenseRequest.WibuCmRaC","Activation.WibuCmRaU" /lpn:"Lizenz ausleihen" /lpid:0001 /lfs:D:1 /los:win
```

Der Ausleih- und Rückgabevorgang einer Lizenz selbst erfolgt über den "[Ausleihe](#)"⁴⁵³ Karteireiter in [CodeMeter Kontrollzentrum](#)⁴⁴³. In [CodeMeter WebAdmin](#)⁴⁶⁴ wird die Belegung der Lizenen [angezeigt](#)⁴⁶⁸ wobei Anzahl und die maximale Ausleihdauer [konfiguriert](#)⁴⁶² werden können.

9.2.9 FSB Einträge Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf die Firm Security Box (FSB) beziehen.

Befehle mit Bezug auf Firm Security Boxen (FSB) bauen sich wie folgt auf:

```
/fsb<Firm Code> [<FSB Optionen>] <grundlegender Befehl>
```

Es stehen die folgenden Optionen zur Verfügung.

Befehl	/fsb - FSB Entry
	Initierung einer Befehlsabfolge für die Firm Security Box (FSB). Eingabe des Firm Codes auf den sich der FSB-Eintrag bezieht.
Syntax	/fsb<Firm Code>
Befehl	/fk - Firm Key
	Angabe eines neuen Firm Key (32 Byte). <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> Bitte lassen Sie bei diesem Befehl äußerste Vorsicht walten! Mit der Änderung eines bestehenden Firm Key greifen Sie tief in Verschlüsselungs- und Programmierprozesse ein! Denn alle künftigen Verschlüsselungen und <i>CmContainer</i> Programmierungen werden sich dann auf diesen "neuen" Firm Key beziehen!  Neu verschlüsselte Anwendungen sind mit "alt" programmierten <i>CmContainer</i> nicht mehr lauffähig! Genauso gilt, dass "alt" verschlüsselte Anwendungen nicht mit "neu" programmierten <i>CmContainer</i> laufen. Zu Ihrer eigenen Sicherheit muss die Option "Änderung des Firm Keys" durch Wibu-Systems freigeschaltet werden. </div>

Befehl	/fk - Firm Key
Syntax	/fk:0x<hex data>

9.2.10 Enabling Optionen

Dieser Abschnitt beschreibt die vorhandenen Optionen, die sich auf Enabling Blöcke beziehen.

Enabling-Befehle bauen sich wie folgt auf:

```
/f<Firm Code> [...] /e[<index>][:<type>] [<Enabling Optionen>] <grundlegender Befehl>
```

Es stehen die folgenden Optionen zur Verfügung.

Befehl	/e - Enabling												
	Bestimmen des zu programmierenden Enabling Blocks. Eingabe als Index und Typ (Simple Pin, Time Pin) des Enabling Blocks.												
Syntax	/e[<index>]:[sp tp]												
Befehl	/eac - Access Code												
	Angeben oder Ändern des Access Codes eines Enabling Blocks. Eingabe der Zeichenabfolge des Passwortes in Anführungszeichen, oder hexadezimal minimal als 2 Byte- und maximal als 16 Byte-Wert.												
	 Wenn der Firm Key als Access Code verwendet wird, geben Sie stattdessen das Kürzel 'fk' ein.												
Syntax	/eac:<access code>[=<new accesscode>] /eac:<access code> = "<text>" /eac:<access code> = 0x<hex data> /eac:<access code> = fk												
Befehl	/eatt - Anfügen Enabling Block												
	Anfügen eines Enabling Blocks an ein Firm Item oder Product Item.												
Syntax	/eatt<Firm Code>[,<Product Code>[,<Feature Code> <product item reference>]]:<Enable Level>,<Disable Level>[:req+ -] <table border="0" style="width: 100%;"> <tr> <td><Firm Code></td> <td>Verweist auf den Firm Code des Blockes</td> </tr> <tr> <td><Product Code></td> <td>Verweist auf den Product Code des Blockes</td> </tr> <tr> <td><Feature Code></td> <td>Verweist auf den Feature Code des Blockes</td> </tr> <tr> <td><product item reference></td> <td>Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/l) mit ausgegeben.</td> </tr> <tr> <td><Enable Level>,<Disable Level></td> <td>Gültige Ebenen umfassen: locate (loc), read (read), encrypt (enc), unituse (uu) oder modify (mod).</td> </tr> <tr> <td>req+ -</td> <td>Das Setzen des Pflicht-Kennzeichners dient in diesem Fall dazu, beim Aktivieren oder Deaktivieren und den damit verbundenen Berechtigungsebenen mögliche Konflikte bei mehreren vorhandenen Bindungszielen zu vermeiden. Ist mindestens ein Pflicht-Kennzeichner im Fall von mehreren Bindungszielen gesetzt, so bestimmt eine logische UND-Verknüpfung, dass alle Einstellungen der Bindungen, die einen Pflicht-Kennzeichner besitzen, zutreffen müssen bevor über eine definierte Operation auf den gesamten</td> </tr> </table>	<Firm Code>	Verweist auf den Firm Code des Blockes	<Product Code>	Verweist auf den Product Code des Blockes	<Feature Code>	Verweist auf den Feature Code des Blockes	<product item reference>	Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/l) mit ausgegeben.	<Enable Level>,<Disable Level>	Gültige Ebenen umfassen: locate (loc), read (read), encrypt (enc), unituse (uu) oder modify (mod).	req+ -	Das Setzen des Pflicht-Kennzeichners dient in diesem Fall dazu, beim Aktivieren oder Deaktivieren und den damit verbundenen Berechtigungsebenen mögliche Konflikte bei mehreren vorhandenen Bindungszielen zu vermeiden. Ist mindestens ein Pflicht-Kennzeichner im Fall von mehreren Bindungszielen gesetzt, so bestimmt eine logische UND-Verknüpfung, dass alle Einstellungen der Bindungen, die einen Pflicht-Kennzeichner besitzen, zutreffen müssen bevor über eine definierte Operation auf den gesamten
<Firm Code>	Verweist auf den Firm Code des Blockes												
<Product Code>	Verweist auf den Product Code des Blockes												
<Feature Code>	Verweist auf den Feature Code des Blockes												
<product item reference>	Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/l) mit ausgegeben.												
<Enable Level>,<Disable Level>	Gültige Ebenen umfassen: locate (loc), read (read), encrypt (enc), unituse (uu) oder modify (mod).												
req+ -	Das Setzen des Pflicht-Kennzeichners dient in diesem Fall dazu, beim Aktivieren oder Deaktivieren und den damit verbundenen Berechtigungsebenen mögliche Konflikte bei mehreren vorhandenen Bindungszielen zu vermeiden. Ist mindestens ein Pflicht-Kennzeichner im Fall von mehreren Bindungszielen gesetzt, so bestimmt eine logische UND-Verknüpfung, dass alle Einstellungen der Bindungen, die einen Pflicht-Kennzeichner besitzen, zutreffen müssen bevor über eine definierte Operation auf den gesamten												

Befehl	/eatt - Anfügen Enabling Block									
	<p><i>CmContainer</i>, eine Lizenzebene oder einen Lizenzeintrag zugegriffen darf.</p> <p>Dies entspricht der Standardeinstellung seit der Firmware-Version 1.18. Beim Binden eines Enabling Blocks über einen Eintrag in einer Lookup Table ist das Required Flag als Standard gesetzt.</p> <p>Sie können beim Programmieren des Zuordnungsprozess zwar explizit den Kennzeichner als nicht erforderlich setzen (<code>NonRequired Flag</code>), dies hat aber keine Auswirkungen, da in der Standardeinstellung <code>NonRequired Flags</code> durch eine logische ODER-Verknüpfung ignoriert werden, sobald mindestens ein Pflicht-Kennzeichner vorliegt. Dies liegt in den Einstellungen des globalen, den ganzen <i>CmContainer</i> betreffenden Enablings begründet. Wollen Sie für eigene Zwecke das globale Enabling ändern, kontaktieren Sie Wibu-Systems Support.</p>									
Befehl	/edet - Detach Enabling Block									
Syntax	<p>Loslösen eines Enabling Blocks von einem Firm Item oder Product Item.</p> <pre>/edet<Firm Code>[,<Product Code>[,<Feature Code> <product item reference>]]</pre> <table> <tr> <td><Firm Code></td><td>Verweist auf den Firm Code des Blockes</td></tr> <tr> <td><Product Code></td><td>Verweist auf den Product Code des Blockes</td></tr> <tr> <td><Feature Code></td><td>Verweist auf den Feature Code des Blockes</td></tr> <tr> <td><product item reference></td><td>Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/1) mit ausgegeben.</td></tr> </table>		<Firm Code>	Verweist auf den Firm Code des Blockes	<Product Code>	Verweist auf den Product Code des Blockes	<Feature Code>	Verweist auf den Feature Code des Blockes	<product item reference>	Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/1) mit ausgegeben.
<Firm Code>	Verweist auf den Firm Code des Blockes									
<Product Code>	Verweist auf den Product Code des Blockes									
<Feature Code>	Verweist auf den Feature Code des Blockes									
<product item reference>	Mit der Product Item Reference kann bei mehrfach vorkommenden Product Codes eine exakte Adressierung erreicht werden. Die Product Item Reference wird zum Beispiel im Rahmen den List-Befehls (/1) mit ausgegeben.									
Befehl	/edt - Disable Time									
Syntax	<p>Angeben einer Disable Time eines Enabling Blocks.</p> <p> Dieses Feature unterstützt keine Enabling Blocks vom Typ Simple PIN.</p>									
Befehl	/edta - Disable Time, absolut									
Syntax	<p>Setzen einer Disable Time eines Enabling Blocks auf ein absolutes Datum, optional gefolgt von einer Zeitangabe und Zeitzone. Ohne Angabe einer Zeitzone wird die Zeitzone des Systems verwendet.</p> <p> Über den Parameter <code>none</code> kann die Disable Time auch außer Kraft gesetzt werden.</p>									
	<p>Setzt die Disable Time auf den 31. Dezember 2008, 1 Sekunde vor Mitternacht, Koordinierte Weltzeit</p> <p><code>/edta2008Dec31,23:59:59UTC</code> <code>/edta2009-12-31T23:59:59Z</code></p> <p>Setzt die Disable Time außer Kraft</p> <p><code>/edta:none</code></p>									

Befehl	/edtr - Disable Time, relativ
	Hinzufügen der angegebenen Zahl an Tagen zum aktuellen Wert der Disable Time des Enabling Blocks.  Ist der Enabling Block noch nicht angelegt, so wird die Systemzeit plus die angegebene Zahl als Disable Time benutzt. Zum Beispiel: /edtr1 ist gleichbedeutend zu 1 Tag von jetzt beginnend.
Syntax	/edtr<Anzahl der Tage>
Befehl	/em - Enabling Mode
	Setzen des Aktivierungsstatus des Enabling Blocks. Gültige Eingabewerte sind (d)isabled oder (e)nabled.  Wenn sich der Enabling Block im Implicit Firm Item befindet kann die Aktivierung zusätzlich temporär (t) an- (+) oder ausgeschaltet (-) werden.
Syntax	/em: [d e][,][t+ -]
Befehl	/et - Text
	Setzen des Textes des Enabling Blocks. Eingabe als Text (bis zu 256 Zeichen) beginnend mit Doppelpunkt und den Text von zwei Anführungszeichen umgeben.  Dieses Feature wird nicht von Enabling Blocks vom Typ Simple PIN unterstützt.
Syntax	/et : "<text>"

9.2.11 Spezielle Befehle

Dieser Bereich beschreibt spezielle Befehle. Es stehen die folgenden Optionen zur Verfügung.

Befehl	/bkp - Backup Datei
	Aktiviert den Modus zum Betrachten der Backup-Datei. Eingabe einer <i>CodeMeter®</i> Backup-Datei.  Nur erlaubt in Kombination mit dem List-Befehl /1.
Syntax	/bkp : \<Backup Datei>\ "
Befehl	/crac - Anlegen Remote Activation Kontext-Datei (*.WibuCmRaC)
	Anlegen einer Remote Activation Kontext-Datei (*.WibuCmRaC). Eingabe (optional) einer Ziel-Datei oder eines Ziel-Verzeichnisses. Wenn eine Remote Activation Kontext-Datei (*.WibuCmRaC) angegeben wird, werden die Inhalte jedes Ziel-CmContainers dort abgelegt. Ohne Angabe der *.WibuCmRaC wird für jeden Ziel-CmContainers nach dem Muster "<serial number>.WibuCmRaC" eine separate Remote Activation Kontext-Datei im angegebenen Ziel-Verzeichnis erstellt. Ist kein Ziel-Verzeichnis angegeben, so wird das aktuelle Verzeichnis als Ziel-Verzeichnis gewählt.  Dieser Befehl darf nicht im Remote Activation Modus benutzt werden.
Syntax	/crac[:*<.WibuCmRaC file> *<.WibuCmRaC target directory>]

Befehl	/ra - Remote Activation	
Ermitteln und Schreiben der angegebenen Befehlssequenzen für die Fernprogrammierung. Die Inhalte der Ziel-CmContainers werden aus der angegebenen Remote Activation Context-Datei (=*.WibuCmRaC) bzw. Remote Activation Modified Context-Datei (=*.WibuCmRaM) importiert. Die erzeugten Befehlssequenzen werden in eine Remote Activation Update-Datei (=*.WibuCmRaU) gespeichert.		
Syntax	<code>/ra:<*.WibuCmRaC file> <*.WibuCmRaM file>[,<*.WibuCmRaU file>[,<*.Wi-buCmRaM file>]]</code>	
	<code>/ra:MyCmContainer</code> <code>/ra:Context-WibuCmRaC,Context-WibuCmRaM,Context-WibuCmRaU</code>	
	Ermitteln und Schreiben der angegebenen Befehlssequenzen.	
Befehl	/rcl - Registry aufräumen (cleanup) (Nur für CmDongle)	
Löschen der Windows-bezogenen USB Registry-Einträge nach Auswahl von Kategorien. Die unterstützten CmDongle-Varianten umfassen: <i>CmStick (c), CmStick/M (m), Removable Media (r).</i>		
	 Wird die Kategorie-Angabe weggelassen, werden als Standard alle Registry-Einträge der Kategorien (c) und (m) gelöscht.  Zum Ausführen dieses Befehls werden Administrator-Rechte benötigt.	
	 Bitte nutzen Sie diesen Befehl mit äußerster Vorsicht! Das Aufräumen der Registry kann unbeabsichtigte Wirkungen nach sich ziehen!  Das Aufräumen der Kategorie Removable Media (r) kann den Effekt haben, dass auch Einträge gelöscht werden, die sich nicht auf CmDongle beziehen.	
Syntax	<code>/rcl[:cmr]</code>	
Befehl	/sqd - Abfolge Ausgabe (dump)	
Ausgeben der erzeugten Befehlssequenzen.		
Syntax	<code>/sqd</code>	
Befehl	/log - Logging	
Aktivieren der Protokollierung. Optional kann eine Protokolldatei (log) angegeben werden.		
	 Wird die Protokolldatei nicht angegeben, so wird ein Standardname benutzt.  Die Angabe des optionalen Anzeigesymbols '+' bewirkt, dass das Protokollierungsergebnis an die Protokolldatei angehängt wird.  Ohne Verwendung dieses Symbols wird der Inhalt der Protokolldatei überschrieben.	
Syntax	<code>/log "[logfile]"[+]</code>	
Befehl	/? - Hilfe	
Ausgeben von weiterführenden Hinweisen zu gewünschten Themen.		
	 Werden keine Themen (Topics) angegeben, so wird die komplette Hilfelisten ausgegeben.	
Syntax	<code>/? [<topic>] <option></code>	
Befehl	/v - Ausführlicher Modus (verbose)	
Aktivieren des ausführlichen Modus.		
Syntax	<code>/v</code>	

Befehl	/val - Validierungsmodus (validation)
	<p>In diesem Modus gibt ein <i>CmContainer</i> nach jeder erfolgreichen Programmoperation eine Bestätigungssequenz zurück.</p> <p>Die empfangenen Daten werden mit der Firm Security Box validiert.</p> <p> Standardmäßig ist dieser Modus deaktiviert und dadurch die Leistungsfähigkeit erhöht. Sie können mit dieser Option jedoch die Bestätigungsroutine wieder aktivieren.</p>
Syntax	/val

Programmierbeispiele

Allgemeine Programmierbeispiele für *CmBoxPgm*

```
CmBoxPgm /qs1-1234 /f206 /p2001 /petr30 /puca1492 /pfm0x8000 /ca
```

Dem Firm Item mit dem Firm Code 206 im *CmContainer* 1-1234 wird ein Product Item mit dem Product Code 2001, eine 30-Tage Expiration Time, ein Unit Counter Wert von 1492 und die Feature Map 0x8000 hinzugefügt.

```
CmBoxPgm /qs1-1234 /f206 /p2001 /petr335 /pucr426 /pt: "Text" /cu
```

Aktualisiert das Product Item mit dem Product Code 2001. Die Expiration Time wird um 335 Tage erweitert und der Unit Counter um 426 Einheiten erhöht. Außerdem wird dem Product Item ein Text hinzugefügt.

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

Löscht die Expiration Time des Product Item 2001.

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

Fügt per Remote Programming (CmFAS) das Product Item 2008 hinzu. Neben der Remote Update Datei wird ebenfalls eine modifizierte Remote Context Datei (*.WibuCmRaM) angelegt, mit zu einem späteren Zeitpunkt erneut Remote Programming Operationen durchgeführt werden können.

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

Setzt im Product Item 2001 im Firm Item 206 die Lizenzanzahl auf 5.

9.3 Die CodeMeter License Central

Ticketsystem

Die Integration des Softwareschutzes in die Software beeinflusst vor allem die Sicherheit des Systems. Demgegenüber entscheidet eine effiziente Integration in Vertriebs-, Produktions- und Support-Prozesse über die Bedienbarkeit eines Systems und damit über die Akzeptanz bei Kunden und eigenen Mitarbeitern. Diese Einbindung wird unter Back Office Integration (BOI) zusammengefasst.

9.3.1 Das Prinzip

Im Mittelpunkt dieser "Back Office Integration" steht die *CodeMeter License Central*. Die *CodeMeter License Central* ist ein Ticketsystem mit einheitlicher Oberfläche, das zur Erstellung, Verwaltung und Auslieferung von *CmDongles* und *CmActLicenses* verwendet wird.



Eine detaillierte Beschreibung der *CodeMeter License Central* entnehmen Sie dem Handbuch, das Sie auf der Webseite www.wibu.de im Entwicklerbereich herunterladen können.

Editionen der CodeMeter License Central

Die *CodeMeter License Central* ist in zwei Editionen verfügbar:

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

Funktionell sind beide Editionen identisch, nur der lizenziertechnisch erlaubte Einsatz ist unterschiedlich; und damit der Support bei der Integration und Installation.

Die *Desktop Edition* darf auf einem Server intern in Ihrem Unternehmen eingesetzt werden. Als Betriebssystem wird Linux Ubuntu unterstützt. Die Datenbank läuft auf MySQL. Sie greifen von beliebig vielen Clients über ein Web-Frontend auf die *CodeMeter License Central* zu. Ein VM-Image, das die Systemvoraussetzungen erfüllt, wird von Wibu-Systems zur Verfügung gestellt.

Die *Internet Edition* können Sie auf mehrere Server in Ihrem Unternehmen verteilen. D.h. Sie können als Datenbank einen bereits vorhandenen Datenbank-Server verwenden, MySQL oder Microsoft SQL Server. Weitere sind auf Anfragen möglich. Der Kern der Anwendung (ein Apache Web Server und ein Apache Tomcat) können auf anderen Linux-Distributionen oder auf Windows installiert werden. Sie können die *CodeMeter License Central* in Ihr Warenwirtschaftssystem, CRM-System und Ihren Online Shop integrieren sowie den Kunden ermöglichen, die Lizenzen direkt per Internet abzuholen.

Sales Interface

Wenn Sie einen *CmContainer* für ein bestimmtes Produkt programmieren möchten, schicken Sie eine entsprechende Anfrage mit der Artikelnummer an die *CodeMeter License Central*. Sie erhalten ein eindeutiges Ticket zurück. Dies geschieht meist im Rahmen eines Verkaufs dieses Artikels, daher nennen wird diese Schnittstelle Sales Interface. Das Ticket stellt die Berechtigung dar, die "gekauften" Lizenzen in einen *CmContainer* zu programmieren.

Depot Interface

Sie entscheiden, ob Sie die Programmierung der Lizenz gleich selbst vornehmen, später erledigen, oder das Ticket an Ihren Kunden weitergeben. Ihr Kunde kann sich dann die gekauften Lizenzen zu einem beliebigen Zeitpunkt in einem beliebigen *CmContainer* abholen. Das Interface zum Abholen von Lizenzen nennen wir Depot Interface.

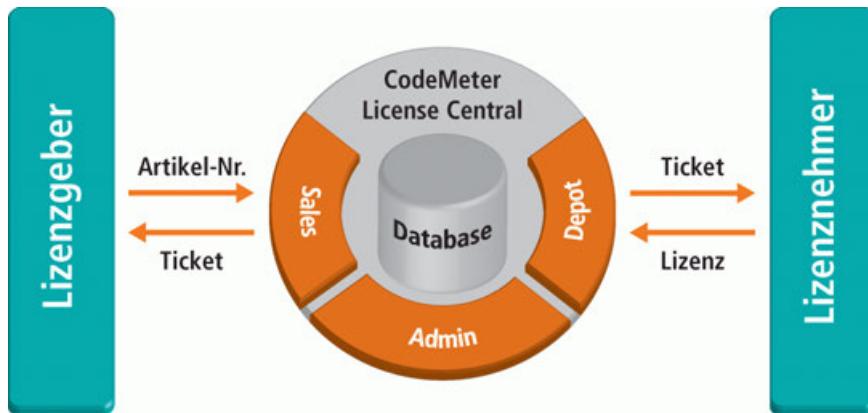


Abbildung 184: Abholung einer Lizenz durch den Lizenznehmer

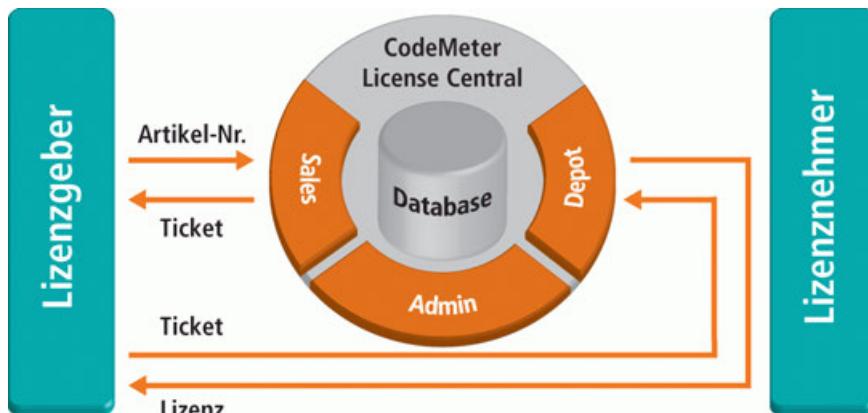
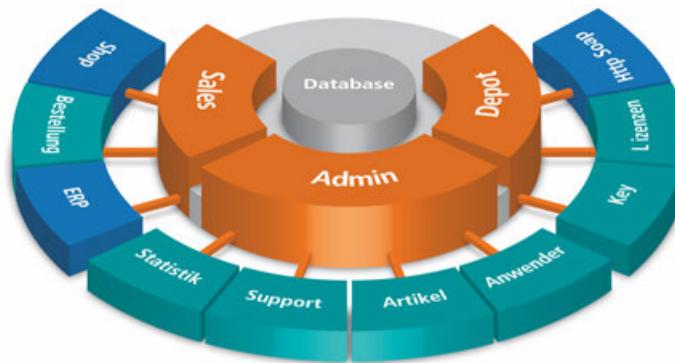


Abbildung 185: Abholung einer Lizenz durch den Lizenzgeber

Admin Interface

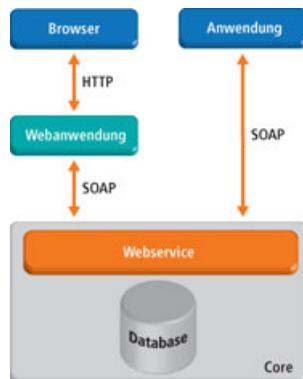
Neben dem Depot Interface und dem Sales Interface besitzt die *CodeMeter License Central* ein Admin Interface. Das Admin Interface bietet Ihnen Funktionen für die Definition von Lizenzeigenschaften (z.B. Ablaufdatum, Lizenzanzahl), die Verwaltung der Zugriffsrechte, die Erzeugung von Statistiken und Berichten sowie die Durchführung von Support-Tätigkeiten.

Die folgende Abbildung gibt einen Überblick über Module und zugehörige Funktionen der *CodeMeter License Central*.

Abbildung 186: *CodeMeter License Central – Module und Funktionen*

9.3.2 Architektur

Der Kern der *CodeMeter License Central* besteht aus einer Datenbank und Web Services für das Sales Interface, das Depot Interface und das Admin Interface. Die Web Services sind plattformunabhängig in Java verfügbar. Voraussetzung ist ein Tomcat Application Server. Die Web Services stellen eine SOAP basierte Schnittstelle zur *CodeMeter License Central* zur Verfügung. Die komplette Kommunikation mit der *CodeMeter License Central* erfolgt über diese Web Services. Die Web Services haben eine interne Schnittstelle zur Datenbank. Als Datenbank werden MySQL (Windows / Linux) und MSSQL (Windows) unterstützt. Weitere Datenbanken können auf Anfrage integriert werden.

Abbildung 187: *SOAP-Zugriffe auf die CodeMeter License Central*

Eine Web-Anwendung (Apache/PHP) sorgt dafür, dass Sie die *CodeMeter License Central* sofort ohne Anpassungen einsetzen können. Bei der Integration des Sales-Interfaces in Ihr ERP-/CRM-System oder eine eigene Anwendung steht Ihnen ein Web Service zur Verfügung.

9.3.3 Funktionen

Die Interfaces umfassen die folgenden Funktionen. Welche davon Sie in welcher Edition der *CodeMeter License Central* zum Integrieren in Ihr ERP-/CRM-System oder zum Freischalten von Lizenzen über das Ticket aus Ihrer eigenen Software heraus benutzen dürfen, entnehmen Sie den Lizenzbedingungen.

9.3.3.1 Sales-Interface

Das Sales Interface nimmt Vorgänge entgegen. Sie schicken die Artikelnummer des Produktes, das ausgeliefert werden soll, sowie eine Kundenummer und eine optionale Auftragsnummer an das Sales-Interface. Das Sales Modul liefert Ihnen dann das passende Ticket zurück.

Bei wiederkehrenden Vorgängen (z.B. Abfrage von Kontrollpunkten, Verlängerung von Lizenzen) kann auch die originale Auftragsnummer mitgegeben werden. In diesem Fall wird das bestehende Ticket um einen Abholvorgang erweitert. D.h. der Lizenznehmer kann die Lizenz mit seinem bereits bekannten Ticket erweitern. Dies spart Ihnen den Verwaltungsaufwand neuer Tickets und erleichtert dem Lizenznehmer das Arbeiten. Sie können dann sogar mit dem vorhandenen Ticket - direkt per SOAP aus Ihrer Anwendung heraus - die Lizenz automatisch erweitern oder verlängern.

Je nach Artikelkonfiguration können bei einem Vorgang auch Parameter dynamisch übergeben werden. Damit können Sie zum Beispiel die Anzahl der Netzwerklicenzen übergeben, oder den Namen des Lizenznehmers, wenn dieser in Customer Owned License Information geschrieben werden soll.

Ein Verkaufsvorgang kann auch effizient und einfach in Online Shops oder ERP-/CRM-Systeme eingebunden werden (Connectoren).

9.3.3.1.1 Connectoren

Die *CodeMeter License Central* ermöglicht bei der Erstellung von Tickets die Anbindung bestehender Systeme über zwischengeschaltete Adapter, den sogenannten Connectoren.



Diese Option steht nicht für die *Desktop* Edition zur Verfügung; hier wird das Ticket manuell über das Web-Interface des Browsers erstellt.

In der *Desktop* und *Internet* Edition können Tickets hingegen auch über eine SOAP-Schnittstelle automatisch erzeugt werden. Dies ist möglich, da in der *CodeMeter License Central* der eigentliche Verkaufsprozess von der Erstellung der Lizenz getrennt ist. Beim Verkauf der Lizenz erstellt die *CodeMeter License Central* ein Ticket, mit dem die Lizenz zu einem späteren Zeitpunkt abgeholt werden kann. Dies macht die Anbindung an bestehende Systeme, wie z.B. Online Shops oder ERP-/CRM-Systeme möglich.

Technisch gesehen, sind Connectoren Adapter, die den Datenaustausch zwischen der SOAP-Schnittstelle und anderen bestehenden Systemen ermöglichen. Dabei werden Daten passend zueinander abgebildet, d.h. gemappt. Ein Connector ist aber nicht nur ein reiner Adapter, sondern er kann auch als aktive Komponente in den Prozess eingreifen, d.h. Zusatzinformationen in eigene Tabellen sichern oder aus eigenen Tabellen auslesen, und diese zur Erstellung von Tickets verwenden, das Versenden des Tickets per E-mail durchführen, u.a. m.

Umgesetzt werden Connectoren über das Ansprechen eines standardisierten Web Services, der die automatisierte Kommunikation ermöglicht.

Online Shops

Im Fall von Online Shops, wie z.B. asknet, Cleverbridge, Digital River, element 5 oder ShareIt, die webbasierte Lizenzgeneratoren anbieten, stehen Ihnen einfache Connectoren als reine Daten-Mapper in PHP zur Verfügung.

ERP/CRM-Systeme

Während die meisten Online Shops bereits web-basierte Interfaces bieten, die einfach konfiguriert werden können, ist die Integration in ein ERP-/CRM-System meist individueller. Der prinzipielle Weg zur Integration ist der Gleiche. Das ERP-/CRM-System ruft einen Connector auf, dieser verarbeitet die Daten, bedient die SOAP-Schnittstelle der *CodeMeter License Central* und liefert das Ticket an das ERP-/CRM-System zurück.

Neben diesem klassischen Fall (das ERP-/CRM-System startet den Connector) besteht auch die Möglichkeit, einen eigenen Connector zu schreiben, der die zu bearbeitenden Daten periodisch aus dem ERP-/CRM-System ausliest bzw. exportierte Daten einliest und diese verarbeitet. Dieses Szenario ist vor allem dann anzutreffen, wenn die Betreuer des ERP-/CRM-Systems dieses nicht anpassen wollen oder nicht können. Nachteil dieser Lösung ist, dass die Tickets nicht im ERP-/CRM-System verfügbar sind und der Mitarbeiter bei Rückfragen immer in einem weiteren System nachschauen muss. Der Vorteil liegt in der einfachen und universellen Umsetzung, ein periodischer automatischer Export der Daten aus dem ERP-/CRM-System ist in allen bisher realisierten Projekten möglich gewesen.

Gerne berät Sie WIBUconcepts bei der individuellen Umsetzung der Integration: von der Konzeption bis zur Realisierung.

9.3.3.1.2 Gateway

Ein Gateway erlaubt Ihnen Lizenzen direkt aus Ihrer geschützten Anwendung abzuholen.



Ein fertiges Gateway zum automatischen Abholen von Lizenzen über das Internet ist als Bestandteil der *Internet Edition* der *CodeMeter License Central* verfügbar. Das Gateway ist in PHP geschrieben.

Warum ein Gateway?

Wie schon im Falle des Verkaufens, können Sie mit der *CodeMeter License Central Internet Edition* selbst entscheiden, ob Ihr Kunde das Web-Interface zum Abholen der Lizenz verwendet oder Sie dies über unsere SOAP-Schnittstelle selbst realisieren.

In der Regel wird dabei die *CodeMeter License Central* nicht direkt aus dem Internet heraus verfügbar sein, sondern aus Sicherheitsgründen im internen Netzwerk stehen. Damit ist ein direkter Zugriff per SOAP von außen – also aus der beim Kunden installierten Software heraus – nicht möglich. Sie benötigen ein Stück Software, das sich in der DMZ (Demilitarisierte Zone) befindet, die Anfragen von außen entgegennimmt und an die *CodeMeter License Central* weiterleitet. Dieses Stück Software wird als „Gateway“ bezeichnet.

Personalisierte Zusatz-Informationen

Wie schon der Connector kann auch ein Gateway mehr machen, als nur die Anfrage durchzureichen. Wenn Sie ein „1 zu 1 Marketing“ realisieren möchten, dann ist das Gateway der optimale Platz um Lizenzinformationen mit Werbebotschaften zu verknüpfen und an den Kunden zu übertragen.

Aber nicht nur Werbe-Botschaften, auch Software-Aktualisierungen können so individuell für jeden Kun-

den zur Verfügung gestellt werden, denn das Gateway kann auf die Lizenzinformationen des entsprechenden Kunden zugreifen und individuelle Angebote bzw. Aktualisierungen herausfiltern.

Abholen der Lizenz per Update-Datei

Egal, ob Sie die Lizenz per Browser oder per Gateway abholen, das Grundprinzip ist dasselbe. Von dem gewünschten *CmContainer* wird eine Remote Kontext-Datei erstellt. Diese wird gemeinsam mit dem Ticket an die *CodeMeter License Central* übertragen. Die *CodeMeter License Central* überprüft, ob das angegebene Ticket gültig ist, d.h. ob es existiert und noch nicht abgeholt wurde, erzeugt dann eine genau für diesen *CmContainer* passende Remote Update-Datei und schickt diese als Antwort zurück. Die Update-Datei wird dann in den *CmContainer* eingespielt. Bei der Verwendung des Web-Interfaces sorgt ein ActiveX Plug-In, bzw. ein Java Applet für die Erstellung der Remote Kontext-Datei und des Einspielen der Remote Update-Datei auf der Kundenseite. Wahlweise kann der Kunde die Remote Kontext-Datei auch von Hand erstellen und im Web-Interface hochladen. Dies ist vor allem dann sinnvoll, wenn der PC für den der *CmContainer* aktiv ist, nicht über einen direkten Zugang zum Internet verfügt. In diesem Fall werden weder ActiveX noch Java benötigt, noch nicht einmal die *CodeMeter Runtime*.

Das Standard Gateway

Bei der Benutzung eines Gateways erstellen Sie die Remote Kontext-Datei selbst, schicken diese gemeinsam mit dem Ticket an die *CodeMeter License Central* und spielen die erhaltene Remote Update-Datei ein.

Für den einfachen Start steht Ihnen in der *Internet Edition* ein Standard Gateway zur Verfügung. Dieses Gateway wird wahlweise per **HTTP/POST** oder per **HTTP/ GET** angesprochen und holt alle offenen Lizenzen des zugehörigen Tickets ab. Mit dem gleichen Mechanismus wird im Internet ein ausgefülltes Formular an den Server geschickt. Im Unterschied dazu liefert das Gateway keine HTML-Seite als Antwort, sondern die Remote Update-Datei.

Aufruf des Gateways

In vielen Programmiersprachen stehen Ihnen Klassenbibliotheken zur Verfügung, mit denen Sie einen HTTP-Request einfach abschicken können.

Remote Context- und Update-Datei

Zum Erstellen der Remote Context-Datei und zum Einspielen der Remote Update-Datei können Sie die Funktionen **CmGetRemoteContextBuffer** und **CmSetRemoteUpdateBuffer** aus dem *CodeMeter Kern-API* verwenden. Diese stehen Ihnen seit der Version 4.0 zur Verfügung.

Wohin mit dem Aufruf?

Wie oben beschrieben ist das Abholen der Lizenz über ein Gateway einfach zu realisieren. Doch wo ist der optimale Platz für den Aufruf? Innerhalb Ihrer geschützten Anwendung, innerhalb Ihrer Fehlerbehandlungs-DLL, die von der geschützten Anwendung heraus aufgerufen wird oder doch gleich in einer zusätzlichen Aktivierungsanwendung?

Je nach Ihrem Anwendungsfall kann jede der drei genannten Lösungen für Sie die optimale Lösung sein. Als flexibelste Lösung hat sich die zusätzliche Aktivierungsanwendung erwiesen. Wenn der Kunde schon eine Basisversion der Software besitzt und ein weiteres Modul freischalten möchte, dann können Sie die Aktivierungsanwendung aus Ihrer Software heraus starten. Besitzt der Kunde noch gar keine Lizenz, dann können Sie die Aktivierungsanwendung aus der Fehlerbehandlungs-DLL heraus starten.

Auch wenn Ihr Kunde eine Netzwerk Lizenz aktivieren möchte, bietet die zusätzliche Aktivierungsanwendung die optimale Lösung. Der Kunde benötigt auf dem Server nur die *CodeMeter Runtime* und Ihre Aktivierungsanwendung.

9.3.3.2 Depot-Interface

Mit Hilfe des Depot Interface können Lizenzen abgeholt werden. Das Abholen erfolgt durch das Hochladen einer Kontext Datei und das Herunterladen einer Update-Datei. Nach dem Einspielen der Update-Datei kann optional eine neue Kontext-Datei hochgeladen werden, um das Einspielen der Lizenz zu quittieren. Dieser Prozess kann natürlich in einem Schritt durchgeführt werden, so dass der Lizenznehmer nur „die Lizenz abholt“.

Das Depot Interface bietet Ihnen zwei Möglichkeiten um Lizenzen abzuholen:

- direkt (der PC mit dem zu programmierenden *CmContainer* verfügt über eine Internetverbindung.)
- indirekt (über Dateiaustausch werden die Freischaltdateien auf einen anderen PC übertragen.)

Neben dem Abholen von Lizenzen bietet das Depot Interface auch Methoden zum Zurückgeben von Lizenzen. Nach dem Zurückgeben einer Lizenz bekommt der Lizenznehmer ein neues Ticket. Dies erhält er erst nach dem Hochladen der Quittung. Mittels des neuen Tickets kann der Lizenznehmer die Lizenz dann auf einen anderen PC übertragen, bzw. kann er diese Lizenz auch Weiterverkaufen und dem Käufer das Ticket geben. Wenn Sie Weiterverkaufen erlauben möchten, dann schalten Sie das Zurückgeben von Lizenzen einfach ein. Standardmäßig ist das Zurückgeben von Lizenzen - damit auch das Weiterverkaufen - abgeschaltet. Wenn der Anwender Lizenzen zurückgeben möchte, dann sollten Sie den Kaufpreis erst nach dem Hochladen der Quittung erstatten

Im Depot Interface ist es zusätzlich möglich Informationen über die gekauften und aktivierte Lizenzen abzurufen.

In der Produktkonfiguration können Sie dem Lizenznehmer das Lizenzierungssystem (*CmDongle / CmActLicense*) vorgeben oder ihm die Wahl - ob Hardware oder Aktivierung – selbst überlassen.

Abholen der Lizenz durch den Lizenznehmer

Falls der Lizenznehmer die Lizenzen direkt abholen soll, dann benötigt er einen Zugriff auf die *CodeMeter License Central*. Je nach geplantem Zugriff, direkt per SOAP aus der Anwendung heraus oder über eine Webseite, stellen Sie dazu den Webserver oder Webserver und Application Server in die DMZ (Demilitarisierte Zone).

In diesem Fall ist es aus Sicherheitsgründen ratsam, die Installation auf mehrere Rechner zu verteilen und die restlichen Module (Datenbank und evtl. den Application Server) hinter der inneren Firewall zu positionieren.

Die Webseiten, die der Kunde zum Abholen der Lizenz angezeigt bekommt, können an Ihr Corporate Design angepasst werden.

In der *CodeMeter License Central Internet Edition* können Sie eine Nutzerverwaltung für den Lizenznehmer aktivieren, z.B. wenn er eine größere Firma ist und mehrere Mitarbeiter Lizenzen abholen sollen. Dann kann der Lizenznehmer selbst einsehen, welche Lizenzen von welchen Mitarbeitern abgeholt wurden. In diesem Fall identifiziert sich der Nutzer nicht mit der Ticketnummer, sondern mit seinem Account.

9.3.3.3 Admin-Interface

Das Admin Interface besteht aus den Teilen Lizenzkonfiguration, Auswertungen, Support und Benutzerverwaltung.

In der Lizenzkonfiguration verwalten Sie die Lizenzeigenschaften und die dazugehörigen Artikelnummern. Hier legen Sie für jede Lizenz individuell fest, welche Parameter fest programmiert werden und welche im Sales Interface übergeben werden können.

Im Statistik-Modul können Sie die Daten aus *CodeMeter License Central* auswerten, zum Beispiel: „Welcher Kunde hat welche Lizenzen in welchem *CmContainer*?“

Für das Abschließen von offenen Vorgängen (z.B. Quittung nicht hochgeladen), die Freigabe von weiteren Aktivierungen und das Bearbeiten von Blacklist-Einträgen steht Ihnen das Support-Modul zur Verfügung.

Die Benutzerverwaltung bietet Ihnen die Möglichkeit, die Zugriffsrechte auf die *CodeMeter License Central* zu konfigurieren. Die Authentifizierung kann über Benutzernamen und Passwort, IP-Adresse oder *CmContainer* erfolgen. So können Sie zum Beispiel festlegen, dass ein Vertriebspartner mit wechselnden IP Adressen sich mit dem *CmContainer* authentifizieren muss, während sich ein automatischer Sales-Connector über IP Adresse anmeldet.

9.3.4 Einsatz-Szenarien für die CodeMeter License Central

Der Einsatz der *CodeMeter License Central* kann zum Beispiel innerhalb der folgenden Szenarien erfolgen.

Nutzung	Beschreibung
am Einzelplatz	Hier ist die <i>CodeMeter License Central</i> als VM Image lokal auf einem Einzelplatz-Rechner installiert und läuft innerhalb des VMware Players oder der VMware Workstation. Über einen Browser greift der Nutzer auf die <i>License Central</i> zu. Der Vorteil hier: alle notwendigen Komponenten sind bereits installiert und die Datenbank-Verwaltung ist nicht notwendig.
im kleinen Netzwerk/ Intranet	Hier ist die <i>Desktop Edition</i> der <i>License Central</i> auf einem Server installiert und mehrere Mitarbeiter eines Unternehmens greifen über Browser auf die <i>License Central</i> zu. Der Vorteil hier: alle Mitarbeiter greifen auf eine zentrale Datenbasis zu.
mit Online-Anbindung	Hier ist die <i>Internet Edition</i> der <i>License Central</i> auf einem Server installiert. Das Gateway befindet sich in der DMZ. Der Kunde kann dabei aus der geschützten Anwendung heraus, die sich auf seinem PC befindet, über das Internet seine Lizenzen aktivieren oder freischalten lassen.
mit einem Online Shop	Hier ist die <i>Internet Edition</i> der <i>License Central</i> auf einem Server installiert. Über einen Connector, der sich in der DMZ befindet, kommunizieren der Online-Shop und die <i>License Central</i> miteinander. Der Vorteil hier: sie können webbasierte Lizenzgeneratoren gängiger Online Shops zur Erstellung von Tickets nutzen.
durch Anbindung an ein ERP-/CRM-System	Hier ist die <i>Internet Edition</i> der <i>License Central</i> auf einem Server installiert. Das ERP-/CRM-System ruft einen internen Connector auf, der die Daten verarbeitet und weiterleitet. Das so generierte Ticket wird dann an das ERP-/CRM-System zurückgeschickt. Der Vorteil hier: Lizenzinformationen können mit Kundendatenbanken, Rechnungs- oder Bestellwesen, etc. zusammengeführt werden.

9.4 Programmierung per Dateiaustausch

CmDongle-Lizenzen

Um aus der Ferne die Aktualisierung eines *CmDongle* vornehmen zu können, benötigt man einige Informationen über den *CmDongle*, der umprogrammiert werden soll. Diese Informationen werden sicher in einer Kontext-Datei, der *.WibuCmRaC-Datei (Lizenzanforderungsdatei), abgespeichert und transportiert.

*.WibuCmRaC-Datei - Lizenzanforderung

Die *.WibuCmRaC-Datei kann nur der erstellen, der physikalisch im Besitz des *CmDongle* ist. Bei der Erstellung wird der Firm Code angegeben, der enthalten sein soll. Üblicherweise gibt man also den eigenen Firm Code an, nur dessen Inhalt kann man auch verändern. Zusätzlich enthält die Datei die Seriennummer des *CmDongle*. Erhalten Sie als Lizenzgeber die *.WibuCmRaC-Datei von Ihrem Lizenznehmer, können Sie detailliert sehen, welche Ihrer Lizenzen und Lizenzoptionen sich aktuell im *CmDongle* befinden. Der Lizenznehmer erzeugt diese Datei beispielsweise im *CodeMeter Kontrollzentrum* über den Vorgang der Lizenzaktualisierung⁴⁵¹.

*.WibuCmRaU-Datei - Lizenzaktualisierung

Als Lizenzgeber können Sie nun mit *CodeMeter License Editor*, *CmBoxPgm* oder *CodeMeter License Central* auf Basis dieser *.WibuCmRaC-Datei eine sogenannte Update-Dateien (*.WibuCmRaU)-Datei (Lizenzaktualisierungsdatei) erzeugen, um die bestehenden Lizenzen zu verändern. Hierbei haben Sie dieselben Möglichkeiten wie bei einem physikalisch vorhandenen *CmDongle*. Sie können also neue Lizenzen hinzufügen, bestehende verändern (z.B. Ablaufdatum verlängern) oder auch löschen. Die *.WibuRaU-Datei enthält die Aktualisierungssequenzen und ist nur für diesen einen bestimmten *CmDongle* gültig. Sie kann vom Lizenznehmer genau einmal in genau den dafür bestimmten *CmDongle* eingespielt werden.

Firm Update Counter (FUC)

Nach dem erfolgreichen Einspielen der Update-Datei durch den Lizenznehmer wird ein Zähler, der Firm Update Counter (FUC), im Firm Item erhöht. Durch das Erhöhen des Zählers wird die *.WibuRaU-Datei für ein weiteres Einspielen ungültig.

Dies ist vor allem relevant, wenn die *.WibuCmRaU-Datei Programmierbefehle enthält, die z.B. einen weiteren Lizenzeintrag anlegen, einen Unit Counter um eine Anzahl an Einheiten erhöhen, oder eine Expiration Time um eine Anzahl an Tagen nach vorne setzen.

*.WibuCmRaM-Datei - modifizierte Kontext-Datei

Beim Erstellen der *.WibuCmRaU Datei wird automatisch auch eine *.WibuCmRaM-Datei erzeugt, mit der Sie ein Abbild des Inhalts besitzen, den Ihr Lizenznehmer hat, wenn er die *.WibuCmRaU-Datei einspielt. Ist irgendwann eine erneute Aktualisierung notwendig (z.B. weitere Verlängerung der Lizenz), können Sie sich entweder wieder eine neue *.WibuCmRaC-Datei vom Lizenznehmer zuschicken lassen, oder stattdessen für Ihre Programmierung die zuletzt erzeugte *.WibuCmRaM-Datei als Basis verwenden. Somit brauchen Sie normalerweise nur einmal eine *.WibuCmRaC-Datei von Ihrem Lizenznehmer, danach können Sie ihm alle weiteren Aktualisierungen einfach zuschicken. Viele Lizenzgeber erzeugen schon direkt nach der Programmierung im eigenen Haus die *.WibuCmRaC-Datei und können so die Aktualisierungen ohne das Zutun Ihrer Lizenznehmer erstellen.



Sollte der *CmDongle* zwischendurch von einem anderen Lizenzgeber umprogrammiert worden sein, behalten alle Dateien weiterhin ihre Gültigkeit.

Die folgende Abbildung illustriert diesen Vorgang:

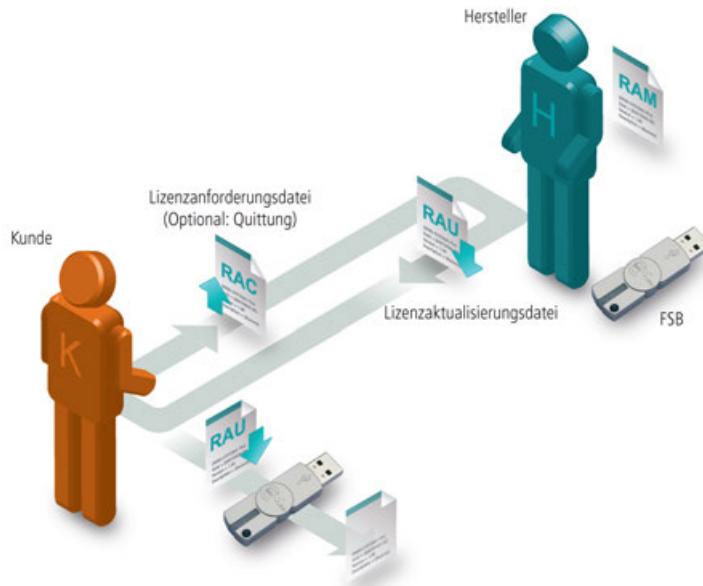


Abbildung 188: CmFAS - Dateibasierte Fernaktualisierung für *CmDongle*

CmActLicense-Lizenzen

Die Programmierung von *CmActLicense*-Lizenzen folgt mit zwei Ausnahmen in groben Zügen dem Prozess für *CmDongles* wie oben beschrieben.

Zum einen muss der Lizenznehmer vor Erstellung der initialen Kontext-Datei (*.wibuCmRaC- oder Lizenzanforderungsdatei) zunächst einen leeren Muster-Lizenzcontainer importieren, der ihm vom Lizenzgeber zugesandt wird. Diese LIF-Datei (License Information File) im *.wbb (WIBU Binary)-Format enthält Informationen über das [Bindungsschema](#)²⁸ und [zusätzliche Aktivierungsoptionen](#)²⁹ der *CmActLicense*-Lizenz, die verwendet werden, um die Lizenz eindeutig an den Rechner oder das Gerät binden zu können. D.h. Hardware-Merkmale eines Rechners oder eines Gerätes werden ermittelt und zusätzliche Aktivierungsoptionen übertragen. Erst auf dieser Grundlage erfolgt die Erstellung der ersten Lizenzanforderungsdatei. Im weiteren Verlauf wird dann auf Grundlage der Lizenzanforderungsdatei durch den Lizenzgeber diese Datei in eine Update-Datei (*.wibuCmRaU- oder Lizenzaktualisierungsdatei) umprogrammiert, die der Lizenznehmer wiederum importiert. Ab diesem Zeitpunkt ist der Datei-Austauschprozess zwischen Lizenzgeber und Lizenznehmer für *CmDongle*- und *CmActLicense*-Lizenzen der gleiche.

Zum anderen wird derzeit beim Umprogrammieren der Kontext-Datei in eine Update-Datei keine modifizierte Kontext-Datei (*.WiBuCmRaM-Datei) erzeugt.

Die folgende Abbildung illustriert diesen Vorgang:

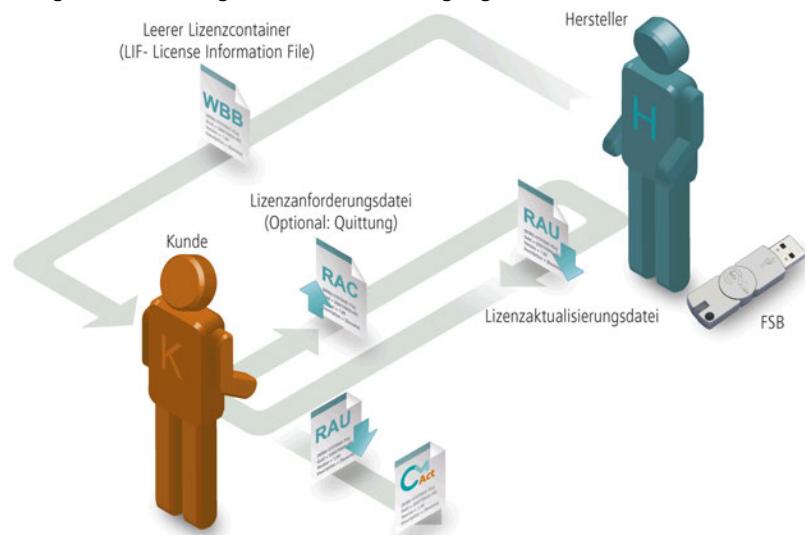


Abbildung 189: *CmActLicense* - Dateibasierte Fernaktualisierung für *CmActLicense*

10 Auslieferungsoptionen (Deployment)

Nachdem Sie Ihre Software erfolgreich geschützt und die Lizenzen in den *CmContainer* übertragen haben, erfolgt die Auslieferung an den Endkunden.

Keine separate Treiber-Installation bei CmDongle

Bei vielen anderen Herstellern wird der direkte Zugriff über einen separaten Kernel Treiber gesteuert, der bei Aktualisierungen mitgeliefert wird. Wibu-Systems geht hier andere Wege.

Hier kommen die jeweiligen betriebssystemeigenen USB-Treiber zum Einsatz, die über *CodeMeter Lizenzserver* mit dem *CmDongle* und der ausgelieferten Software kommunizieren.



i Eine separate Installation von Treibern für die unterschiedlichen Betriebssysteme ist nicht erforderlich.

Jeweilige Aktualisierungen der Betriebssysteme liefern diese Treiber automatisch mit. So müssen Sie nicht auf ein Update von Wibu-Systems mit aktuellen Treibern warten. Ihre unter Windows Vista geschützte Software läuft mit der gleichen *CodeMeter® Runtime-Umgebung* sofort auch unter Windows 7.

Empfehlung

i Wibu-Systems empfiehlt trotzdem die Verwendung der Installationspakete, die auf der Webseite erhältlich sind (www.wibu.de). Dies vermeidet Versionskonflikte, die aus der gleichzeitigen Installation von verschiedenen Produkten entstehen könnten, die *CodeMeter Lizenzserver* benötigen. Wibu-Systems stellt damit sicher, dass auch eine Software, die mit einer Vorgängerversion von *CodeMeter®* geschützt wurde, mit dem aktuellen *CodeMeter Lizenzserver* funktioniert. Außerdem beinhalten die Installationspakete einige zusätzliche Support-Werkzeuge.

Die Auslieferung Ihrer geschützten Anwendung durch [einfaches Kopieren der benötigten Bestandteile](#) ⁴⁰⁴ empfiehlt Wibu-Systems hingegen nicht. Durch das Kopieren in ein separates Applikationsverzeichnis können bei der Verwendung mehrerer *CodeMeter®*-geschützter Anwendungen auf dem Rechner Versionskonflikte auftreten.

i Wibu-Systems übernimmt in diesem Fall keine Verantwortung für Versionskonflikte während der Laufzeit.

Eine Ausnahme bildet die Auslieferung von Komplettsystemen, d.h. wenn kein anderer Hersteller den *CmDongle* nutzt und außer der eigenen Software keine anderen *CodeMeter®*-geschützten Anwendungen installiert sind. Beispielsweise bei vorinstallierten Rechnern eines Kassen- oder Brandmeldezentralen-

systems.

10.1 Installationspakete für Nicht-Windows Betriebssysteme

Die Mindestanforderung für die Auslieferung Ihrer geschützten Software besteht in der Installation der **CodeMeter® Runtime**. Wibu-Systems empfiehlt zur Installation die Verwendung der vorkonfigurierten Installationspakete für die verschiedenen Betriebssysteme.

Diese kompletten Pakete dürfen Sie als Software-Hersteller kostenlos an Ihre Endkunden weitergeben. Alternativ können Ihre Kunden die Pakete aber auch kostenfrei ohne Einschränkungen, ohne Anmeldung oder Passwort von der Wibu-Systems Webseite direkt im Anwenderbereich herunterladen (<http://www.wibu.com/de/anwendersoftware.html>) und installieren.

Die folgenden Installationspakete für Nicht-Windows Betriebssysteme sind verfügbar:



Mac

CodeMeter Runtime-Kit (Mac OS X ab 10.6) für PowerPC und Intel Prozessoren



Linux

RPM Pakete, z.B. für SuSe, Red Hat

CodeMeter Runtime 64-Bit – für Rechner auf AMD64-Basis

AxProtector/Java Runtime – für mit AxProtector geschützte Java-Programme

CodeMeter Runtime – enthält alle notwendigen Dateien für den Endanwender

DEB Pakete, z.B. für Debian, Ubuntu

CodeMeter Runtime 64-Bit – für Rechner auf AMD64-Basis

AxProtector/Java Runtime – für mit AxProtector geschützte Java-Programme

CodeMeter Runtime – enthält alle notwendigen Dateien für den Endanwender

CodeMeter Lite – Reiner Treiber-Installer, für Systeme ohne Oberfläche



Sun Solaris

CodeMeter Runtime für SPARC

CodeMeter 64-Bit Erweiterung für SPARCV9 (ab Version 4.10)

CodeMeter Runtime für i386

CodeMeter 64-Bit Erweiterung für AMD64 (ab Version 4.10)

10.2 Auslieferung für Windows Betriebssysteme

Die Mindestanforderung für die Auslieferung Ihrer geschützten Software besteht in der Installation der **CodeMeter® Runtime**. Wibu-Systems bietet zur Installation vorkonfigurierte Installationspakete³⁹⁷ für Windows an.

Diese Pakete dürfen Sie als Software-Hersteller kostenlos an Ihre Endkunden weitergeben. Alternativ können Ihre Kunden die Pakete aber auch kostenfrei ohne Einschränkungen, ohne Anmeldung oder Passwort von der Wibu-Systems Webseite direkt herunterladen und installieren. Ebenfalls ist die Verwendung einzelner Merge-Module³⁹⁷ möglich, die Dateien, Registrierungseinträge und Einstellungen bestimmter Runtime-Komponenten kapseln und die vom Setup-Entwickler für eigene Installer verwendet

werden können.

10.2.1 Vorkonfigurierte Installationspakete

Volumfähiges Installationspaket

Dieses Paket, das alle notwendigen Komponenten der *CodeMeter® Runtime* enthält, ist erhältlich für 32- und 64-Bit Betriebssysteme.

Es ist verfügbar als ausführbare Datei (*CodeMeterRuntime32/64.exe*) und als eigenständiges Paket zur Managed Software Installation über den Windows Installer-Dienst *msiexec.exe* (*CodeMeterRuntime32/64.msi*).

Installationspaket mit reduziertem Umfang

Dieses Paket, ebenfalls erhältlich für 32- und 64-Bit Betriebssysteme, besitzt einen reduzierten Funktionsumfang der *CodeMeter® Runtime*. Nicht enthalten in diesem Paket sind die relevanten Dateien des *CodeMeter Kontrollzentrums*, die separate Anwenderhilfe und die Einträge im Windows Startmenü (shortcuts).

Es ist verfügbar als ausführbare Datei (*CodeMeterRuntime32/64Reduced.exe*) und als eigenständiges Paket zur Managed Software Installation über den Windows Installer-Dienst *msiexec.exe* (*CodeMeterRuntime32/64Reduced.msi*).

- | |
|---|
|  Die ausführbare Datei des reduzierten Installationspaketes kann auf der Wibu-Systems Webseite nicht direkt im Anwenderbereich, sondern im Entwicklerbereich heruntergeladen werden. |
|  Wenn Sie das reduzierte Installationspaket verwenden, beachten Sie bitte, dass der <i>CmDust</i> -Eintrag des Startmenüs nicht mehr vorhanden ist. Die Erstellung der Protokolldatei muss dann alternativ über das Kommandozeilen-Werkzeug <i>cmu</i> angestoßen ⁵⁰⁰ werden. |

Vorkonfigurierte Installationspakete (Windows) für FSB-Verwendung

Dieses Paket erhältlich für 32- und 64-Bit Betriebssysteme enthält die *CodeMeter® Runtime* und das Modul *CmRuntimeInternal* mit FSB-Funktionalitäten. Damit lassen sich beispielsweise FSB-Lizenzserver im Netzwerk betreiben oder *CodeMeter®*-Verschlüsselungen in einer Entwicklungsumgebung (IDE) zur Verfügung stellen.

Es ist verfügbar als ausführbare Datei (*CodeMeterRuntimeLicensor32/64.exe*) und als eigenständiges Paket zur Managed Software Installation über den Windows Installer-Dienst *msiexec.exe* (*CodeMeterRuntime32/64Licensor.msi*).

CodeMeter® Merge-Module

Wibu-Systems bietet Ihnen für einzelne Komponenten der *CodeMeter® Runtime* auch Merge-Module an, die Sie in einen eigenen Installer einbauen können.

Diese *.msm-Dateien sind nicht eigenständig installierbar und kapseln Dateien, Registrierungseinträge und Einstellungen einzelner Komponenten.

Diese Module laden Sie aus dem passwortgeschützten Entwicklerbereich der Wibu-Systems Webseite herunter (<http://www.wibu.com/de/software-development-kit.html>).

Die folgenden Dateien sind Bestandteile der  Wibu-Systems *CodeMeter® Runtime* Distribution für Windows:

Datei	Merge-Modul
CmRuntimeMerger.msm	CodeMeter® Runtime (Win 32)
CmRuntimeMergerReduced.msm	CodeMeter® Runtime mit reduzierten Umfang (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	Enthält .NET policies

Die *CodeMeter® Runtime Merge-Module* beinhalten alle notwendigen Teile des *CodeMeter® Runtime Kit*, wie *CodeMeter Lizenzserver*, *CodeMeter Kontrollzentrum* und die Laufzeitbibliotheken.

Die Merge-Module *CmRuntimeMerger.msn* oder *CmRuntimeMergerReduced.msn* müssen auf jedem System installiert werden. Im reduzierten Merge-Modul sind nicht: relevante Dateien des *CodeMeter Kontrollzentrums*, die separate Anwenderhilfe und die Einträge im Windows Startmenü (shortcuts).

Das Merge-Modul *CmRuntimeMerger64.msn* wird für *CodeMeter®-Zugriffe* von 64-Bit-Anwendungen benötigt. Wenn selbst keine 64-Bit-Anwendung ausgeliefert wird, braucht das Modul nicht installiert werden.

Das Merge-Modul *CmUserHelp.msn* installiert die Benutzerhilfe auf das Zielsystem, was Ihren Kunden hilft, sich mit *CodeMeter®* vertraut zu machen.

Das Merge-Module *Wibu-ShellExtMerger32/64.msn* beinhalten u.a. die Erweiterung, per Doppelklick fernprogrammierte Aktualisierungsdateien auszuführen.

Das Merge-Modul *WibuCmNet.msn* wird bei der Auslieferung von .NET-Anwendungen benötigt. Es enthält die Referenzen des Global Assembly Cache (GAC).

Firewall-Einstellungen

CodeMeter® verwendet standardmäßig TCP/IP zur Kommunikation mit den geschützten Programmen und zur Darstellung des *CodeMeter WebAdmin*. Damit dies auch bei aktiverer "Windows Firewall" funktioniert, tragen die *CodeMeter® Runtime Merge-Module* im privaten wie öffentlichen Profil je eine Ausnahme für den *CodeMeter Lizenzserver* (*CodeMeter.exe*) ein. Beim 'mobilen' Einsatz von *CodeMeter®*, d.h. ohne Verwendung der Merge-Module prüft der *CodeMeter Lizenzserver* selbst auf einen Ausnahme-Eintrag im aktuellen Profil der Firewall und setzt diese, wenn sie nicht vorhanden ist. Dies aber nur, wenn *CodeMeter Lizenzserver* mit Administratorrechten gestartet wurde.



Firewall-Anwendungen anderer Hersteller außer Microsoft werden derzeit nicht unterstützt. Hier müssen ggf. die Ausnahmen manuell eingetragen werden.

10.2.2 Anpassungsoptionen für Installationspakte

In der Mehrzahl der Fälle sollten die vorkonfigurierten Installationspakte der *CodeMeter® Runtime* in Form von ausführbaren Dateien (*.exe), Windows Installationspaketen (*.msi) und Merge-Modulen (*.msm) den Auslieferungs- und Installationsanforderungen der mit *CodeMeter®* geschützten und lizenzierten Software erfüllen.

In Ausnahmefällen kann es aber erforderlich sein, die vorkonfigurierten Installationspakte weiter anzupassen.

Wibu-Systems bietet zu diesem Zweck mehrere Optionen an: [Installationsoptionen](#)³⁹⁹, [gezieltes Instal-](#)

lieren von Features⁴⁰⁰ und die [Verwendung von zentralen Konfigurationsparametern beim Einbinden von Merge-Modulen⁴⁰⁰](#) in eigene Installer.

Installationsoptionen

Im Fall von Windows Betriebssystemen haben Sie die Möglichkeit, beim Aufruf der ausführbaren *CodeMeter® Runtime-Installationspakte* über Angabe weiterer Parameter den Installationsvorgang zu konfigurieren.

Zur Auflistung bestehender Kommandozeilen-Optionen gehen Sie bitte wie folgt vor:

1. Geben Sie in einem offenen Eingabeaufforderungsfenster z.B. die folgende Kommandozeile ein:
`CodeMeterRuntime.exe /?`

Ein Fenster öffnet sich, das bestehende Optionen auflistet.

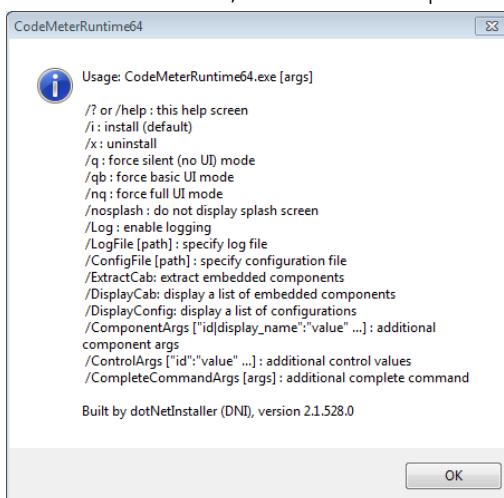


Abbildung 190: *CodeMeterRuntime.exe* - Kommandozeilen-Optionen

Bitte beachten Sie, dass die Optionen '/q, /qb, /nq' zur GUI-Anzeige-Steuerung und "stilen" Installation ab der *CodeMeter*-Version 5.0a obsolet sind.

Dies liegt an der Änderung des Standardverhaltens des EXE- und MSI-Installers.

Da der EXE-Installer (Bootstrapper) nun standardmäßig im "stillen" Modus startet, können Änderungen nur noch über die Adressierung des MSI-Installers erfolgen, der dafür standardmäßig im GUI-Modus startet.

Ab der *CodeMeter*-Version 5.0a wird die Kontrolle der GUI-Anzeige und "stilen" Installation daher über den Kommandozeilenbezeichner *ComponentArgs* umgesetzt.



Die Kommandozeilen-Eingabe

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

ermöglicht eine "stille" Installation ohne Eingaben des Benutzers.

Die Kommandozeilen-Eingabe

`CodeMeterRuntime.exe /ComponentArgs " * " : " /qn REINSTALLMODE=oemusv`

REINSTALL=ALL"

führt eine "stille" Reparatur-Installation durch.

Die Kommandozeilen-Eingabe

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

führt eine "stille" Deinstallation durch.

Gezieltes Installieren von Features

Über zusätzliche Optionen des Kommandozeilenbezeichners `ComponentArgs` ist es außerdem möglich, gezielt zu bestimmen, welche Features installiert werden sollen.

Dazu integrieren Sie die zu installierenden Features in den Bezeichner `ComponentArgs` nach folgenden Regeln:

- `ADDLOCAL` installiert die Features
- `REMOVE` entfernt bereits vorhandene Features
- Nennen der Feature ID-Namen
- Einzelne Feature-ID-Namen werden durch Komma getrennt.



Hinter `ADDLOCAL` werden alle zu installierenden Features aufgeführt. Features, die nicht aufgeführt werden, werden auch nicht installiert.

Die folgende Tabelle listet die Feature IDs des vollumfänglichen ausführbaren Installationspaketes auf:

Feature ID	Beschreibung
Complete	Haupt-Feature, enthält das <code>CmRuntimeMerger</code> -Modul und die folgenden Sub-Features
DotNET_Modules	Enthält das <code>WibuCmNet.msn</code> Modul
WibuShellExtension	enthält das <code>ShellExtensionMerger</code> -Modul
User_Help	enthält das <code>CmUserHelp</code> Merge-Modul



Die Kommandozeile-Eingabe:

```
CodeMeterRuntime64.exe /componentargs "*" : "/l*v Runtime_msi.log
```

```
ADDLOCAL=Complete,WibuShellExtension,User_Help"
```

installiert neben dem `CmRuntimeMerger`-Modul die Features `WibuShellExtension` und die `UserHelp` aber nicht die `.NET_Modules`.

Die folgende Tabelle listet die Feature IDs des reduzierten ausführbaren Installationspaketes auf:

Feature ID	Beschreibung
Complete	Haupt-Feature, enthält das <code>CmRuntimeMerger</code> -Modul und die folgenden Sub-Features
DotNET_Modules	Entspricht dem Merge-Modul <code>WibuCmNet.msn</code> und enthält die Datei <code>wibucmnet.dll</code> , die Sprachdateien und die policy files
WibuShellExtension	enthält das <code>ShellExtensionMerger</code> -Modul



Die Kommandozeile-Eingabe:

```
CodeMeterRuntime64Reduced.exe /componentargs "*" : "/l*v Runtime_msi.log
```

```
ADDLOCAL=Complete,DotNET_Modules"
```

installiert neben dem `CmRuntimeMergerReduced`-Modul zusätzlich nur das Feature `.NET-Module` und nicht das `ShellExtension`-Modul.

Einbinden der Merge-Module in Fremd-Installer über Konfigurationsparameter

Über die Einführung von zentralen Konfigurationsparameter ist es auch möglich, für die Merge-Module `CmRuntimeMerger` und `CmUserHelp` zu steuern, ob z.B. bei der Installation das *CodeMeter Kontrollzentrum* automatisch starten soll und ob Einträge im Windows Start Menü (Shortcuts) angelegt werden.

Dazu wurden die Parameter `PROP_CMCC` für das Startverhalten des *CodeMeter Kontrollzentrums* und `PROP_MAKESC` für das Anlegen von Shortcuts eingeführt.

CmRuntime Merger-Modul

Im Modul `CmRuntimeMerger` stehen die Parameter `PROP_CMCC` und `PROP_MAKESC` mit dem folgenden Verhalten und vordefinierten Werten zur Verfügung.

Parameter `PROP_CMCC`

Wert	Beschreibung
None	<ul style="list-style-type: none"> Verhindern des Starts von <i>CodeMeter Kontrollzentrum</i> am Ende der Installation Abschalten des <i>CodeMeter Kontrollzentrum</i>-Eintrages im Autostart-Ordner
run	Abschalten des <i>CodeMeter Kontrollzentrum</i> -Eintrages im Autostart-Ordner
auto	Verhindern des Starts von <i>CodeMeter Kontrollzentrum</i> am Ende der Installation
all	<ul style="list-style-type: none"> Starten von <i>CodeMeter Kontrollzentrum</i> am Ende der Installation Erzeugen des <i>CodeMeter Kontrollzentrum</i>-Eintrages im Autostart-Ordner
	 Ist der Parameter <code>PROP_CMCC</code> nicht gesetzt, so entspricht dies dem Wert <code>all</code> .



Verhindern des Starts von *CodeMeter Kontrollzentrum* am Ende der Installation:
`CodeMeterRuntime64.exe /componentargs "*" : "/l*v rtk_install.log
 PROP_CMCC= "auto" "`

Parameter `PROP_MAKESC`

Wert	Beschreibung
no	Verhindern der Erzeugung jeglicher Shortcuts (<i>CodeMeter Kontrollzentrum</i> , <i>User Help</i> , <i>CmDust</i> usw.)  Bitte beachten Sie, dass der <i>CmDust</i> -Eintrag des Startmenüs dann nicht mehr vorhanden ist. Die Erstellung der Protokolldatei muss dann alternativ über das Kommandozeilen-Werkzeug cmu ⁶⁰² erzeugt werden.
yes	Erzeugen jeglicher Shortcuts (<i>CodeMeter Kontrollzentrum</i> , <i>User Help</i> , <i>CmDust</i> usw.)  Ist der Parameter <code>PROP_MAKESC</code> nicht gesetzt, so entspricht dies dem Wert <code>yes</code> .



Verhindern, dass bei der Installation des *CodeMeter® Runtime Kit* Shortcuts angelegt werden:
`CodeMeterRuntime64.exe /componentargs "*" : "/l*v rtk_install.log
 PROP_MAKESC= "no" "`

	Wird das <code>CmRuntimeMerger</code> -Modul sowohl über <code>PROP_CMCC</code> als auch über <code>PROP_MAKESC</code> gesteuert, dann verhindert der Wert <code>PROP_MAKESC= "no"</code> auch den Autostarteintrag, da dieser auch ein Shortcut ist.
---	---

CmUserHelp-Modul

Im Modul `CmUserHelp` steht der Parameter `PROP_MAKESC` mit dem folgenden Verhalten und vordefinierten Werten zur Verfügung.

Parameter `PROP_MAKESC`

Wert	Beschreibung
<code>no</code>	Verhindern der Erzeugung des Startmenü-Eintrages der <i>User Help</i> .
<code>yes</code>	Abschalten des <i>User Help</i> -Eintrages im Startmenü
	Ist der Parameter <code>PROP_MAKESC</code> nicht gesetzt, so entspricht dies dem Wert <code>yes</code> .

10.3 Installation mobil auf dem CmDongle (Windows)

Da die *CodeMeter®*-Technologie neben dem Kopierschutz-Bestandteil (*CodeMeter® Chip*) zusätzlich mit dem Flash-Speicher die Möglichkeit bietet, Anwendungen direkt ohne vorherige Treiberinstallation vom *CmDongle* zu starten, haben Sie die Option, Ihre Software direkt auf dem Dongle auszuliefern, ohne auf den gewohnten Softwareschutz verzichten zu müssen.

Um *CodeMeter®* portabel zu nutzen, benötigen Sie lediglich den *CodeMeter Lizenzserver* (`CodeMeter.exe`) aus dem Verzeichnis [%Program Files%\CodeMeter\Runtime\bin].

Kopieren Sie diese Datei zusammen mit Ihrer geschützten Anwendung in dasselbe Verzeichnis auf den Flash-Speicher eines *CmDongle*. Beim Start der geschützten Anwendung wird nun `CodeMeter.exe` automatisch gestartet und die Anwendung kann mit *CodeMeter Lizenzserver* kommunizieren.

Um Ihren Kunden den kompletten Funktionsumfang von *CodeMeter®* zur Verfügung zu stellen, sollten Sie die folgenden Dateien in das Anwendungsverzeichnis Ihrer Applikation auf den *CmDongle* kopieren:

Datei	Beschreibung
<code>CodeMeter.exe</code>	<i>CodeMeter Lizenzserver</i>
<code>CodeMeter.1**</code>	Sprachdateien für den <i>CodeMeter Lizenzserver</i>
<code>CodeMeterCC.exe</code>	<i>CodeMeter Kontrollzentrum</i> inklusive der Support-Anwendung <i>CmDust</i> .
<code>CodeMeterCC_**.qm</code>	Sprachdateien für das <i>CodeMeter Kontrollzentrum</i>
<code>CodeMeter**.wbb</code>	<i>CodeMeter WebAdmin</i>
<code>WibuCm32.dll</code>	<i>CodeMeter®</i> Laufzeitbibliothek (aus %Windows%\system32)
<code>WibuCm32.1**</code>	Sprachdateien für die Laufzeitbibliothek (aus %Windows%\system32)

 Bei der mobilen Benutzung legen Sie zusätzlich zur *CodeMeter® Runtime* eine `CodeMeter.ini`-Datei auf Ihren *CmDongle*. In diesem Fall werden alle Einstellungen aus der `CodeMeter.ini`-Datei gelesen und in diese geschrieben.

Somit bleiben keine Rückstände auf der Festplatte, oder in der Registry des PC.

Die Konfigurationsdatei `CodeMeter.ini`

Die Konfigurationsdatei `CodeMeter.ini` beinhaltet alle Einstellungen des *CodeMeter Lizenzservers*.

Um eine `CodeMeter.ini` mit rechnerunabhängigen Standardwerten zu erstellen, legen Sie im gleichen Verzeichnis wie `CodeMeter.exe` eine leere Datei mit dem Namen `CodeMeter.ini` an.

Beim erneuten Starten der `CodeMeter.exe` werden die Standardwerte in die Datei geschrieben. Sämt-

liche Konfigurationsänderungen, die Sie nun im *CodeMeter Kontrollzentrum* oder im *CodeMeter WebAdmin* vornehmen, werden automatisch in der *CodeMeter .ini* gespeichert.

Was passiert, wenn nun auf dem PC *CodeMeter®* bereits installiert ist? Dies stellt kein Problem dar. Wenn *CodeMeter Lizenzserver* installiert ist und bereits läuft, dann wird dieser genommen. Alle automatischen Mechanismen zum Starten oder Beenden von *CodeMeter Lizenzserver* werden außer Kraft gesetzt.

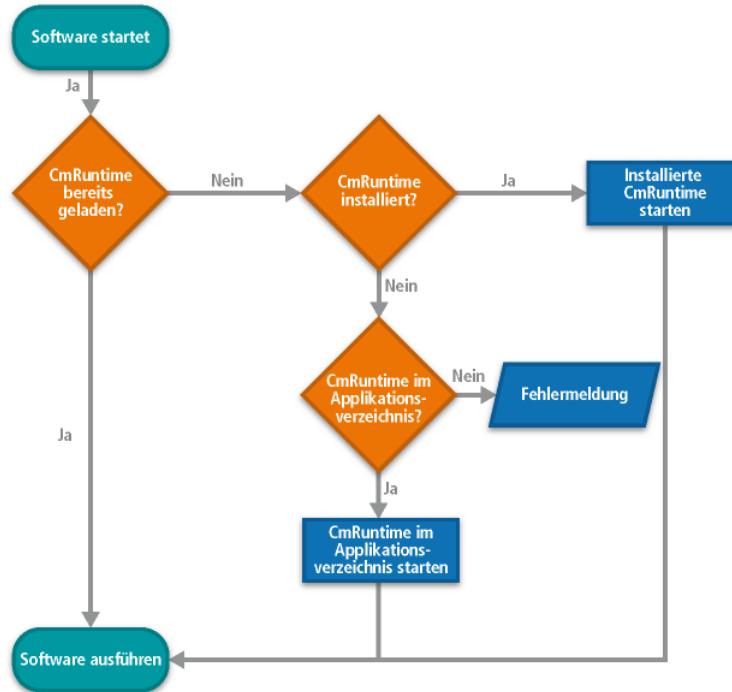


Abbildung 191: Verhalten CodeMeter Lizenzserver

UseMobileHandling

Das Beenden von *CodeMeter Lizenzserver* ist ab Version 4.0 automatisiert. Tragen Sie in die *CodeMeter .ini*⁴⁰²-Datei dazu "UseMobileHandling=1" ein. Mit diesem Eintrag beenden sich *CodeMeter Lizenzserver* sowie das *CodeMeter Kontrollzentrum* automatisch mit Ihrer Anwendung. Sollen mehrere Anwendungen gleichzeitig gestartet sein und auf *CodeMeter Lizenzserver* zugreifen, dann beendet sich *CodeMeter Lizenzserver* natürlich erst beim Beenden der letzten Anwendung.

```
[General]
ExePath=${CODEMETER_HOME}
CleanupTimeout=120
UDPPoolingTime=1000
UDPCachingTime=20
ApiCommunicationMode=2
ISNetworkServer=0
NetworkAccessSmb=0
NetworkPort=22350
NetworkTimeout=100
MaxMessageLen=67108864
BindAddress=0.0.0.0
UseMSDA=1
CMACEnabled=0
UseMobileHandling=1
```

Abbildung 192: Auszug codemeter.ini

Shared Memory Modus

Der *CmDongle* wird nicht über TCP/IP angesprochen, sondern die Kommunikation findet über Shared Memory statt. Dann funktioniert der *CmDongle* auch in Fällen, in denen TCP/IP abgeschaltet ist. Dies entspricht der Standardeinstellung bei mobiler Installation).

Tragen Sie in der [CodeMeter.ini](#)⁴⁰²-Datei dazu einen "ApiCommunicationMode=2"-Eintrag ein.



Wibu-Systems empfiehlt bei der mobilen Installation diese Einstellung zu setzen.

10.4 Kopieren der CodeMeter Runtime ohne Installation unter Windows

Wibu-Systems empfiehlt nicht die Auslieferung Ihrer geschützten Anwendung durch einfaches Kopieren der benötigten Bestandteile. Durch das Kopieren in ein beliebig separates Applikationsverzeichnis können bei der Verwendung mehrerer CodeMeter®-geschützter Anwendungen auf dem Rechner Versionskonflikte auftreten.

Eine Ausnahme bildet die Auslieferung von Komplettsystemen, d.h. wenn kein anderer Hersteller den *CmDongle* nutzt und außer der eigenen Software keine anderen CodeMeter®-geschützten Anwendungen installiert sind. Beispielsweise bei vorinstallierten Rechnern eines Kassen- oder Brandmeldezentralensystems.

In begründeten Einzelfällen kann die Installation der CodeMeter® Runtime auch durch Kopieren einzelner Bestandteile vorgenommen werden. Die nachfolgende Tabelle zeigt einen Überblick über Bestandteile, Status und Beschreibung der Dateien.



Bitte beachten Sie, dass bei einer Nichtinstallation einzelner Bestandteile bestimmte essentielle Bedienmöglichkeiten und Funktionen nicht mehr verfügbar sind.

Bestandteil	Status	Beschreibung
CodeMeter.exe	notwendig	Ausführbares Programm (executable) des CodeMeter Lizenzservers Kann bei ausreichenden Rechten mit der Option /i als Dienst implementiert werden.
CodeMeter.l**	optional	Sprachdateien für CodeMeter.exe Beim Weglassen aller Sprachdateien ist die Standardsprache englisch verfügbar.
CodeMeterCC.exe	empfohlen	Ausführbares Programm (executable) des CodeMeter Kontrollzentrums

Bestandteil	Status	Beschreibung
CodeMeterCC**.qm	optional	Sprachdateien für <i>CodeMeter Kontrollzentrum</i> Bei Nichtinstallation aller Sprachdateien ist die Standardsprache englisch verfügbar.
cmu32(64).exe	empfohlen	Ausführbares Programm (executable) des cmu-Kommandozeilenprogramm inklusive der Support-Anwendung <i>CmDust</i> . Bei Nichtinstallation sind keine kommandozeilenbasierten <i>CodeMeter Kontrollzentrum</i> -Funktionen verfügbar. Bei Nichtinstallation kann nicht auf Informationen durch <i>CmDust</i> zurückgegriffen werden was den Support erheblich einschränkt.
CodeMeterXX.wbb	empfohlen	<i>CodeMeter WebAdmin</i> in verschiedenen Sprachvarianten.
WibuCm32(64).dll	empfohlen	Beinhaltet <i>CodeMeter® API</i> Funktionen, z.B. für die Support-Anwendung <i>CmDust</i> . Der übliche Installationspfad ist [%\Windows\System32].
WibuCm32(64).1XX	optional	Sprachdateien für die <i>WibuCm32(64).dll</i> ; Installationspfad:[\Windows\System32]. Bei Nichtinstallation aller Sprachdateien ist die Standardsprache englisch verfügbar.
WibuCmTrigger32(64).dll	optional	Wird von Microsoft Internet Explorer benötigt, z.B. beim Online-Abholen von Lizenen von der <i>CodeMeter License Central</i> .
WibuCmTrigger32(64).1XX	optional	Sprachdateien für die <i>WibuCmTrigger32(64).dll</i> . Bei Nichtinstallation aller Sprachdateien ist die Standardsprache englisch verfügbar.

11 Erweiterte CodeMeter Eigenschaften

Die folgenden Teile des Entwicklerhandbuchs beschreiben zusätzliche Eigenschaften des Schutz- und Lizenzierungssystems *CodeMeter*®.

11.1 Die Implicit Firm Item (IFI) Ebene

Die Implicit Firm Item-Ebene im *CmContainer* verhält sich in der Regel genauso wie die üblichen Ebenen (Firm Items). Er weist lediglich einige Besonderheiten auf.

Firm Code 0

Während sich alle anderen Ebenen durch das Vorhandensein eines exklusiven Firm Codes auszeichnen, der für jeden Lizenzgeber einzigartig ist, besitzt die Implicit Firm Item Ebene den Firm Code 0.



Dies bedeutet, dass jeder Besitzer eines *CmContainers* für die Implicit Firm Item -Ebene Lizenzgeber-Eigenschaften besitzt, und damit nicht nur lesend, sondern auch schreibend auf "seine" Ebene, das Implicit Firm Item zugreifen kann.

Daher bietet sich die Ebene des Implicit Firm Items an, dort Anwendungen zu hinterlegen, auf die jeder Besitzer eines *CmContainers* zugreifen kann.



Für OEM-Produkte gilt, dass die Product Codes bis 1000 für Wibu-Systems reserviert sind. Wollen Sie als Software-Hersteller bei Wibu-Systems kostenlos Lizenzeinträge (Product Items) im Implicit Firm Item-Lizenzcontainer reservieren, dann kontaktieren Sie bitte Wibu-Systems.

Kennwort statt Firm Security Box

Der schreibende Zugriff weist auf der Implicit Firm Item-Ebene Besonderheiten auf, da anstelle der Firm Security Box - wie bei anderen Ebenen üblich - hier das *CmStick*-Kennwort, der sogenannte User Individual Key (UIK) verwendet wird.

11.2 Enabling

Das *CodeMeter*® Feature Enabling erlaubt über die Verwendung eines Zugriffscodes, den gesamten *CmContainer* oder auch einzelne Firm Items oder Lizenzeinträge zu aktivieren oder zu deaktivieren.

Bezieht sich das Enabling auf die Implicit Firm Item (IFI)-Ebene kann der ganze *CmContainer* aktiviert oder deaktiviert werden. Das Enabling lässt sich aber auch für die gesamten Ebene Ihres Firm Items oder Ihre Lizenzeinträge, die Product Items umsetzen.

Zusätzlich können Sie das Aktivieren temporär definieren: dann erlaubt das Temporary Enabling den Zugriff auf den *CmDongle*, einzelne Firm Item-Ebenen oder Lizenzeinträge nur solange der *CmDongle* angesteckt ist und mit Strom versorgt wird.



Sie können nur die Implicit Firm Item-Ebene und eigene Firm Item-Ebene oder Lizenzeinträge aktivieren oder deaktivieren. Ebenso wenig kann ein anderer Hersteller das Enabling auf Ihre Firm Item-Ebenen oder Lizenzeinträge anwenden.

Das geregelte Aktivieren und Deaktivieren über das Enabling umfasst das Zusammenspiel zweier Bestand-

teile:

- Ein-/Ausschalter (Enabling Blöcke) und
- Zuordnungen (Lookup) zwischen Enabling Blöcken, Firm Item-Ebenen oder Lizenzteinträgen.

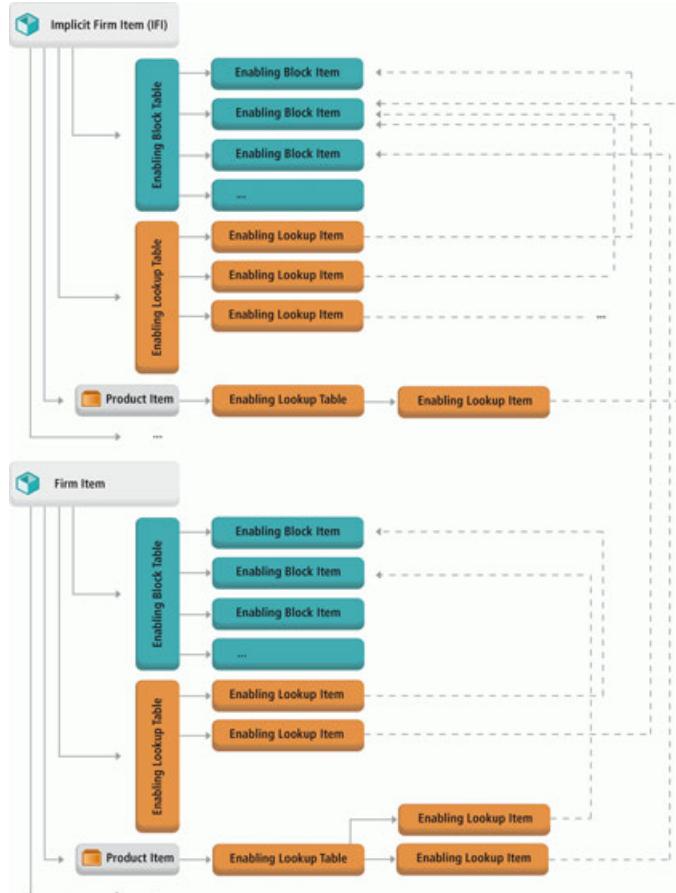


Abbildung 193: Enabling-Struktur im *CmContainer*

Einzelne Enabling Blöcke (Ein-/Ausschalter) und Lookups (Zuordnungen) werden in Tabellen zusammenfasst. Die obige Abbildung zeigt die Lage und Bezüge dieser verschiedenen Bestandteile in einem *CmContainer*.

11.2.1 Enabling Blocks als Ein- und Ausschalter

Ein Enabling Block entspricht einer Art Ein/Aus-Schalter, der Firm Item-Ebenen und Lizenzeneinträge aktiviert oder deaktiviert.

Ein Enabling Block wird immer als Ganzes aktiviert oder deaktiviert. Er ist als Eintrag in einer Tabelle vorhanden, die bis zu 31 Einträge enthalten kann. Diese Enabling Block Table kann sich auf der Ebene des IFI oder einer Firm Item-Ebene befinden.

Die Parameter, die in einem Enabling Block-Tabelleneintrag gesetzt werden, umfassen Angaben über:

- den Zugriffscode (Enabling Access Code)
- die Zugriffsart (Simple PIN oder Time PIN).

Bei der Zugriffsart Time PIN stehen zusätzlich die Optionen eines beschreibenden Textes (Enabling Text) und die Angabe einer zeitlichen Gültigkeit der Aktivierung zur Verfügung (Disable Time).

- den Aktivierungsmodus (Enabling Mode mit den Modi Enabled, Disabled oder temporary Enabled).

11.2.1.1 Zugriffscode - Enabling Access Code

Der Zugriffscode authentifiziert den Zugriff auf einen Enabling Block.

Dieser Enabling Access Code leitet sich aus der Eingabe des Access Keys als frei wählbarem Schlüssel (Abfolge von Zeichen) beim Anlegen eines Enabling Blocks ab. Er wird weiterhin benötigt, wenn ein Bestandteil des Enabling Blocks geändert wird.

Dieser Access Key wird von einem Hash-Algorithmus in eine Abfolge von 16 Bytes umgewandelt; das Ergebnis wird als Enabling Access Code (EAC) bezeichnet. Dieser Code wird im Enabling Block selbst gespeichert. Es können auch direkt die 16 Byte Enabling Access Code eingegeben werden.

Für jeden Vorgang, der die Einstellungen in einem Enabling Block ändert, muss der Benutzer den Access Key erneut angeben. Dieser wird vom selben Hash-Algorithmus dann in den Enabling Access Code umgewandelt und mit dem gespeicherten Enabling Access Code im *CmContainer* verglichen.

11.2.1.2 Zugriffsart - Simple oder Time PIN

Die Zugriffsart ist entweder über die Definition einer Simple PIN, die lediglich den Enabling Access Code (EAC) umfasst, geregelt oder über eine Time PIN, die zusätzlich Angaben in Form eines beschreibenden Textes enthält sowie eine Zeitdauer für das Enabling festlegt.

Enabling Text

Der Enabling Text ermöglicht bei der Zugriffsart Time PIN die Beschriftung eines Enabling Blocks.

Disable Time

Die Disable Time definiert einen Zeitpunkt, zu dem ein Enabling Block automatisch deaktiviert (disabled) wird. Sie stellt damit eine Art Ablaufdatum dar.

Die Disable Time ist sekundengenau und kann Werte bis zum 31. Dezember 2099, 23:59:59 annehmen.

Die Disable Time wird im *CmContainer* mit der *System Time* verglichen (zur Synchronisierung der einzelnen Zeitangaben siehe [hier](#)⁴¹⁷). Ist die System Time neuer als die Disable Time, wird der gesamte Enabling Block automatisch deaktiviert.



Sie können auch bestimmen, dass kein Ablaufdatum für einen Enabling Block gesetzt wird. Verwenden Sie dann den Parameter Disable Time=Never.

11.2.1.3 Aktivierungsmodus - Enabling Mode

Der Aktivierungsmodus gibt über verschiedene Statuszustände an, ob ein Enabling Block als Ganzes permanent aktiviert, deaktiviert oder temporär aktiviert wird (Enabling Mode).

Permanent Enabling-Status

Der Permanent Enabling-Status legt fest, ob ein gesamter Enabling Block ein- oder ausgeschaltet wird.

Dieser Status kann auf enabled (aktiviert), disabled (deaktiviert), oder temporary enabled gesetzt werden.

Temporary Enabling-Status

Der Temporary Enabling-Status aktiviert oder deaktiviert einen gesamten Enabling Block in Abhängigkeit zur Stromversorgung des CmDongles. Er steht nach erfolgter Aktivierung auf enabled (aktiviert) solange der CmDongle mit Strom versorgt wird.

Wird der CmDongle abgezogen und neu angesteckt, ist für den Zugriff erneut die Eingabe des Zugriffscodes notwendig. Dies entspricht auf der Implicit Firm Item Ebene der Option "Aktiviert solange angegeschlossen", die im [CodeMeter Kontrollzentrum](#) □⁴⁴⁹ gesetzt werden kann.

Ab CodeMeter® Version 4.30 und Firmware Version 1.18 kann das Temporary Enabling auf alle Enabling Blocks angewendet werden.



Bis CodeMeter® Version 4.30 und Firmware Version 1.18 gibt es das Temporary Enabling eines Enabling Blocks ausschließlich auf der Ebene des Implicit Firm Item (IFI). Man kann aber zur Nutzung des Temporary Enabling auch Verweise aus einem Lizenzcontainer oder einem Lizenzeintrag auf Enabling Blocks im IFI setzen. Die Verweisrichtung ist dabei nur in diese Richtung möglich. Aus dem IFI heraus kann nicht auf Enabling Blocks an anderer Stelle verwiesen werden.

Die Beziehung zwischen Permanent Enabling, Temporary Enabling und Disable Time

Für die Beziehung der Bestandteile Permanent Enabling-Status, Disable Time und Temporary Enabling-Status in einem CmContainer zueinander gilt:

- ist eine Disable Time abgelaufen, so wird immer der gesamte Enabling Block deaktiviert,
- ist der Temporary Enabling-Status aktiviert, so überschreibt er den Permanent Enabling-Status.

11.2.1.4 Löschen und Bearbeiten von Enabling Blocks

Zum Löschen eines existierenden Enabling Blocks wird nicht der Enabling Access Code benötigt, sondern in Abhängigkeit, wo dieser gespeichert ist:

- das CmDongle Kennwort im Falle der Implicit Firm Item-Ebene sowie
- die Firm Security Box im Rest der Fälle.



Der Enabling Access Code regelt beim Enabling die Zugriffsoptionen, nicht aber die Sicherheit.



Ein Enabling Block kann nur gelöscht werden, wenn keine Zuordnung auf ihn besteht, also keine Enabling Lookup-Einträge (siehe unten) auf ihn verweisen.

11.2.2 Zuordnung (Lookup) von Enabling Blocks

Ein Enabling Block ist nicht direkt mit einem Lizenzcontainer oder Lizenzeintrag verknüpft. Vielmehr findet die Bindung zwischen einem Enabling Block und dem Lizenzcontainer oder Lizenzeintrag über einen Zuordnungsprozess statt.

Dieser Prozess wird als Lookup bezeichnet. Hierfür steht eine sogenannte Lookup Table zur Verfügung. Bis zu 31 einzelne Bindungen können in dieser Tabelle zusammengefasst werden.

Wahlweise können Einträge dieser Tabelle an einen Enabling Block gebunden oder von ihm gelöst werden. Dieses Verfahren wird als Attaching bzw. Detaching bezeichnet.



Eine Enabling Lookup Table kann sich auf den Ebenen des Implicit Firm Items (IFI), der Lizenzcontainer und der Lizenzinträge befinden.



Innerhalb einer Lookup-Tabelle darf ein Lookup-Eintrag nur eine Zuweisung zu einem bestimmten Enabling Block haben. Mehrfache Zuweisungen sind untersagt. Wird bei der Zuweisung ein Enabling Block doppelt verwendet, so wird der bestehende Eintrag überschrieben.

Die Parameter, die für einen Lookup Table-Eintrag gesetzt werden, umfassen neben der Adressierung des Lizenzcontainers oder des Lizenzintrags:

- das Setzen von Zugriffsberechtigungen im aktivierte und deaktivierten Zustand eines Enabling Blocks (Enabling Level),
- das Definieren eines Kennzeichners, der bestimmt, ob das Aktivieren oder Deaktivieren eines Enabling Blocks zwingend erfüllt sein muss, oder nicht (required Flag).

11.2.2.1 Berechtigungsebenen - Enabling Level

Die Berechtigungsebenen definieren abgestufte Berechtigungen für Vorgänge, die aktivierte bzw. deaktivierte Lizenzcontainer oder des Lizenzintrags betreffen. Die folgenden Berechtigungsebenen (Enabling Levels) existieren:

Enabling Level	Berechtigung
Locate	Zulässige Vorgänge der Ebene Locate erlauben lediglich das Lesen des Firm und des Product Codes, aber keiner weiteren Informationen.
Read	Die Ebene Read erlaubt das komplette Lesen aller nicht versteckten Informationen auf der Ebene der Lizenzinträge. Erlaubt sind nicht: das Verschlüsseln, das Entschlüsseln, die Authentifizierung, oder die Berechnung eines Public Key aus einem gespeicherten Private Key durch den adressierten Lizenzcontainer oder die adressierten Lizenzinträge.
Encrypt	Die Ebene Encrypt erlaubt das Verschlüsseln, die Authentifizierung und die Berechnung eines Public Keys – aber nur wenn durch den Verschlüsselungsvorgang kein Begrenzungszähler (Unit Counter) verringert und kein Firm Access Counter auf der Ebene des Lizenzcontainers geändert werden müssen.

Enabling Level	Berechtigung
	 Diese Ebene sollte gesetzt werden, wenn der Benutzer den Stand des Unit Counter behalten soll, der im CmDongle gespeichert ist.
UnitUse	Die Ebene UnitUse erlaubt, ungehindert zu ver- oder entschlüsseln, zu authentifizieren und Public Keys zu berechnen, aber ohne spezielle Lizenzcontainer (Firm Items) oder Lizenzinträge (Product Item) hinzuzufügen, zu aktualisieren oder zu löschen.  Diese Ebene bietet Schutz vor ungewolltem und unberechtigtem Modifizieren von lokalen Inhalten.
Modify	Zulässige Vorgänge der stärksten Ebene Modify erlauben auch das Modifizieren auf der Ebene der Lizenzcontainer und Lizenzinträge. Weitere Einschränkungen bestehen nicht.
	 In der Standardeinstellung besitzen Lizenzcontainer (IFI, Firm Item) oder Lizenzinträge (Product Item) keine Zuordnungen zu einem Enabling Block über das Enabling Lookup. In diesem Fall liegt keine Einschränkung in der Benutzung dieses Items vor. Dies ist identisch mit der Berechtigungsebene modify .

11.2.2.2 Der Pflicht-Kennzeichner - Required Flag

Eine Zuordnung von Enabling Blocks über Einträge in Lookup-Tabellen kann gleichzeitig für mehrere Bindungsziele, d.h. verschiedene Lizenzcontainer oder Lizenzinträge, vorgenommen werden.

Das Setzen des Pflicht-Kennzeichners dient in diesem Fall dazu, beim Aktivieren oder Deaktivieren und den damit verbundenen Berechtigungsebenen mögliche Konflikte bei mehreren vorhandenen Bindungszielen zu vermeiden.

Ist mindestens ein Pflicht-Kennzeichner im Fall von mehreren Bindungszielen gesetzt, so bestimmt eine logische UND-Verknüpfung, dass alle Einstellungen der Bindungen, die einen Pflicht-Kennzeichner besitzen, zutreffen müssen bevor über eine definierte Operation auf den gesamten *CmContainer*, einen Lizenzcontainer oder einen Lizenzintrag zugegriffen darf.



Dies entspricht der Standardeinstellung seit der Firmware-Version 1.18. Beim Binden eines Enabling Blocks über einen Eintrag in einer Lookup Table ist das Required Flag als Standard gesetzt.

Sie können beim Programmieren des Zuordnungsprozess zwar explizit den Kennzeichner als nicht erforderlich setzen (NonRequired Flag), dies hat aber keine Auswirkungen, da in der Standardeinstellung NonRequired Flags durch eine logische ODER-Verknüpfung ignoriert werden, sobald mindestens ein Pflicht-Kennzeichner vorliegt. Dies liegt in den Einstellungen des globalen, den ganzen *CmContainer* betreffenden Enablings begründet.

Wollen Sie für eigene Zwecke das globale Enabling ändern, kontaktieren Sie Wibu-Systems Support.

Die nachfolgende Übersicht zeigt Ihnen, welche CodeMeter®-Werkzeuge und Schnittstellen Sie für das Enabling benötigen.

Enabling Block-Optionen	
CmBoxPgm <small>379</small>	Anlegen, Ändern und Löschen von Enabling Blöcken
Kern-API <small>334</small>	Für Enabling-Optionen

Enabling Block-Optionen

[Programmier-API](#)  ³³⁶

Aufruf der entsprechenden Klassen

11.2.3 Enabling-Beispiel

Das folgende kleine Beispiel soll das Konzept des Enabling grundlegend beschreiben.

Anforderung

Wir schützen eine Editor-Basisanwendung mit *AxProtector*, in der über einen Menüpunkt die neue Funktion zum Ändern des Schrift-Fonts hinzugefügt wurde. Diese Funktion soll nur ausgewählten Anwendern zur Verfügung gestellt werden und keine zeitliche Begrenzung besitzen; die anderen Anwender sollen den Editor weiter wie bisher nutzen können.

Beim Verwenden der neuen Funktion zum Abändern des Fonts soll sich ein Dialog öffnen, der nach Eingabe eines Passwortes, das dem Enabling Access Code entspricht, die Lizenz aktiviert (*enabled*). Wird beim nächsten Hochfahren des Rechners die Funktion wieder aufgerufen, öffnet sich der Dialog erneut. Dieses Beispiel ist an die Second Sample-Hilfdatei angelehnt, die Sie nach Installation des CodeMeter Development Kit im Benutzer-Verzeichnis "%\Dokumente\Öffentliche Dokumente\WIBU-SYSTEMS" finden.

Lösung

Die Lösung besteht im Temporary Enabling der neuen Funktion. Dazu werden zunächst zwei separate Lizenzeneinträge angelegt: die Editor-Basisanwendung besitzt den Firm Code 10, Product Code 201000 und einen Feature Code 1; die neue Funktion besitzt den Firm Code 10, Product Code 201001 und ebenfalls einen Feature Code 1.

Für das Temporary Enabling legen Sie zuerst einen Enabling Block im Lizenzcontainer mit dem Firm Code 10 an. Danach binden Sie diesen Enabling Block an die Lookup Table des Firm Items 10. Nutzen Sie dazu das Kommandozeilen-Tool *CmBoxPgm*.

Sie finden *CmBoxPgm* in Form der ausführbaren Datei *cmboxpgm.exe* standardmäßig im Verzeichnis "%\Program Files%\CodeMeter\DevKit\bin". Für [andere Betriebssysteme](#)¹⁷ finden Sie *CmBoxPgm* an den gewohnten Stellen. Sie auch die [Enabling-Optionen für CmBoxPgm](#)³⁷⁹.

- Zum Anlegen des Enabling Blocks geben Sie die folgende Kommandozeile ein und adressieren Sie den *CmDongle* und das Firm Item 10.

```
cmboxpgm.exe -qs<serial number> -f10 -e:tp -eac:"MyAccess" -et:"FontApp" -edta:none -em:d,t+ -ca
```

Die folgende Tabelle listet die verwendeten Optionen und deren Bedeutung auf.

Option	Beschreibung
-e:tp	Anlegen einer Zugriffsart vom Typ Time PIN (tp). Wenn Sie hier eine Zugriffsart von Typ Simple PIN eingeben, so können Sie im weiteren keine Beschreibung und keine Disable Time als Ablaufdatum des Enablings angeben.
-eac:	Eingabe des Enabling Access Codes. Hier "MyAccess".
-et:	Eingabe des Enabling Texts zur Beschreibung des Enabling Blocks. Hier "FontApp".

Option	Beschreibung
-edta:	Eingabe der Disable Time. Hier none für keine Zeitbeschränkung des Enablings. Sie könnten hier abweichend vom Beispiel auch ein Ablaufdatum eintragen.
-em:	Eingabe des Enabling Mode. Hier d,t+ für das temporary Enabling.
-ca	Legt den Enabling Block an.

2. Zum Binden des Enabling Blocks an die Lookup Table des Firm Items 10 geben Sie die folgende Kommandozeile ein und adressieren Sie den *CmDongle* und das Firm Item 10:

```
cmboxpgm.exe -qs<Seriennummer> -f10 -e0:tp -eac:"MyAccess" -eat-
t10,201001,1:mod,loc:req+ -ca
```

Die folgende Tabelle listet die verwendeten Optionen und deren Bedeutung auf.

Option	Beschreibung
-e0:tp	Addressieren des Enabling Blocks mit dem Index 0 der Zugriffsart vom Typ Time PIN (tp). Welchen Index Sie hier verwenden hängt von der Anzahl und Reihenfolge möglicher anderer Enabling Blocks ab, die hier angelegt sind.
-eac:	Eingabe des Enabling Access Codes. Hier "MyAccess", den Zugriffscode, den Sie beim Anlegen des Enabling Blocks verwendet haben.
-eatt	10,201001,1 spezifiziert den Lizenzeintrag der neuen Funktion 'Font'. Im Falle des IFI als Adressierung würde hier 0 stehen, oder der Firm Code, wenn Sie an einen Firm Item binden möchten. :mod,loc Im aktivierte Zustand umfasst das Enabling die höchste Berechtigungsstufe modify (mod). Im deaktivierten Zustand ist lediglich das Lesen des Firm und Product Codes zulässig (loc). :req+ Sollte dieser Enabling Block an weitere Lookup Tables gebunden sein, stellt eine UND-Verknüpfung sicher, dass alle geforderten Faktoren, z.B. Aktivierungsmodi oder Berechtigungsebenen, zwingend zutreffend sind bevor eine Operation für einen gesamten <i>CmContainer</i> , ein Firm Item oder einen Lizenzintrag vorgenommen werden kann.
-ca	Nimmt das Lookup vor.

3. Zum Anzeigen des *CmContainer*-Inhaltes über *CmBoxPgm* nach beiden Vorgängen geben Sie die folgende Kommandozeile ein:

```
cmboxpgm.exe -qs<Seriennummer> -f10 -e0:tp -eac:"MyAccess" -eat-
t10,201001,1:mod,loc:req+ -ca
```

Sie erhalten den folgenden Zeilenauszug:

```
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: 2011-02-11 14:49:36 (UTC)
* Enable Block Table
  0|-]: TimePin, 16 bytes Access Code
    Disable Time = (never)
    Text (13 character(s)): "FontApp"
- No Enable Lookup Table exists
** Product Code 201000 at [17], dependencies = dsu
- No Enable Lookup Table exists
  - Feature Map, dependencies = dsu
  0x00000001
** Product Code 201001 at [16], dependencies = dsu
* Enable Lookup Table
  0:Loc[0] - Required, valid - E:Modify (7), D:Locate (0)
```

Information über das erfolgreiche Anlegen des Enabling Blocks und der Bindung an die Lookup Table erhalten Sie ebenfalls über das Kommandozeilen-Tool *cmu*.

Sie rufen *cmu* im Verzeichnis %\Program Files%\CodeMeter\Runtime\bin über den Befehl **cmu[32].exe** auf. Alternativ rufen Sie *cmu* über den Systemmenü-Eintrag "**Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt**" auf. Für die Betriebssysteme Mac OS, Linux und Sun ist dieser Befehl im Suchpfad hinterlegt.

Geben Sie die folgende Kommandozeile ein:

Cmu32.exe --enabling

Sie erhalten den folgenden Zeilenauszug:

```
- 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

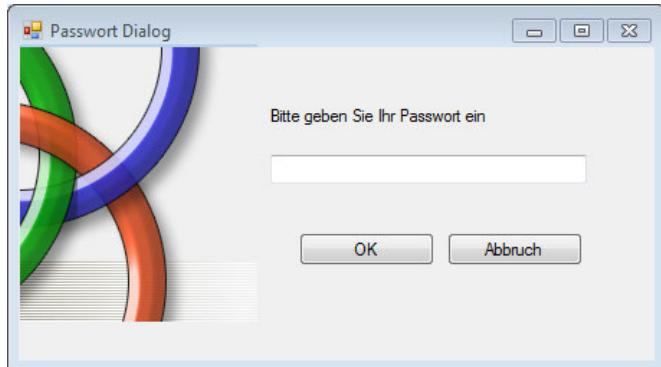
+-* PC=0 (Ref=16)
+-* PC=1000 (Ref=17)

* FC=100003
+-* PC=1 (Ref=16)

* FC=10
| 1 EnableBlock: +-----+
|   |Index 00: "FontAnwendung" (TimePin)   temp enabled
|   +-----+                               No DisableTime+
|   |
|   +-* PC=201000 (Ref=17)
|   +-* PC=201001 (Ref=16)
|   1 Item attached: +-----+
|   |   | Status | Level | Index | required? | enable level | disable level |
|   |   +       +       +       +       +       +       +       +
|   |   | -     | 0     | 00     | yes     | modify     (7) | locate     (0) |
|   |   +-----+
|   PI-Level = locate
```

Den Dialog zur Passwort-Abfrage (Enabling Access Code), der beim Verwenden der neuen Funktion erscheint, können Sie z.B. in *AxProtector* über die Anpassung der UserMessage programmtechnisch imple-

mentieren. Dann erscheint beispielsweise der folgende Passwort-Abfrage-Dialog.



11.3 Verwendung eigener Schlüssel

Zusammen mit der Firm Security Box haben Sie von Wibu-Systems initial einen Firm Key erhalten. Wenn Sie aus einem höheren Sicherheitsbedürfnis heraus, den Firm Key selber festlegen möchten, so können Sie das tun.

Bei einer Änderung des initialen Wertes des Firm Keys müssen Sie aber unbedingt sicherstellen, dass Sie den Firm Key äußerst sicher verwahren.

 Bei Verlust des Schlüssels oder einem Defekt der Firm Security Box kann auch Wibu-Systems den Firm Key nicht wiederherstellen.

Hidden und Secret Data

Auf der Product Item Ebene besitzen Sie außerdem auch die Möglichkeit, für einzelne Lizenzentriegen den Firm Key durch eigene Schlüssel zu ersetzen. Diese legen Sie entweder in einem Secret Data oder Hidden Data Feld ab.

 Aus Sicht der CodeMeter® Sicherheitsarchitektur bietet diese Alternative allerdings keine zusätzliche Sicherheit.

 Beispielsweise hat das Secret Data Feld den gleichen Sicherheitsstatus wie der Firm Key, d.h. es kann nur verwendet, aber nicht ausgelesen werden. Arbeiten Sie bereits schon mit einem eigenen Firm Key, dann haben Sie durch diese Alternative auch keinen zusätzlichen Sicherheitsnutzen.

Wie die untenstehende Abbildung aus dem *CodeMeter API Guide* zeigt, haben Sie dann für Ver- und Entschlüsselungsvorgänge die Auswahl zwischen der Verwendung des Firm Key, eines Secret Data oder Hidden Data Feldes.

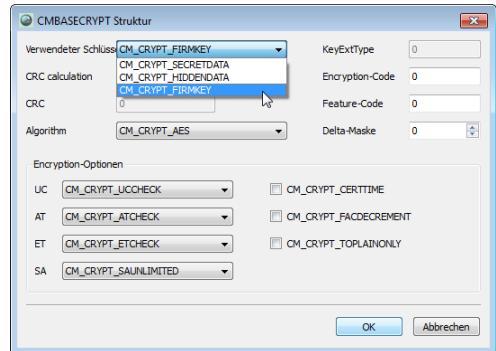


Abbildung 194: Verschlüsselungs-Alternativen

Anwenderszenarien umfassen hier z.B. die sicherheitstechnische Trennung unterschiedlicher Benutzergruppen einer Anwendung über Ver- und Entschlüsselungen mit verschiedenen Secret- oder Hidden Data Feldern. Ein Auftraggeber kann dadurch über eine Anwendung Aufträge getrennt an verschiedene Auftragnehmer vergeben, die wiederum untereinander nicht auf Auftragsdaten der anderen Auftragnehmer zugreifen können. Das bietet zusätzlichen Datenschutz. Oder Sie möchten gewährleisten, dass die Kommunikation zwischen verschiedenen technischen Geräten (Telefone, Brandschutzzentralen, etc.), an denen ein CmDongle angeschlossen ist, nur mit bestimmten Geräten mit identischen Schlüsseln möglich ist. Auch dann macht der Einsatz von Secret oder Hidden Data Feldern durchaus Sinn.

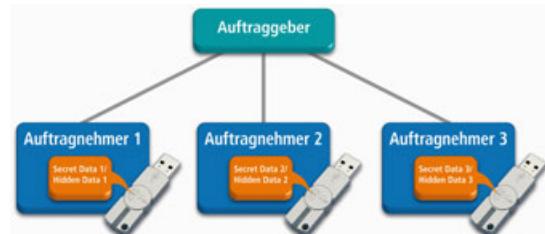


Abbildung 195: Anwendungsbeispiel: Secret Data, Hidden Data

Außerdem haben Sie noch die Möglichkeit mit der Option "AES direkt" Secret oder Hidden Data direkt mit dem Algorithmus zu ver- und entschlüsseln (siehe Abbildung unten). Diese Option bietet sich an, wenn Sie beispielsweise Rechenoperationen innerhalb einer geschützten Software, aber außerhalb des CmContainers durchführen möchten. Dies findet dabei ohne die komplette CodeMeter® Schlüsselableitung statt. Lediglich der Block mit den Parametern Firm Key und Black Key wird ent- oder verschlüsselt, d.h. ohne die sichtbaren Bestandteile Firm Code, Product Code, Feature Code und ohne den Encryption Code.

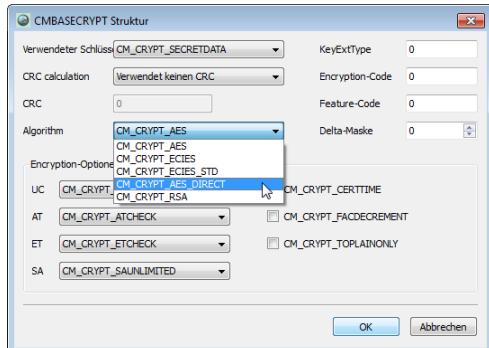


Abbildung 196: AES direkt für Secret Data, Hidden Data direkt verschlüsselt

Sollten Sie Fragen zu diesem Themenbereich haben, kontaktieren Sie bitte den WIBU Support.

Asymmetrische Verschlüsselung

Neben der symmetrischen Verschlüsselung bietet CodeMeter® auch die Möglichkeit, Daten über private und öffentliche Schlüssel asymmetrisch zu ver- und entschlüsseln, Signaturen für die Authentizitätsprüfung zu erzeugen und zu verifizieren.

Auch hier kann mit eigenen Schlüsseln in den Datenfeldern Secret und Hidden Data Feldern gearbeitet werden. Genau wie für den Firm Key gilt hier, dass bei der Verschlüsselung mit dem Elliptic Curves Cryptography (ECC) Algorithmus die gesamten 32 Byte als privater Schlüssel verwendet werden, um eine ECDSA Signatur zu berechnen. Der zu diesem privaten Schlüssel passende öffentliche Schlüssel kann dann über den *CmContainer* berechnet und anschließend verifiziert werden.

Im [CodeMeter API Guide](#)³³⁷ finden Sie dazu die notwendigen API Befehle und Funktionsblöcke: Authentifizierungs-API, Verschlüsselungs-API, Blöcke für die Durchführung verschiedener Ver- und Entschlüsselungsoperationen.

Sollten Sie Fragen zu diesem Themenbereich haben, kontaktieren Sie bitte den WIBU Support.

11.4 Der Zeitserver: System-Zeiten und die Zertifizierte Uhrzeit

Über die Product Item Optionen Activation und Expiration Time sowie Usage Period, aber auch in anderen Zusammenhängen spielen Zeitreferenzen in verschiedenen Lizenzierungsmodellen eine wichtige Rolle.

In jedem *CmContainer* sind verschiedene Zeiten hinterlegt, die dazu dienen einen regulären und sicheren Betrieb von zeitlich begrenzten Softwarelizenzen sicherzustellen. Was diese Zeiten bedeuten und wie und wann diese sich ändern, ist nachfolgend beschrieben. Sie finden die aktuellen Zeiten für jeden *CmContainer* in *CodeMeter WebAdmin* auf der Seite "[Inhalt | CmContainer](#)⁴⁶⁹".

Da der *CmContainer* keine Echtzeituhr (real time clock) im herkömmlichen Sinne enthält, tritt an ihre Stelle ein wesentlich ausfall- und manipulationssicherer Prüfungsmechanismus.

Jeder PC besitzt eine interne Uhr, ganz gleich ob unter Windows, Mac OS oder Linux. Jedoch ist es ein Leichtes, die Systemzeit des Computers entweder vor- oder zurückzustellen. Software mit einer zeitbasierten Lizenzierung, die sich allein auf die Zeit des Betriebssystems verlässt, kann leicht getäuscht werden. Wenn z.B. ein Abonnement eines Anwenders am 31. Dezember endet, kann er die Systemuhr auf November oder Oktober zurückstellen, und die Software unter Verletzung der Lizenzbedingungen länger

auch weiterhin nutzen. Offensichtlich ist es also notwendig, solche Umgehungsversuche zu verhindern. Eine Möglichkeit ist die Verwendung einer eigenen Uhr mit einer Batterie im Dongle. Doch was passiert, wenn die Batterie leer ist? Wie sicher ist eine Uhr mit Batterie? Eine andere Möglichkeit ist die Verwendung eines NTP Servers (Network Time Protocol) über das Internet. Hierbei stellt sich die Frage, wie man die Verwendung eines manipulierten NTP Servers erkennt und unterbindet sowie was passiert, wenn der Kunde gerade online ist.

CodeMeter® unterscheidet sich hier komplett von den anderen Dongles und bietet eine einzigartige Lösung des oben beschriebenen Dilemmas.

Die Uhr im CmDongle SmartCard Chip

Jeder CmDongle hat eine eigene laufende Uhr, die sich im internen SmartCard Chip befindet. Diese wird als **CodeMeter Systemzeit** bezeichnet. Nicht zu verwechseln mit der **Systemzeit des Computers**. Für CodeMeter® ist dies die einzige gültige Zeit. Eine Ver- oder Entschlüsselung kann nur erfolgen, wenn das Ablaufdatum der entsprechenden Lizenz noch nicht durch die interne Uhr erreicht oder überschritten wurde.

Die Uhr in den SmartCard Chip zu legen hat einen unschlagbaren Vorteil: Die Uhr ist dort sicher gegen Manipulation geschützt. Eine Uhr in einem extra Flash-Speicher, wie es bei einigen anderen Dongles erfolgt, kann durch einen Hobbybastler mit wenig Aufwand manipuliert werden. Leider hat die Uhr im SmartCard Chip aber auch einen Nachteil: Sie läuft nur wenn der *CmDongle* angeschlossen ist und mit Strom versorgt wird.

Daher bleibt die CodeMeter®-Uhr stehen, sobald der *CmDongle* abgezogen bzw. der Rechner ausgeschaltet wird. Beim nächsten Power-On, also beim Einsticken oder Einschalten, wird die CodeMeter®-interne Uhr (**Systemzeit CodeMeter**) mit der Zeit des Computers (**Systemzeit PC**) synchronisiert.

Aber nur vorne, also in Richtung Zukunft. Ist dies nicht möglich, wird ab der letzten gespeicherten Zeit gestartet. Die CodeMeter®-interne Uhr läuft also monoton in Richtung Zukunft und kann durch den Endanwender nicht zurückgestellt werden. Damit läuft die CodeMeter®-Uhr aber auch noch nach vielen Jahren, während herkömmliche batteriebetriebene Uhren längst den Dienst eingestellt haben.

Die zertifizierte Zeit

In vielen Fällen reicht die Genauigkeit und Sicherheit der internen Uhr aus. Für alle anderen Fälle bietet Wibu-Systems die Möglichkeit die interne Uhr gegen einer der Wibu-Systems-Zeitserver abzugleichen. Die Wibu-Systems-Zeitserver bekommen ihre Uhrzeit ähnlich wie ein NTP Server aus mehreren vertrauenswürdigen Quellen (Atomuhr, Funkuhr, ..), stellen aber zusätzlich einen geschützten Kanal für die Übertragung der Zeit in den *CmContainer* zur Verfügung. Damit ist eine Manipulation der Übertragung und ein Vortäuschen eines falschen Zeitservers ausgeschlossen.

Beim Abgleich der Systemzeit im *CmContainer* gegen einen Wibu-Systems Zeitserver, wird die interne Uhr auf das aktuelle Datum gestellt und zusätzlich wird dieser Zeitpunkt als Zeitstempel im *CmDongle* abgespeichert. Dieser Zeitstempel wird als zertifizierte Zeit bezeichnet. Der Zeitstempel wurde beim Verschicken auf dem Zeitserver signiert und kann somit nicht manipuliert werden.

Eine eventuell (zu Testzwecken oder durch Versehen) in die Zukunft verstellte Systemzeit im *CmDongle* kann somit auch durch das Abholen einer neuen zertifizierten Zeit über den Zeitserver, ohne Eingriff des Softwareherstellers, wieder korrigiert werden.

Abfrage in der Software

Für die Benutzung der Systemzeit des *CmDongles* müssen Sie bei der Implementierung nichts beachten. Diese wird durch *CodeMeter®* automatisch ausgewertet. Ist die entsprechende Lizenz abgelaufen, egal ob durch ein Ablaufdatum oder durch einen Nutzungszeitraum, kann die Software nicht entschlüsselt und damit nicht genutzt werden. Ist die Lizenz noch gültig, oder besitzt sie gar kein Ablaufdatum, dann startet die Software.

Die Verwendung des Zeitservers ist eine Option, die Sie zusätzlich nutzen können, nicht müssen. Ohne diese Option läuft Ihre Software komplett ohne Online-Zugriff. Bei der Verwendung der Zeitserver-Option haben Sie die folgenden Möglichkeiten:

- Software startet nur, wenn die Zeit seit dem letzten Einschalten der Uhr über den Zeitserver abgeglichen wurde.
Ein Nachteil dieser Methode ist, dass der Computer mit dem verbundenen *CmContainer* hierzu ständig mit dem Internet verbunden sein muss - sonst ist keine Zertifizierte Zeit verfügbar.
- Software versucht die Zeit abzugleichen und startet nur, wenn der letzte erfolgreiche Abgleich nicht älter als xy Stunden ist.
- Software versucht die Zeit abzugleichen, startet aber immer.
- Software startet ohne die Zeit abzugleichen

Siehe z.B. Verschlüsselung mit *AxProtector* encryption: Laufzeiteinstellungen | Erweiterte Laufzeiteinstellungen:

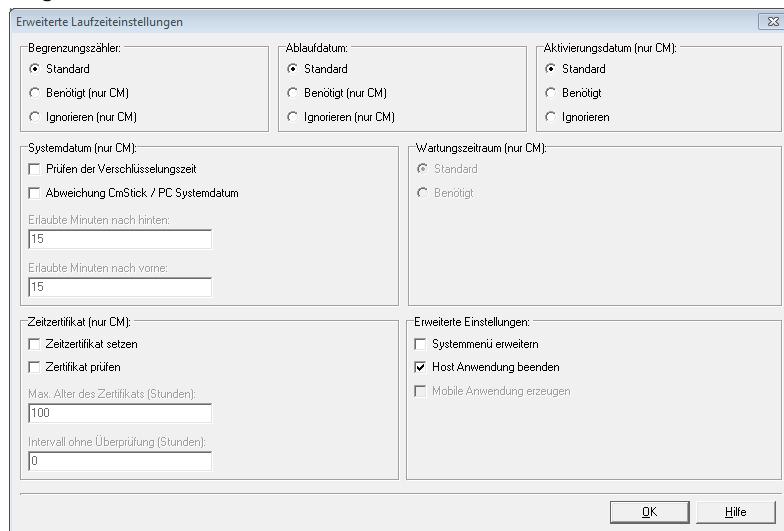


Abbildung 197: *AxProtector* - "Erweiterte Laufzeiteinstellungen"

Zeiten im CodeMeter WebAdmin

Im *CodeMeter WebAdmin* sehen Sie die Systemzeit des *CmContainers*, des PCs und die zertifizierte Zeit.

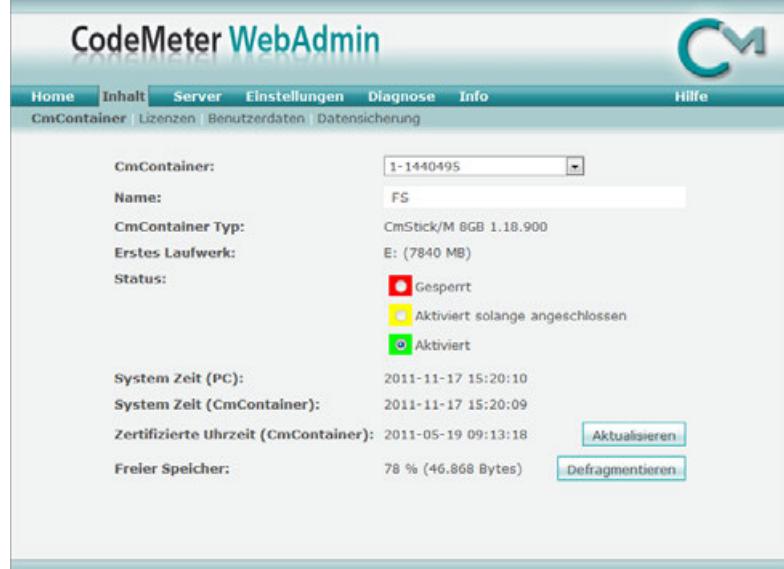


Abbildung 198: CodeMeter WebAdmin – "Inhalt | CmContainer"

Zeiten bei CmActLicense

Das gleiche Prinzip kommt auch bei CmActLicense zum Einsatz. D.h. jede CmActLicense-Lizenz hat Ihre eigene monoton in Richtung Zukunft laufende Uhr. Im Gegensatz zum CmDongle wird die letzte Uhrzeit allerdings nicht in einer sicherer Hardware gespeichert, sondern verschlüsselt auf dem Computer versteckt.

Im Falle eines Zurückkopierens einer älteren Lizenzdatei (mit noch nicht abgelaufener Lizenz), passen die versteckten Zeitinformationen nicht mehr und die komplette Lizenz wird von CodeMeter® als ungültig markiert und kann nicht mehr verwendet werden. Auch durch das Zurückstellen der Systemzeit des Computers, kann eine einmal abgelaufene CmActLicense-Lizenz nicht wieder verwendet werden.

CmStick/T mit Batterie

In wenigen Fällen, zum Beispiel wenn die Lizenzen nur selten und für kurze Zeit verwendet werden, ist eine permanent laufende Uhr wünschenswert, auch während der CmDongle nicht am Computer angeschlossen ist. Für diese Fälle bietet Wibu-Systems den CmStick/T an.

Der CmStick/T enthält eine Batterie. Über diese Batterie wird die Power-Off Zeit überbrückt. Beim nächsten Einschalten des CmDongle wird diese zusätzlich als eine weitere Quelle neben der Uhrzeit des Computers verwendet, um die Systemzeit im CmDongle zu stellen. Das Konzept der sicheren Uhr im Smartcard Chip wird also beibehalten. Und nach Ablauf der Batterie funktionieren alle Zeiten im CmDongle so wie heute ohne Batterie.

11.5 Sperren des CmContainers

Es gibt verschiedene Szenarien, in denen ein Lizenzgeber daran interessiert sein kann, dass die Nutzung eines *CmContainers* gesperrt wird. Das Sperren kann sich auf einzelne Firm Item Ebenen, aber auch auf ganze *CmContainer* beziehen.

Sperren eines Firm Items

Das Sperren einer Firm Item Ebene, d.h. eines Lizenzcontainers ist sinnvoll:

- aus der Software heraus, wenn ein Manipulationsversuch an der geschützten Software festgestellt wird,
- bei einem gemeldeten Diebstahl oder Verlust eines *CmDongles*,
- wenn einem bestimmten Lizenznehmer die Nutzung untersagt werden soll, z.B. wegen Zahlungsrückständen bei pay-per-use Lizenz.

Sperren aus der Software heraus

Beim Sperren einer Firm Item Ebene aus der Software heraus übernehmen dies Anti-Debugger-Mechanismen im Zusammenspiel mit dem Firm Access Counter (FAC).

Firm Access Counter (FAC)

Der Firm Access Counter (FAC) liegt auf der Firm Item Ebene eines *CmDongles*. Über diesen Zähler ist es möglich zu kontrollieren, ob eine Firm Item Ebene für Ver- und Entschlüsselungsvorgänge benutzt werden kann oder nicht. Standardmäßig ist der FAC deaktiviert und besitzt den Wert 65535. Er kann auf jeden anderen beliebigen Wert programmiert werden.



Hat der FAC einen Wert von 0 ist das Firm Item gesperrt.

In *AxProtector* ist dieser Mechanismus auf der Seite "Sicherheitsoptionen" über die Aktivierung von "Sperren der Hardware" realisiert. Mit dem Softwareschutz-API WUPI implementieren Sie dies über den **WapiCheckDebugger** Befehl, im *Kern-API* besitzt die **CmCrypt** Funktion die **Fac_Decrement** Option, die den FAC um 1 herunterzählt.

Erkennt die Software den Versuch einer Manipulation wird eine Sperrsequenz geschickt, die den FAC um die definierte Zahl herunterzählt. Wird die Zahl 0 erreicht, wird die Firm Item Ebene für die weitere Nutzung gesperrt. Dadurch ist nicht der ganze *CmDongle* gesperrt, sondern nur die Lizenz, die sich im Firm Item-Lizenzcontainer des betreffenden Lizenzgebers befinden! Der Anwender kann Software-Produkte anderer Lizenzgeber über die Lizenz im *CmDongle* weiterhin benutzen.

Der Lizenzgeber kann über die Fernprogrammierung den FAC jederzeit hochsetzen und die gesperrte Firm Item Ebene wieder entsperren.

Diebstahl oder Verlust – eigene Blacklist

Wird ein *CmDongle* als gestohlen oder verloren gemeldet, hat der Lizenzgeber die Möglichkeit, eine eigene Liste anzulegen, die diese *CmDongles* enthält.

Bei der nächsten Aktualisierung der lizenzierten Software wird bei diesen *CmDongles* der Firm Access Counter auf 0 gesetzt. Sollte der *CmDongle* wieder auftauchen, oder auch eventuell ausstehende Rechnung beglichen sein, kann auch hier über die Fernprogrammierung der FAC hochgesetzt und die Sperrung rückgängig gemacht werden.



Wibu-Systems empfiehlt das Anlegen einer solchen Liste.

Sperren des gesamten *CmDongles*

Das Sperren eines ganzen *CmDongles* ist möglich:

- bei einem gemeldeten Diebstahl oder Verlust.

Dann hat der Lizenzgeber die Option, den ganzen *CmDongle* global über Wibu-Systems sperren zu lassen.



Dieser Prozess kann nur ausschließlich online erfolgen.

Das Sperren erfolgt über die Nutzung des Wibu-Systems Zeitservers und die Aktualisierung des *CmDongles* über die zertifizierte Zeit (Certified Time). Dabei kommt eine globale Wibu-Systems Sperrliste zum Tragen, die die gemeldeten *CmDongles* enthält, die bei der nächsten Aktualisierung der Zeit gesperrt werden sollen. Die Aktualisierung kann auch in die lizenzierte Software selbst eingebaut werden, so dass ein Lizenznehmer gefordert ist, in regelmäßigen Abständen auf dem Wibu-Systems Zeitserver die zertifizierte Zeit zu aktualisieren.

Wibu-Systems sperrt dann den betreffenden *CmDongle*, wenn zur Aktualisierung auf den Zeitserver zugriffen wird. Natürlich realisiert Wibu-Systems dies nur für *CmDongles*, wenn die Identität des Lizenzgebers eindeutig gewährleistet ist.



Das Sperren kann hier nicht rückgängig gemacht werden.

11.6 Sicherung des *CmDongle*-Inhaltes

Backup Mechanismus

CodeMeter® speichert alle Lizenzen im *CmDongle*. Die Hardware hat dabei einen besonderen Wert, der durch die Summe des Kaufpreises all der Lizenzen definiert wird, die sich im *CmDongle* befinden. Wenn der *CmDongle* beschädigt oder gestohlen wird oder verloren geht, ist auch dieser Wert verloren. Das kann ein großer Verlust sowohl für den Besitzer des *CmDongle*, als auch im besonderen für den Eigentümer dieser Lizenzen darstellen. Deswegen besitzt CodeMeter® einen Backup-Mechanismus, der den Inhalt eines *CmDongles* in eine separate binäre *.wbb Datei auf den Rechner schreibt und speichert.

Sicherung durchführen

Zur Sicherung kann in *CodeMeter WebAdmin* angegeben werden, wohin diese Datei gespeichert werden soll und ein Sicherungs-Zeitintervall angegeben werden. Standardmäßig wird eine Sicherung alle 24 Stunden durchgeführt.



Diese Datei ist verschlüsselt abgespeichert und daher sicher vor Angriffen und Manipulation.

Diese Backup-Datei enthält alle Lizenzinformationen aus dem *CodeMeter® SmartCard Speicher* – mit Ausnahme des Secret Data Feldes – d.h.

- die gesamte *CmDongle* Informationsstruktur (Seriennummer, Serienschlüssel, *CmDongle* Version etc.),
- die Implicit Firm Item Ebene und
- die Inhalte aller Firm Item Lizenzcontainer.

Sicherung einspielen

Allerdings umfasst das Zurückspielen dieser Daten durch *CodeMeter WebAdmin* derzeit nur die Daten für die Firm Item Ebene mit dem Firm Code 0, d.h. das Implicit Firm Item. Damit können diese gesicherten Daten auf einen anderen *CmDongle* überspielt werden, solange für den zweiten *CmDongle* dasselbe Kennwort (User Individual Key) verwendet wird. Für das Einspielen der anderen Daten auf der Firm Item sowie Product Item Ebene gibt es derzeit kein separates *CodeMeter®*-Werkzeug.

In den meisten Fällen protokollieren Software-Hersteller jedoch eigene Historien von *CmDongle*-Programmierungen, oder benutzen *CodeMeter®*-Werkzeuge dazu, so dass ein Nachvollzug möglich ist

Einschicken an Wibu-Systems

Geht ein *CmDongle* verloren und hat der Software-Hersteller ein Backup angelegt und möchte wichtige Information auslesen, z.B. Nachweis von bestimmten Aktionen über den Stand eines Unit Counter, etc., muss die Backup-Datei an Wibu-Systems geschickt werden. Diese Datei kann dann mit der passenden Firm Security Box manuell bearbeitet werden. Natürlich kann auch hier das Secret Data Feld nicht ausgelesen werden. Wurden dabei beim Kunden veränderliche Daten wie Usage Period oder Unit Counter lokal umprogrammiert, kann ein Nachweis des letzten aktuellen Standes, z.B. Tage oder Stände in Verbindung mit dem *CmDongle*-internen Zeitstempel ausgelesen werden.

11.7 CodeMeter im Wide Area Network (WAN)

CodeMeter® unterstützt standardmäßig den Zugriff auf Lizenzen, die sich auf einem Server in Netzwerken befinden basierend auf der Kommunikation zwischen zwei Instanzen der *CodeMeter®*-Laufzeitumgebung (*CodeMeter Lizenzserver*).

Im Fall eines lokalen Netzwerkes (Local Area Network, LAN) findet die Kommunikation zwischen einem lokalen *CodeMeter Lizenzserver* und einem Netzwerk-*CodeMeter Lizenzserver* über das TCP/IP-Protokoll und die Kommunikationsart *CmLAN* statt.

Seit der *CodeMeter®*-Version 5.0 ist die Kommunikationsart *CmWAN* für Weitverkehrsnetze (Wide Area Network, WAN) verfügbar. Ein WAN ist ein Rechnernetz, das sich im Unterschied zu LAN über ein sehr großen geografischen Bereich erstrecken kann und die Anzahl der angeschlossenen Rechner auf keine bestimmte Zahl begrenzt ist.

Im Fall eines Wide Area Network (WAN) findet die Kommunikation zwischen *CodeMeter Lizenzserver* auf Clients und einem Netzwerk-*CodeMeter Lizenzserver* über das HTTPS-Protokoll und die Kommunikationsart *CmWAN* statt.

Im folgenden wird ein Überblick über die [WAN-Infrastruktur](#)⁴²⁴ gegeben, die für die Verwendung von *CmWAN* notwendig ist. Danach folgt eine Beschreibung der erforderlichen Schritte, die für die *CodeMeter®*-seitige [Implementierung](#)⁴²⁵ benötigt werden.

11.7.1 WAN-Infrastruktur

Die Verwendung der Kommunikationsart *CmWAN* in einem WAN benötigt eine spezielle Infrastruktur. Wesentlicher Bestandteil ist dabei ein Reverse Proxy, der sich hinter einer Firewall in einer Demilitarisier-ten Zone (DMZ) befindet.

Der Reverse Proxy dient als Kommunikationsdrehscheibe für *CodeMeter®-Clients*, die auf Lizenzen zu-greifen, die sich auf einem internen Server befinden, auf dem ebenfalls ein *CodeMeter Lizenzserver* läuft. Dabei kommuniziert der *CodeMeter®-Client* mit dem Reverse Proxy immer über eine TLS/SSL-gesicherte und verschlüsselte Verbindung (HTTPS). Über diese zentrale Anlaufstelle sind die *CodeMeter®-Clients* nicht direkt mit dem internen Server verbunden und die Identität des Servers bleibt verborgen.

Auf der Kommunikationsebene leitet der Reverse-Proxy die HTTPS-Anfrage der *CodeMeter®-Clients* als HTTP-Anfrage an den *CodeMeter Lizenzserver* auf dem Server weiter. Umgekehrt leitet der Reverse Proxy die HTTP-Antwort des *CodeMeter Lizenzservers* auf dem Server für den Rückweg an die *CodeMeter®-Cli-ents* gesichert über HTTPS weiter.

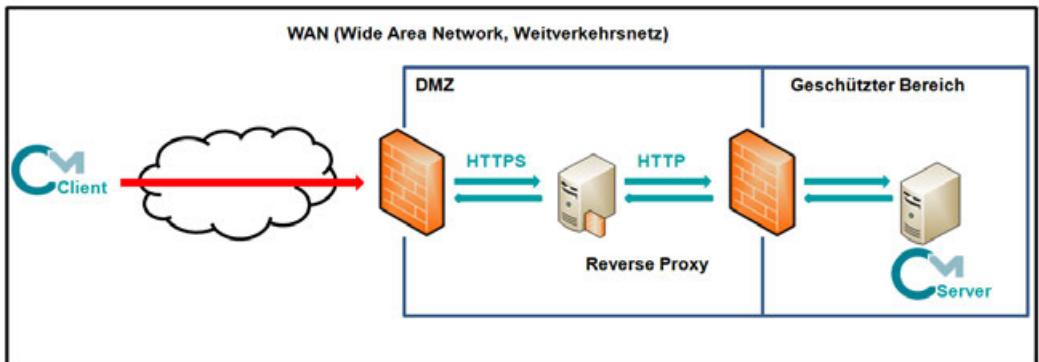
Zusätzlich kann der Reverse Proxy Authentifizierungsaufgaben übernehmen. Dann muss sich ein Client über einen Benutzernamen und ein Kennwort am Proxy Server authentifizieren (user, password) oder der Reverse Proxy liefert ein Client-Zertifikat aus, das durch den *CodeMeter®-Client* zur Authentifizierung be-nutzt wird.



Derzeit unterstützt *CodeMeter®* eine Digest Authentication auf der Client-Seite. Es ist geplant, dass zukünftige *CodeMeter®-Versionen* auch die Verwendung von Client-Zertifikaten unterstüt-zen.

Das folgende Schaubild schematisiert diese Infrastruktur.

Kommunikationsebene:



Authentifizierungsebene (HTTPS):

Client-Anwender:	Reverse Proxy
<ul style="list-style-type: none">überprüft ein Server-Zertifikatweist sich gegenüber dem Reverse Proxy aus mit "username" und "password"	<ul style="list-style-type: none">liefert ein Server-Zertifikat ausweist sich gegenüber einem Client-Anwender aus

Abbildung 199: CmWAN: Netzwerkkommunikation und Authentifizierung

Die Einrichtung und Konfiguration des WAN inklusive des Reverse Proxys wird nicht von Wibu-Systems übernommen, dies muss vom Kunden selbst vorgenommen werden. Benötigen Sie dabei Unterstützung, so helfen Ihnen hier gerne WIBU Professional Services weiter.

i Falls Sie für Testzwecke ein selbst ausgestelltes Test-Zertifikat am Reverser verwenden, beachten Sie bitte, dass Sie dieses als Root-Zertifikat auf dem Client einspielen. Das Root-Zertifikat, mit dem der Client das empfangene Server-Zertifikat validieren soll, muss im System-Zertifikatsspeicher liegen und damit für den gesamten Rechner Gültigkeit besitzen.

Voraussetzungen:	
Proxy Server	Server mit Lizenz
<ul style="list-style-type: none">Unterstützung TLS/SSL-gesicherte Verbindung (HTTPS)Umsetzung von HTTPS nach HTTP und umgekehrtUnterstützung Authentifizierung	<ul style="list-style-type: none">Installierter CodeMeter License Server mind. Version 5.0

11.7.2 CodeMeter-seitige Implementierung

Für die Verwendung von CmWAN müssen Sie CodeMeter® für die folgenden Bereiche mit folgenden Tools konfigurieren:

- [Lizenz-Programmierung](#)⁴²⁶ (*CmBoxPgm, CodeMeter API Guide*)
- [Anwendungsverschlüsselung](#)⁴²⁹ (*AxProtector*)
- [Konfigurieren der CmWAN-Netzwerkkommunikation](#)⁴²⁹ (*CodeMeter WebAdmin; Registry- bzw. Server.ini-Einträge*)

11.7.2.1 Programmierung der Lizenzen

Firm Security Box-Lizenzeintrag

Um Lizenzen für die CmWAN-Verwendung programmieren zu können, benötigen Sie zunächst einen separaten Lizenzeintrag [100021:10000:1] in Ihrer Firm Security Box (FSB).



Diesen separaten FSB-Lizenzeintrag erhalten Sie von Wibu-Systems.

Programmierung eines Lizenzeintrages über CmBoxPgm

Mit dem Tool CmBoxPgm können Sie über die License Quantity-[Option](#)³⁶⁴ w für jeden Lizenzeintrag festlegen, ob dieser über CmWAN verwendet wird.

Sie finden CmBoxPgm in Form der ausführbaren Datei `cmboxpgm.exe` standardmäßig im Verzeichnis `%Program Files%\CodeMeter\DevKit\bin`. Für andere Betriebssysteme finden Sie CmBoxPgm an den gewohnten Stellen.

Die Programmiersequenz folgt dem Muster:

```
cmboxpgm.exe / [CmContainer] /f [...] /p[...] /plq<Anzahl>:w
```

/ [CmContainer] adressiert den zu programmierenden CmContainer (siehe [hier](#)³⁵⁷)

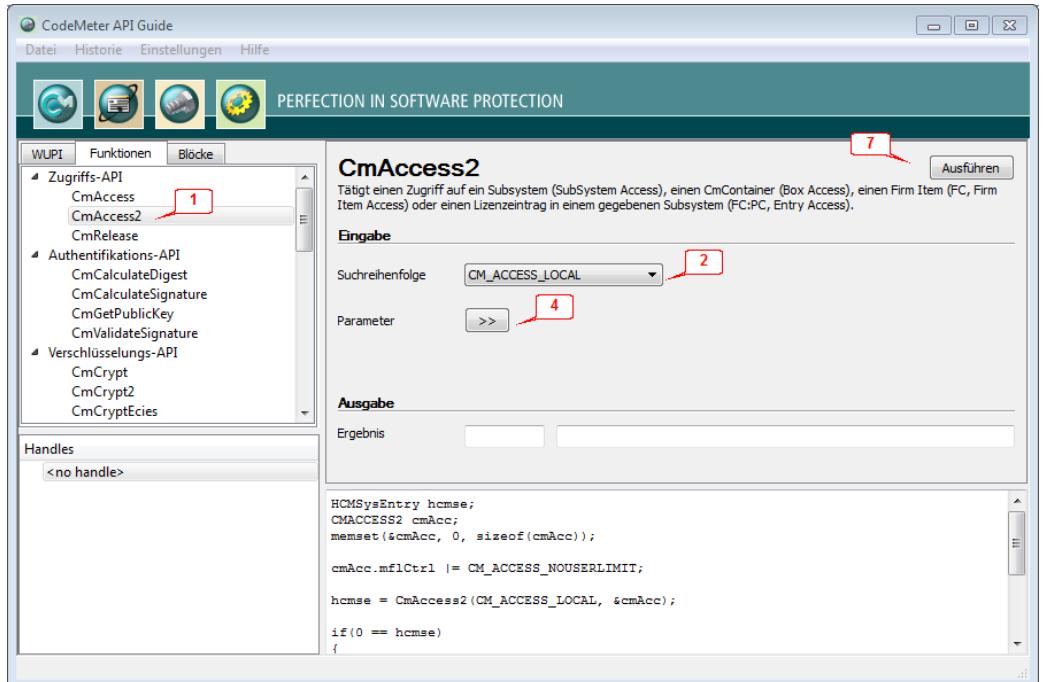
/f [...] / p [...] spezifiziert den Lizenzeintrag (Firm Code/Product Code)

Programmierung des Lizenzzugriffes über CodeMeter API Guide

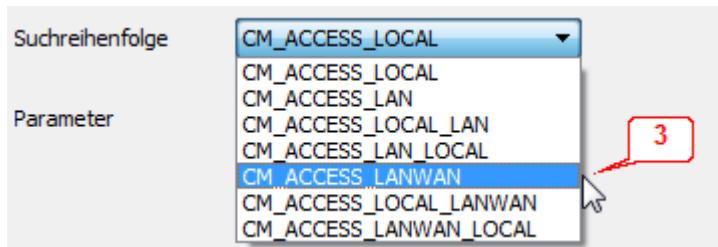
Um den erforderlichen Lizenzzugriff über die [CmAccess2](#)-Struktur mit Hilfe des Tools [CodeMeter API Guide](#)³³⁷ zu programmieren, öffnen Sie den *CodeMeter API Guide* über den Startmenü-Eintrag

"**CodeMeter | Tools | CodeMeter API Guide**" und gehen dann wie folgt vor:

1. Auswählen des **CmAccess2**-Eintrages über den Karteireiter "**Funktionen**" und **Zugriffs-API**.
Sie erhalten zusätzliche eine kontextsensitive Hilfe, wenn Sie die F1-Taste drücken.



2. Aufklappen der **Suchreihenfolge**-Liste.
3. Auswählen des WAN-Eintrages.



Generell wird *CmWAN* ähnlich wie der normale *CodeMeter®*-Netzwerzugriff behandelt. Die folgenden WAN-Flags existieren:

- CM_ACCESS_LANWAN:
Wenn in der Serversuchliste oder in der **CmAccess2**-Struktur in `mszServername` (siehe hier) eine *CmWAN*-Adresse eingetragen ist, wird wie beim Netzwerzugriff versucht, über diese Adresse einen Lizenzzugriff durchzuführen.
- CM_ACCESS_LOCAL_LANWAN:
Wie oben, mit der üblichen Suchreihenfolge "zuerst lokal, dann *CmLAN* und *CmWAN* laut Server"

suchliste".

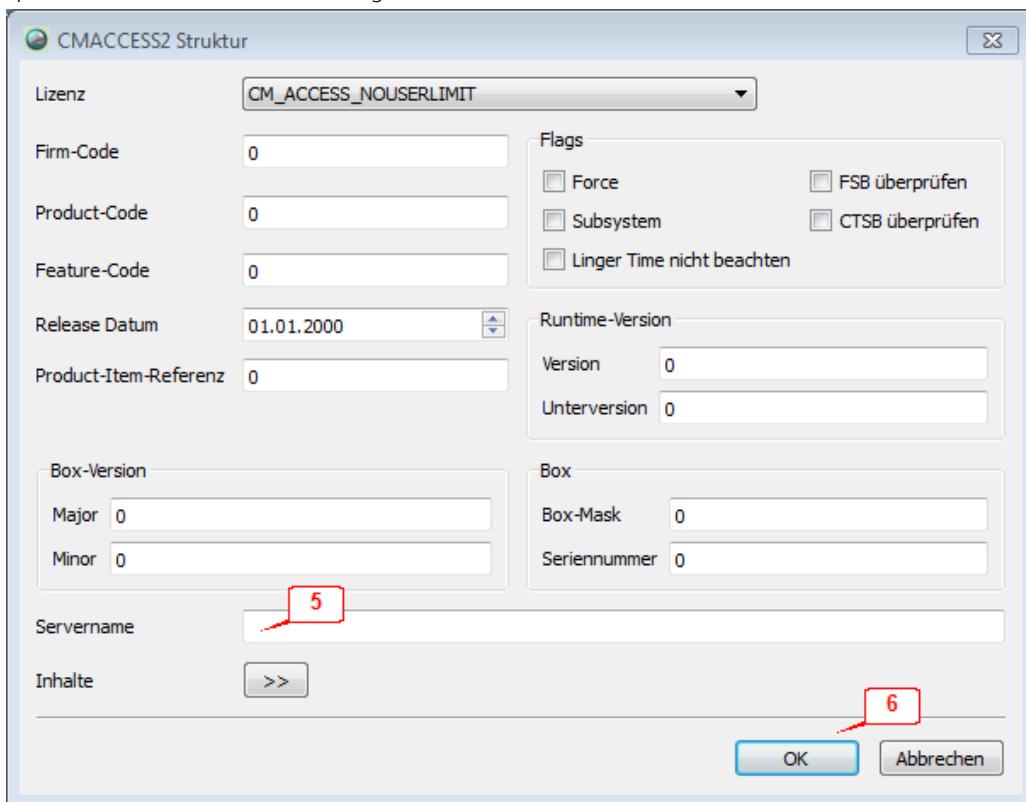
- CM_ACCESS_LANWAN_LOCAL:

Zuerst mit der üblichen Suchreihenfolge "CmLAN und CmWAN laut Serversuchliste, sonst lokal".

 Wenn Sie CmWAN über einen der Flags erlauben, wo wird automatisch auch der CmLAN-Zugriff aktiviert. Auf der anderen Seite erfolgt bei **CmAccess2** nie ein CmWAN-Zugriff, wenn nicht eines der drei Flags angegeben wird.

4. Klicken der Schaltfläche ">>" zum Öffnen der **CMACCESS2-Struktur**.

Spezifizieren oder aktivieren Sie die gewünschten Lizenzdetails oder Parameter.



5. Optionales Eingeben der Adresse (**Servername**), unter der die CodeMeter®-Laufzeitumgebung (CodeMeter License Server) auf dem Server erreichbar ist.

Die Adressierung erfolgt nach dem Muster:

`https://user1:password1@reverse proxy address/servername`

z.B. `https://user1:password1@cmwantest1.wibu.local/cmwan/test`



Beachten Sie bitte, dass Sie unbedingt den Präfix `https://` angeben.



Derzeit wird lediglich eine Standard-Digest Authentication unterstützt.

6. Klicken der Schaltfläche "**OK**", um die Parameter des CmWAN-Lizenzzugriffes abzuspeichern.
7. Klicken der Schaltfläche "**Ausführen**".
8. Kopieren des erzeugten Inhaltes des Ausgabefensters in den Quelltext der zu schützenden Anwendung.

11.7.2.2 Verschlüsselung der geschützten Anwendung

Die Verwendung von *CmWAN* muss beim Verschlüsseln der zu schützenden Anwendung explizit angegeben werden. Verwenden Sie dazu die Kommandozeilen-Variante des Tools *AxProtector* für den automatischen Softwareschutz. Gehen Sie dazu wie folgt vor:

1. Aufrufen der für den Projekttyp [passenden AxProtector-Version](#)²⁹².
Der Aufruf folgt dem Muster:
`AxProtector-Aufruf -<Optionen> <Pfad und Name der zu schützenden Anwendung>`
2. Setzen Sie im Bereich der Einstellungen für die Lizenzierungssysteme die Option `-s`²⁹⁴ nach den WAN-Erfordernissen. Die folgenden Parameter stehen zur Verfügung:

Parameter `-SW`

benutzt das Wide Area Network-Subsystem (WAN).

Parameter `-SLW`

benutzt zuerst das lokale Subsystem (Lokal), danach das Wide Area Network-Subsystem (WAN).

Parameter `-SWL`

benutzt zuerst das Wide Area Network-Subsystem (WAN), danach das lokale Subsystem (Lokal).



Bitte beachten Sie, dass bei der Verwendung von WAN automatisch auch LAN aktiviert ist, da WAN eine Erweiterung der LAN-Kommunikation darstellt.

11.7.2.3 Konfigurieren der CmWAN-Netzwerkkommunikation

Zur Konfiguration der *CmWAN*-Netzwerkkommunikation bestehen zwei alternative Vorgehensweisen: entweder über [CodeMeter WebAdmin](#)⁴³⁰ oder über das [Profiling](#)⁴³¹.



Bitte beachten Sie, dass Einstellungen im Profiling nur notwendig sind, wenn CodeMeter Lizenzserver nicht laufen sollte.

11.7.2.3.1 CodeMeter WebAdmin-Konfiguration

Zum Einrichten von CodeMeter® in einem WAN, gehen Sie wie folgt vor:

Server konfigurieren

1. Starten des CodeMeter WebAdmin (siehe [hier](#)⁴⁶⁷).
2. Navigieren zur Seite "[Einstellungen | Server](#)"⁴⁷³.
3. Aktivieren der **Starte CmWAN Server**-Option, um den Rechner in einem Wide Area Network (WAN) zu nutzen und Lizenzzugriffe zu ermöglichen.
4. Angeben eines **CmWAN Port** im gleichnamigen Feld.
Der Port 22351 ist die Standardeinstellung für die CodeMeter® Kommunikation über WAN. Sie können diesen Wert an Ihre Bedürfnisse anpassen. In diesem Fall sollten Sie allerdings sicherstellen, dass:
 - alle CodeMeter Lizenzserver diesen Port benutzen, wenn die CodeMeter®-geschützte Anwendung über das Wide Area Network (WAN) auf Lizenzern zugreifen.
 - der eingerichtete Reverse Proxy dieselbe Port-Einstellung besitzt.

4. Klicken der "**Übernehmen**" Schaltfläche speichert die Einstellungen ab.

Das Setzen der Server-Einstellungen macht für manche Änderungen einen Neustart des CodeMeter® Dienstes erforderlich. Dazu müssen Sie den *CmContainer* aber nicht auswerfen oder deaktivieren.

Nachdem Sie die Änderungen durchgeführt haben, können Sie in [CodeMeter Kontrollzentrum](#)⁴⁴⁸ den CodeMeter® Dienst stoppen und danach wieder starten. Für Nicht-Windows-Betriebssysteme siehe [hier](#)⁴⁴³.

Serversuchliste konfigurieren

5. Navigieren zur Seite "[Einstellungen | Netzwerk](#)"⁴⁷¹.
6. Verwenden einer **Serversuchliste**, die eingerichtete CodeMeter® Netzwerk- und WAN (Wide Area Network)-Server und deren Reihenfolge für die Beantwortung für Client-Anfragen definiert.
7. Eingeben der IP-Adresse(n) für Client-Anfragen an den definierten CodeMeter License Server im Wide Area Network.



Bitte beachten Sie, dass Sie bei der Angabe der IP-Adresse immer ein "https://\" voranstellen. Diese wird für die abgesicherte Kommunikation zu einem Reverse Proxy im WAN benötigt.

- Die Serversuchliste bearbeiten Sie, indem Sie über die entsprechenden Schaltflächen Server "**Hinzufügen**", "**Entfernen**", aber auch in der Reihenfolge ändern ("**Auf**" und "**Ab**").
8. Klicken der "**Übernehmen**" Schaltfläche speichert die Einstellungen ab.
Das Setzen der Server-Einstellungen macht für manche Änderungen einen Neustart des CodeMeter® Dienstes erforderlich. Dazu müssen Sie den *CmContainer* aber nicht auswerfen oder deaktivieren. Nachdem Sie die Änderungen durchgeführt haben, können Sie in [CodeMeter Kontrollzentrum](#)⁴⁴⁸ den CodeMeter® Dienst stoppen und danach wieder starten. Für Nicht-Windows-Betriebssysteme siehe [hier](#)⁴⁴³.

11.7.2.3.2 Profiling in der Registry oder in der server.ini-Datei

Über Registry-Einträge bzw. Einträge in die `server.ini`-Datei (Abschnitt [General]) sind Sie in der Lage, CmWAN-Netzwerkeinstellungen zu konfigurieren.

Die folgende Tabelle zeigt Ihnen wo sie, für welches Betriebssystem im Profiling die CmWAN-Netzwerk-kommunikationseinstellungen setzen können.

Betriebssystem	Eintrag
Windows	<code>HKLM\SOFTWARE\WIBU-SYSTEMS\CodeMeter\Server\CurrentVersion</code>
Windows	<code>%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini</code>
Mac OS	<code>/Library/Preferences/com.wibu.CodeMeter.Server.ini</code>
Linux	<code>/etc/wibu/CodeMeter/Server.ini</code>
Solaris	<code>/etc/opt/CodeMeter/Server.ini</code>

Die Konfiguration umfasst zwei Schritte:

- bearbeiten bestehender Einträge,
- erstellen neuer Einträge für neue CodeMeter® Netzwerk-Server wenn erforderlich und Festlegen einer Reihenfolge, wie auf Client-Anfragen geantwortet wird über eine Serversuchliste..

Bestehende Einträge

1. Setzen des Eintrages "`IsWanServer`" auf einen Wert von "1". Standardmäßig ist der Wert auf "0" gesetzt, d.h. CmWAN ist deaktiviert..
2. Der Standard-Portnummer unter der ein CodeMeter-Server CmWAN-Requests entgegennimmt ist "22351". Über den Eintrag "`HttpPort`" kann ein anderer Port festgelegt werden, falls die Reverse Proxy-Einrichtung es erfordert

 Achten Sie in diesem Fall darauf, dass dieser nicht der Port-Nummer entspricht, die für die "normale" Netzwerkkommunikation verwendet wird. Diese Portnummer entnehmen Sie dem Eintrag "`NetworkPort`".

Anzulegende Einträge

Für jeden neuen CodeMeter® Netzwerkserver ist ein zusätzlicher Eintrag zu den bestehenden hinzuzufügen.

Diese Beschreibung bezieht sich auf die Bearbeitung der `server.ini`-Datei. In der Windows Registry müssen Sie neue Schlüssel und Zeichenfolgen anlegen.

 Derzeit unterstützt CodeMeter® eine Digest Authentication auf der Client-Seite. Es ist geplant, dass zukünftige CodeMeter®-Versionen auch die Verwendung von Client-Zertifikaten unterstützen.

Navigieren Sie zum Eintrag "`ServerSearchList`". Alle Server-Einträge müssen unterhalb dieses Eintrags bestehen.

Beim Anlegen eines neuen Eintrages zur Verwendung einer Digest Authentication definieren Sie die Parameter `Address`, `User` und `Password`.

```
[ServerSearchList]  
[ServerSearchList\Server1]
```

```
Address=https://cmwanserver.example.org  
User=user123  
Password=...
```

Beim Anlegen eines neuen Eintrages zur Verwendung eines Client-Zertifikates definieren Sie die Parameter **Address**, **User** und **Certificate**.

```
[ServerSearchList\Server2]  
Address=https://cmwanserver.example.org  
User=user456  
Certificate=...
```

Nach dem Anlegen der neuen Server-Einträge definieren Sie eine Server-Suchliste, d.h. eine Reihenfolge dieser und der bereits bestehenden Server-Einträge, in der auf Client-Anfragen geantwortet wird.

 Wenn ein oder mehrere *CmWAN* Server-Einträge existieren, wird derzeit keine automatische Suchfunktion der Server-Suchliste ausgeführt. D.h., wenn *CmWAN* und *CmLAN* genutzt wird, müssen alle LAN-Server in der gewünschten Reihenfolge aufgelistet werden.

12 Handbuch

Die folgenden Teile des Entwicklerhandbuchs über Installation und Handhabung vieler Werkzeuge sind auch für Administratoren von Interesse und sind deswegen in einem separaten Handbuch abgelegt.

12.1 Wichtige erste Informationen

Erstes Anschließen des *CmDongle*

Stecken Sie Ihren *CmDongle* an eine freie USB-Schnittstelle Ihres PCs. Die Leuchtdiode des *CmDongles* leuchtet ca. 1-2 Sekunden abwechselnd rot und grün. Ihr PC zeigt an, dass ein neues USB-Gerät gefunden wurde. Bei *CmDongles* mit zusätzlichem Flash-Speicher, z.B. dem *CmStick/M*, können beliebige Daten permanent in diesem dann angezeigten Laufwerk abgelegt werden. Alternativ zur Massenspeicher-Anzeige (Mass Storage Device) ist auch eine Anmeldung am System als HID (Human Interface Device) möglich; dann wird kein Laufwerk angezeigt (mehr Information siehe [hier](#)⁵¹⁰).

 Bei *CmDongles* ohne Speicher ist dieses Laufwerk rein virtuell, d.h. darauf abgelegte Daten gehen nach dem Abziehen des *CmDongles* verloren!

Der *CodeMeter® Lizenzserver* (Runtime Server) wird standardmäßig unter Windows als Dienst bzw. Daemon (Linux, Mac) installiert und demzufolge bei jedem Systemstart automatisch gestartet. Das Verhalten beim Systemstart ist über die Verwendung von Standardwerten optimiert und verhindert möglicherweise auftretende Prozesszugriffskonflikte. Sollten dennoch Probleme auftreten, kontaktieren Sie bitte den WiBu-Systems Support.

Sollte *CodeMeter® Runtime Server* nicht aktiv sein, kann er auch [manuell gestartet oder gestoppt](#)⁴⁴³ werden.

Betriebssystem	Menüsteuerung	Name
 Windows	[Start Alle Programme CodeMeter CodeMeter Kontrollzentrum]	CodeMeter.exe
 Mac OS	[Programme CodeMeter CodeMeter Kontrollzentrum]	CodeMeterMacX
 Linux	[Anwendungen System CodeMeter Kontrollzentrum] bzw. [Anwendungen Zubehör CodeMeter Kontrollzentrum]	CodeMeterLin
 Sun Solaris	[/opt/CodeMeter/CodeMeterCC]	CodeMeterSun

Im *CodeMeter Kontrollzentrum*, dass Sie über das *CodeMeter®*-Symbol neben der Uhr oder über "**Start | Alle Programme | Codemeter | CodeMeter Control Center**" öffnen können, sehen Sie nun die Seriennummer des *CmDongles*.

Aktivieren von *CmActLicense*-Lizenzen

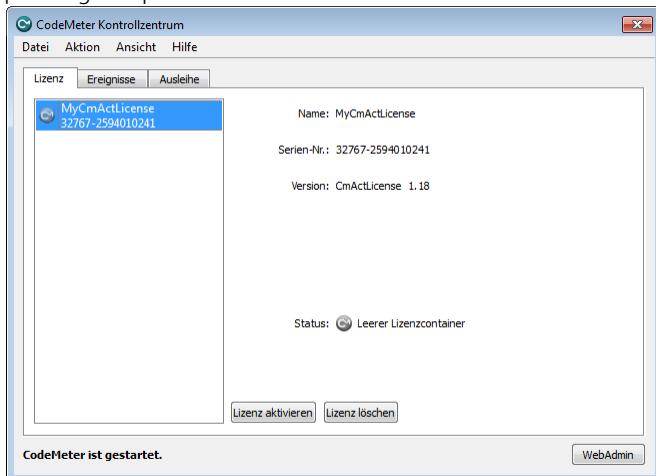
CmActLicense, die software- und aktivierungsbasierte *CodeMeter®*-Variante benötigt keine Hardware. Vielmehr sind *CmActLicense*-Lizenzen an Hardware-Eigenschaften des PCs gebunden, auf dem sie verwendet werden.

 Führen Sie daher die Aktivierung der *CmActLicense*-Lizenz unbedingt auf dem PC durch, für den Sie die Lizenz verwenden wollen.

Bevor Sie *CmActLicense*-Lizenzen für Ihren PC aktivieren können, benötigen Sie eine separate Datei, die

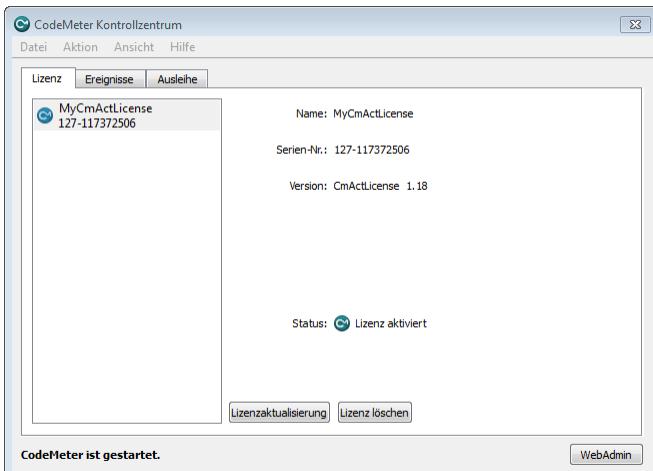
Sie von Ihrem Software-Hersteller erhalten. Diese Lizenzinformationsdatei entspricht einem leeren Lizenz-Container. Er dient dazu, Informationen der Hardware-Eigenschaften des PCs in einer Art "Fingerabdruck" aufzunehmen. Gehen Sie dazu wie folgt vor:

- Ziehen Sie die vom Software-Hersteller erhaltene *.wbb-Datei, z.B.  MyCmActLicense.wbb, per drag&drop in das *CodeMeter Kontrollzentrum*.



Das Status-Feld zeigt an, dass es sich bei dieser Datei lediglich um einen leeren Lizenzcontainer und keine Lizenz handelt. Gleichzeitig ändert sich das *CodeMeter®*-Symbol auf rot.

- Klicken Sie die "**Lizenz aktivieren**"-Schaltfläche, um eine Lizenzanforderungsdatei zu erstellen (siehe [hier](#)⁴⁵⁷) und an den Software-Hersteller zu senden.
Sie erhalten danach vom Software-Hersteller eine Lizenzaktualisierungsdatei.
- Ziehen Sie die vom Software-Hersteller erhaltene *.WibuCmRaU-Datei, z.B.  MyCmActLicen-seRaU.wbb, per drag&drop in *CodeMeter Kontrollzentrum*.



Das Status-Feld zeigt an, dass die Lizenz aktiviert wurde. Gleichzeitig hat die Lizenz eine Seriennummer erhalten und das *CodeMeter®*-Symbol hat auf aktiviert gewechselt.

CodeMeter FAQ

Einen umfangreichen FAQ-Bereich zum Thema *CodeMeter®* und zu verschiedenen Zusatzprodukten finden Sie im Internet auf unserer [CodeMeter Support-Seite](#).

Bitte sehen Sie sich zunächst die Einträge der *CodeMeter®* Support-Seite an, bevor Sie unser Support-Team kontaktieren, da Sie dort schnelle Antworten zu den am häufigsten auftretenden Fragen und Problemen erhalten.

Support

Sie besitzen mehrere Möglichkeiten, uns zu erreichen:

E-Mail	Schreiben Sie uns eine e-Mail an support@codemeter.com . Bitte beschreiben Sie das Problem möglichst genau und fügen Sie die Datei <i>CmDust-Result.log</i> hinzu, die mit CmDust ⁵⁰⁰ erstellt worden ist.
Telefon	Sie erreichen uns werktags (Baden-Württemberg-Ferientagregelung) (Montag bis Freitag) durchgehend von 8.00 bis 17.00 unter der Telefonnummer 0721-93172-15 (<i>CodeMeter®</i> -Hotline).

12.2 Installation

Der folgende Abschnitt enthält Installations- und Deinstallationsinformationen der *CodeMeter®*-Laufzeitumgebung (*CodeMeter®* Runtime Kit) für verschiedene Betriebssysteme.

- [Windows 32-Bit/64-Bit](#)⁴³⁶
- [Mac OS X](#)⁴³⁸
- [Linux](#)⁴⁴⁰
- [Sun Solaris](#)⁴⁴¹

12.2.1 Installation unter 32/64-Bit Windows

Für Windows 32- und 64-Bit steht Ihnen ein CodeMeter® Runtime Kit-Installationsprogramm zur Verfügung (CodeMeterRuntime32.exe, CodeMeterRuntime64.exe):

12.2.1.1 Installierte Dateien unter 32/64-Bit Windows

Die Dateien, die das CodeMeter® Runtime Installation Kit auf Ihren Rechner installiert hat, finden Sie im Installationsverzeichnis üblicherweise unter [%Program Files%\CodeMeter\Runtime\bin]).

Insgesamt gilt für 32-Bit Windows die folgende Ablagestruktur:

```
%ProgramFiles%
`--CodeMeter
  |-- Backup (ab Version 4.30 ein Shortcut)
  |-- Logs (ab Version 4.30 ein Shortcut)
  `-- Runtime
    |-- bin
      |-- CodeMeter.exe
      |-- CodeMeter.l*
      |-- CodeMeterCC.exe
      |-- CodeMeterCC.l*
      |-- CodeMeterZZ.wbb - WebAdmin
      |-- CmRmtAct32.*
      |-- cmu32.exe
      |-- WibuCmId32.*
      |-- WibuCmTrigger32.*
    |-- help
      '-- CmUserHelp
%WINDIR%
`-- System32
  |-- WibuCm32.lxx
  |-- WibuCm32.dll (CodeMeter Calling Driver)
  |-- WibuCmJni.dll
  '-- WibuXpm4J32.dll
%ProgramData%
`-- CodeMeter
  |-- Backup
  |-- Logs
```

Insgesamt gilt für 64-Bit Windows die folgende Ablagestruktur:

```
%ProgramFiles(x86)%
`--CodeMeter
  |-- Backup (ab Version 4.30 ein Shortcut)
  |-- Logs (ab Version 4.30 ein Shortcut)
  `-- Runtime
    |-- bin
      |-- CodeMeter.exe
      |-- CodeMeter.l*
      |-- CodeMeterCC.exe
      |-- CodeMeterCC.l*
      |-- CodeMeterZZ.wbb - WebAdmin
      |-- CmRmtAct32.*
      |-- cmu32.exe
      |-- WibuCmId32.*
      |-- WibuCmTrigger32.*
```

```

        '-- help
        '-- CmUserHelp
%ProgramFiles%
    '--CodeMeter
        '-- Runtime
            '-- bin
                |-- WibuCmId64.dll
                |-- WibuCmTrigger64.*

%WINDIR%
    '-- SysWOW64
        |-- WibuCm32.lxx
        |-- WibuCm32.dll (CodeMeter Calling Driver)
        |-- WibuCmJni.dll
        '-- WibuXpm4J32.dll
    '-- System32
        |-- WibuCm64.lxx
        |-- WibuCm64.dll (CodeMeter Calling Driver)
        |-- WibuCmJni64.dll
        '-- WibuXpm4J64.dll

%ProgramData%
    '-- CodeMeter
        |-- Backup
        '-- Logs

```

Die nachfolgende Tabelle gibt einen Auszug an installierten Dateien.

Datei	Beschreibung
CodeMeter.exe	Prozess des <i>CodeMeter Lizenzservers</i>
CodeMeter.l**	Sprachdateien für <i>CodeMeter.exe</i>
CodeMeterCC.exe	Prozess des <i>CodeMeter Kontrollzentrums</i>
CodeMeterCC**.qm	Sprachdateien für <i>CodeMeter Kontrollzentrum</i>
cmu32(64).exe	Prozess des cmu-Kommandozeilenprogramm.
CmRmtAct32(64).dll	Dynamic Link Library (DLL), wird von <i>CodeMeter.exe</i> zum Lizenzupdate benötigt.
CmRmtAct32(64).l**	Sprachdateien für das Lizenzupdate.
CodeMeterXX.wbb	<i>CodeMeter WebAdmin</i> in verschiedenen Sprachvarianten.
WibuCm32(64).dll	Beinhaltet alle <i>CodeMeter® API</i> Funktionen. Diese DLL muss auf allen PCs installiert sein, die eine <i>CodeMeter®</i> geschützte Anwendung benutzen wollen; Installationspfad: [\Windows\System32].
WibuCm32(64).lXX	Sprachdateien für die <i>WibuCm32(64).dll</i> ; Installationspfad: [\Windows\System32].
WibuCmTrigger32(64).dll	Wird von Microsoft Internet Explorer benötigt.
WibuCmTrigger32(64).lXX	Sprachdateien für die <i>WibuCmTrigger32(64).dll</i> .
CmUserhelp*.*	<i>CodeMeter® Online Hilfe</i> in verschiedenen Sprachen. Installationspfad [%CodeMeter%\Runtime\help].

12.2.1.2 Deinstallation unter 32/64-Bit Windows

1. Wählen Sie in der Windows Systemsteuerung die Option "Software".
 2. Wählen Sie den Eintrag "CodeMeter Runtime Kit" und die Option "Entfernen".
- Alle CodeMeter® Dateien, die im Installationspaket enthalten waren und Registry-Einträge werden gelöscht. Lediglich das Log- und Backup-Verzeichnis verbleiben.

12.2.2 Installation unter Mac OS Betriebssystemen

Für Mac OS X 10.3-10.5 steht Ihnen ein einheitliches CodeMeter® Runtime Kit-Installationsprogramm zur Verfügung:

Datei	Beschreibung
CmRuntimeUser.dmg	installiert alle benötigten CodeMeter® Runtime Komponenten

1. Führen Sie die Datei `CmRuntimeUser.dmg` aus, um das CodeMeter® Runtime Kit zu installieren.
2. Wählen Sie im neuen Verzeichnis `CmRuntime` die Datei `CmInstall.mpkg` aus und folgen Sie den Anweisungen des Installationsassistenten.

12.2.2.1 Installierte Dateien unter Mac OS

Insgesamt gilt für Mac OS die folgende Ablagestruktur:

```

/
  -- Applications
    |   -- CodeMeter.app
    |   |   -- CmUserHelp
    |   |   |   ...
    |   |   -- CodeMeterCn.wbb
    |   |   -- CodeMeterDe.wbb
    |   |   ...
    |   |   -- CodeMeterMacX
    |   |   -- CodeMeterUs.wbb
    |   -- Contents
    |       |   -- Info.plist
    |       |   -- MacOS
    |       |       |   -- CodeMeterCC
    |       |       |   -- CodeMeterCC_de.qm
    |       |       |   ...
    |       |   -- Resources
    |       |       |   -- CodeMeterCC.icns
    |       |       |   -- com.wibu.CodeMeter.Server.ini
    |       |       |   -- English.lproj
    |       |       |   ...
    |       |       |   -- zh_TW.lproj
    |   -- PkgInfo
  -- Library
    |   -- Application Support
    |       |   -- CodeMeter
    |           |   -- Backup
    |           |   -- CmAct
    |   -- Frameworks
  
```

```

    '-- WibuCmMacX
  |-- Logs
    '-- CodeMeter
  -- Preferences
    '-- com.wibu.CodeMeter.Server.ini (permissions
                                              -rw-rw-rw-)
  -- Java
    '-- Extensions
      '-- libwibuKJni.jnilib
-- System
  '-- Library
    '-- Extensions
      '-- CmUSBMassStorage.kext
        '-- Resources
          '-- CodeMeter.icns
    '-- PreferencePanes
      '-- CodeMeter.prefPane
-- usr
  '-- bin
    '-- cmu

```

Die nachfolgende Tabelle gibt einen Auszug an installierten Dateien.

Datei	Beschreibung
CodeMeterMacX	[Programme/CodeMeter.app]; des CodeMeter Lizenzserver Prozesses.
CodeMeterXX.wbb	[Programme/CodeMeter.app]; CodeMeter WebAdmin in verschiedenen Sprachvarianten.
CodeMeterUserhelp	[Programme/CodeMeter.app/help]; CodeMeter® Endbenutzerhilfe.
CodeMeterCC	[Programme/CodeMeter.app/contents]; CodeMeter Kontrollzentrum.
CodeMeterCC**.qm	[Programme/CodeMeter.app//contents/resources]; Sprachdateien für CodeMeter Kontrollzentrum.
Cmu	Das cmu-Kommandozeilenprogramm.
WibuCmMacX	[Library/Frameworks/WibuCmMacX.framework]; beinhaltet alle CodeMeter® API Funktionen.
CodeMeterMacX	[Library/StartupItems]; das CodeMeter Lizenzserver Startup Item.
libwibuKJni.jnilib	[Library/Java/extensions]; die CodeMeter® Java Erweiterung.
com.wibu.CodeMeter.Server.ini	[Library/Preferences]; beinhaltet "Profil Grundeinstellungen" für CodeMeterMacX.
CodeMeter.prefPane	[System/Library/PreferencePanes] enthält die Systemsteuerung für CodeMeterMacX.

Starten des WebAdmin

Sie starten CodeMeter WebAdmin in Mac/Linux:

- über die "**Web Admin**" -Schaltfläche im "**CodeMeterGUI**" -Tool
- direkt in Ihrem Internet Browser, wenn Sie die URLs: <http://localhost:22350> oder <http://127.0.0.1:22350> eingeben.

12.2.2.2 Deinstallation unter Mac OS

Um das **CodeMeter®** Runtime Kit zu deinstallieren:

1. Öffnen Sie erneut das Disk-Image `CmRuntimeUser.dmg`.
2. Starten Sie im Verzeichnis `CmRunTime` das Programm `CmUninstall.mpkg` und folgen Sie den Anweisungen des Programmes (in der Kommandozeile verwenden Sie den folgenden Befehl: `$ sudo installer -pkg /Volumes/CmRuntimeUser/CmUninstall.mpkg -target`. Bitte beachten Sie eventuell abweichende Pfadangaben).

12.2.3 Installation unter Linux Betriebssystemen

Für Linux Betriebssysteme stehen Ihnen verschiedene Installationspakete in den gängigen Formaten zur Verfügung:

Datei	Beschreibung
<code>CodeMeter-[CodeMeter-Version].[Packetnummer].i386.rpm</code>	Basistreiber 32-Bit im RPM-Format (Red Hat Package Manager Format) (z.B.: Suse 9x,)
<code>CodeMeter-[CodeMeter-Version].[Packetnummer]_i386.deb</code>	Basistreiber 32-Bit im DEB-Format gcc3 basiert (z.B.: Debian 3.0, Ubuntu 6.06)
<code>CodeMeter64-[CodeMeter-Version].[Packetnummer].x86_64.rpm</code>	Treibererweiterung 64-Bit im RPM-Format (Red Hat Package Manager Format) (z.B. Suse, RHEL, FC)
<code>CodeMeter64-[CodeMeter-Version].[Packetnummer].amd64.deb</code>	Treibererweiterung 64-Bit im DEB-Format (z.B. Debian, Ubuntu)

Um *CodeMeter Lizenzserver* zu installieren:

1. Wählen Sie sich das gewünschte Installationspaket aus, und.
2. Installieren Sie es wie gewohnt, z.B. Shell-Kommando oder entsprechende Hilfsprogramme.

rpm-Pakete: [rpm -ivh `CodeMeter-[CodeMeter-Version].[Paketnummer].i386.rpm`]

deb-Pakete: [dpkg -i `CodeMeter-[CodeMeter-Version].[Paketnummer]_i386.deb`]

Insgesamt gilt für Linux die folgende Ablagestruktur:

```
/ 
  -- etc
    |-- hotplug
    |   '-- usb
    |       '-- codemeter
    |-- init.d
    |   '-- codemeter
    |-- udev
    |   '-- rules.d
    |       '-- 52-codemeter.rules
    '-- wibu
        '-- CodeMeter
            |-- CmFirm.wbc      (permissions -rw-rw-rw-)
            '-- Server.ini      (permissions -rw-rw-rw-)
  '-- usr
      |-- bin
```

```

    |-- CodeMeterCC
    |-- CodeMeterLin
    |-- cmu
    |-- codemeter-info      (permissions -rwsr-xr-x)
-- lib
-- share
    |-- CodeMeter
        |-- CodeMeterCC
        |-- CodeMeterDe.wbb
        |-- ..
        |-- CodeMeterLin
        |-- CodeMeterUs.wbb
        |-- WibuCmSTrigger.jar
        |-- codemeter.rc   (copy of /etc/init.d/codemeter)
        |-- getpath.class
        |-- libWibuCmWebLin.so  ../../lib/libWibuCmWebLin.so
    |-- applications
        '-- codemeter.desktop
    |-- doc
        '-- CodeMeter
    |-- man
    |-- pixmaps
        '-- codemeter.png
-- var
    |-- lib
        '-- CodeMeter
            |-- Backup
            '-- CmAct
    '-- log
        '-- CodeMeter

```

12.2.3.1 Deinstallation unter Linux

Führen Sie das entsprechende Shell-Kommando zum Deinstallieren des *CodeMeter® Runtime Kits* aus:

- auf RPM basierten Distributionen (wie Suse/RedHat/Fedora) [rpm -e CodeMeter]
- auf DEB basierten Distributionen (wie Debian/Ubuntu) [dpkg -r CodeMeter]

12.2.4 Installation unter Sun Solaris Betriebssystemen

Für Sun Solaris 10 und 11 stehen Ihnen zwei Installationspakete zur Verfügung:

Datei	Beschreibung
codemeter_4.0-sol-sparc.tar.bz2	installiert alle benötigten <i>CodeMeter® Runtime Komponenten</i> auf Ihrer Sun mit SPARC-Architektur
codemeter_4.0-sol-i386.tar.bz2	installiert alle benötigten <i>CodeMeter® Runtime Komponenten</i> auf Ihrer Sun mit I386-Architektur
codemeter_4.40-sol-SPARCV9.tar.bz2	installiert 64-Bit-Erweiterungen (siehe WIBUCM64 im Baum).
codemeter_4.40-sol-x64.tar.bz2	

Um das *CodeMeter® Runtime Kit* zu installieren gehen Sie bitte wie folgt vor:

1. Entpacken Sie das Paket mit: [bunzip2 codemeter_4.0-sol-(sparc).tar.bz2].

2. Entpacken Sie das tar-Archiv mit: [tar xvf codemeter_4.0-sol-(sparc).tar].

3. Installieren Sie das Paket mit dem Kommando: [pkgadd WIBUcm].

Insgesamt gilt für Solaris die folgende Ablagestruktur:

```
/  
  -- etc  
    `-- opt  
      `-- CodeMeter  
        `-- Server.ini  
  -- lib  
    `-- svc  
      `-- method  
        `-- codemeter  
  -- opt  
    `-- CodeMeter  
      |-- CodeMeterCC  
      |-- CodeMeterCn.wbb  
      |-- CodeMeterDe.wbb  
      |-- CodeMeterFr.wbb  
      |-- CodeMeterJp.wbb  
      |-- CodeMeterSun  
      |-- CodeMeterUs.wbb  
      |-- Tools  
        |-- cmdust.sh  
        |-- cmu  
        `-- cmu64 (part of package WIBUcm64)  
      |-- codemeter.png  
      |-- help  
        |-- CmUserHelp  
          |-- de/  
          |-- us/  
  -- usr  
    |-- lib  
      |-- libwibucmJNI.so  
      |-- libwibuxpm4j.so  
        `-- 64 (part of package WIBUcm64)  
          |-- libwibucmsun.so  
          |-- libwibucmJNI.so  
          |-- libWibuCmWebsun.so  
          |-- libwibuxpm4j.so  
    `-- share  
      `-- applications  
        `-- codemeter.desktop  
  -- var  
    |-- log  
      `-- CodeMeter/  
    `-- svc  
      `-- manifest  
        |-- device  
        `-- codemeter.xml
```

12.2.4.1 Deinstallation unter Sun Solaris

Führen Sie das entsprechende Shell-Kommando zum Deinstallieren des *CodeMeter® Runtime Kits* aus:
[pkgrm WIBUcm]

12.3 CodeMeter Kontrollzentrum

CodeMeter Kontrollzentrum dient dazu, lokale Konfigurationseinstellungen für *CodeMeter Lizenzserver* vorzunehmen. *CodeMeter Lizenzserver* ist softwareseitig als Laufzeitumgebung das Herz von *CodeMeter®*. Er ermöglicht den Zugriff auf *CmContainer*. *CmContainer* können hierbei sowohl lokal, als auch im Netzwerk angeschlossen sein. *CodeMeter Lizenzserver* ist standardmäßig als Dienst bzw. Daemon (Linux, Mac) installiert und wird daher bei jedem Systemstart automatisch gestartet.

Ist der Dienst gestartet, so können andere Programme auf Lizenzen zugreifen, die im *CmContainer* gespeichert sind, und geschützte Datenbereiche im *CmContainer* verwenden.

Betriebs-System	Menüsteuerung
 Windows	[Start Alle Programme CodeMeter CodeMeter Kontrollzentrum]
 Mac OS	[Programme CodeMeter CodeMeter Kontrollzentrum]
 Linux	[Anwendungen System CodeMeter Kontrollzentrum] bzw. [Anwendungen Zubehör CodeMeter Kontrollzentrum]
 Sun Solaris	[/opt/CodeMeter/CodeMeterCC]



CodeMeter Lizenzserver kann auf jedem Rechner nur einmal gestartet werden!

Starten und Stoppen des *CodeMeter®-Dienstes* oder *Daemon*

Die folgende Tabelle zeigt Ihnen wie Sie den *CodeMeter®-Dienst* für verschiedene Betriebssysteme starten bzw. stoppen.

Betriebs-System	Beschreibung
 Windows	<ol style="list-style-type: none"> 1. Navigieren über "Windows Systemsteuerung Verwaltung Dienste" auf <i>CodeMeter Runtime Server</i>. 2. Rechter Mausklick und 'Starten' oder 'Beenden' des Dienstes. Alternativ über das "Aktion"-Menü des <i>CodeMeter Kontrollzentrums</i>.

Betriebs-System	Beschreibung
Mac OS	<p>1. Navigieren über "Systemeinstellungen Sonstige" zum CodeMeter® Icon.</p>  <p>2. Klicken auf das CodeMeter® Icon. Der folgende Dialog erscheint.</p> 
Linux	<p>3. Klicken auf die "Dienst Stoppen" bzw. "Dienst Starten"-Schaltfläche zum Stoppen bzw. Starten des Dienstes.</p> <p>1. Zum Stoppen des Dienstes Aufrufen des folgenden Skriptes mit 'sudo' Root-Rechten:</p>

Betriebs-System	Beschreibung
Sun Solaris	<pre>etc/init.d/ codemeter stop 2. Zum Starten des Dienstes Aufrufen des folgenden Skriptes mit 'sudo' Root-Rechten: etc/init.d/ codemeter start</pre> <ol style="list-style-type: none"> Zum Stoppen des Dienstes Aufrufen des folgenden Kommandos als 'root': %> svcadm disable codemeter Zum Starten des Dienstes Aufrufen des folgenden Kommandos als 'root': %> svcadm enable codemeter Stoppen des Dienstes bis zum nächsten Hochfahren Aufrufen des folgenden Kommandos folgenden Kommandos als 'root': %> svcadm disable -t codemeter

 *CodeMeter Lizenzserver verwendet zur Kommunikation das TCP/IP Netzwerkprotokoll und den voreingestellten Port 22350. Dieser Port darf also nicht von Ihrer Firewall geblockt werden. Sorgen Sie dafür, dass der benutzte IP-Port 22350 für CodeMeter® frei verwendbar ist, d.h. geben Sie die Kommunikation für diesen IP-Port in Ihrer Firewall frei.*

12.3.1 Struktur und Navigation

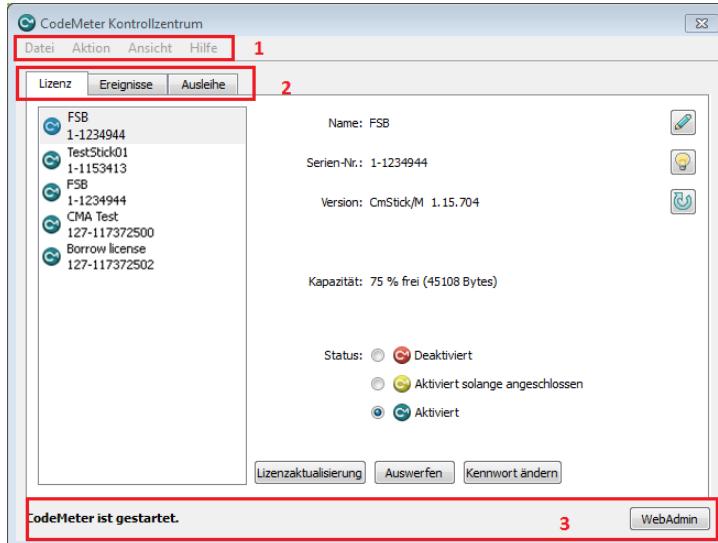


Abbildung 200: CodeMeter Kontrollzentrum - Übersicht

Die CodeMeter Kontrollzentrum-Benutzeroberfläche teilt sich in drei separate Bereiche auf:

- [Menüleiste](#) (1)
- Karteireiter-Bereiche (2)
- [Status und Öffnen von CodeMeter WebAdmin](#) (3).

Starten CodeMeter Kontrollzentrum

CodeMeter Kontrollzentrum erreichen Sie auf mehreren Wegen:

Öffnen

- Doppelklick auf die CodeMeter®  oder  Symbole im Infobereich der Windows Task-Leiste
- Rechter Mausklick auf die CodeMeter®  oder  Symbol dort, und nachfolgende Auswahl des "Anzeigen" Menü-Eintrages.

Das CodeMeter Kontrollzentrum Kontextmenü (rechte Maustaste auf das CodeMeter Symbol) bietet Ihnen die folgenden weiteren Einträge:

Eintrag	Beschreibung
WebAdmin	Startet CodeMeter WebAdmin im Standard Internet Browser.
CmDongle(s) auswerfen	Option zum sicheren Entfernen von CmDongles.
CmDongle deaktivieren	Aufforderung, das Kennwort einzugeben.
Hilfe	Öffnet die CodeMeter® Hilfe.
Über	Zeigt generelle Informationen über die benutzte CodeMeter® Komponenten an.
Beenden	Beendet CodeMeter Lizenzserver.

- Navigation über den "Start | Alle Programme | CodeMeter Kontrollzentrum" Eintrag des Systemmenüs.

Im Infobereich der Windows-Task-Leiste repräsentieren dabei unterschiedliche Farben der CodeMeter®-Symbole die Status-Zustände der verbundenen CmContainer.

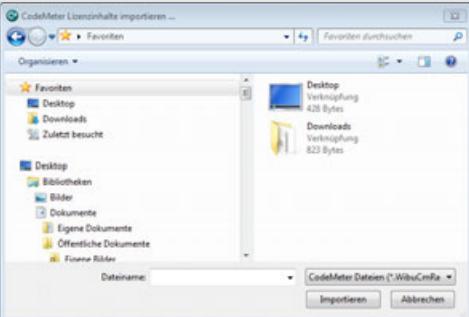
Farbe	Status
Grau 	Es ist kein CmContainer verbunden oder CodeMeter Lizenzserver ist nicht gestartet.
Grün 	Es ist ein aktiver CmContainer verbunden.
Blau  doppelt	Es sind mehrere CmContainer verbunden und aktiviert bis sie abgezogen oder deaktiviert werden.
Gelb 	Es ist ein CmDongle verbunden und aktiviert bis er abgezogen wird.
Rot 	Es ist ein deaktiver CmContainer verbunden.

Abbildung 201: CodeMeter® Symbole Windows-Task-Leiste

12.3.2 Menüleiste

Das Datei-Menü

Element	Beschreibung
Lizenz importieren	Um über CodeMeter Kontrollzentrum Lizenzinhalte zu importieren, gehen Sie wie folgt vor: <ol style="list-style-type: none"> Wählen Sie den  "Datei Lizenz importieren..."-Eintrag. Wählen Sie im nachfolgenden "CodeMeter Lizenzinhalte importieren ..." -Dialog die CodeMeter®-Dateien vom Typ *.wibuCmRaU; *.wbb; *.wbc aus und lesen Sie die Lizenzdaten über die "Importieren"-Schaltfläche ein.

Element	Beschreibung
	 <p>Abbildung 202: CodeMeter Kontrollzentrum – Lizenzinhalte importieren</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Alternativ können Sie die Lizenzdatei auch direkt aus dem Windows Explorer per Drag & Drop in den Lizenz Karteireiter ziehen. </div>
WebAdmin 	Öffnet CodeMeter WebAdmin im Standard Internet Browser. Alternativ dazu können Sie die Tastenkombination <STRG>+W drücken.
Protokollierung 	<p>Speichert alle CodeMeter® Ereignisse in eine Protokollierungsdatei. Alternativ dazu können Sie die Tastenkombination <STRG>+L drücken.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Die Aktivierung der Protokollierung wirkt sich auch auf die Anzeige des Protokolls in CodeMeter WebAdmin auf der ""Diagnose Protokoll"" Seite aus. </div> <p>Unter Windows ist die Protokollierungsdatei ist im Verzeichnis %\Program Files%\CodeMeter\Logs abgelegt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> Diese Protokollierungsdatei ist vor allem bei Problemanalysen sehr von Vorteil. </div>
Einstellungen	Öffnet CodeMeter WebAdmin und ist voreingestellt auf die Seite, auf der Sie Netzwerkeinstellungen vornehmen können.
Beenden 	Beendet CodeMeter Kontrollzentrum. Alternativ dazu können Sie die Tastenkombination <STRG>+Q drücken.
	<div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> CodeMeter Lizenzserver wird dadurch nicht beendet. </div>

Das Aktion-Menü

Element	Beschreibung
Alle CmDongles auswerfen 	Wirft alle angeschlossenen CmDongles in einem Vorgang aus. Alternativ dazu können Sie die Tastenkombination <STRG>+ALT+Q drücken.
Lizenzspeicher defragmentieren 	Defragmentiert den Lizenzspeicher des ausgewählten CmDongles. Alternativ dazu können Sie die Tastenkombination <STRG>+ALT+D drücken.
Zeitzertifikate aktualisieren 	Aktualisiert die Zeitzertifikate des ausgewählten CmContainers. Alle Zeitstempel werden aktualisiert.

Element	Beschreibung
CodeMeter Dienst starten 	<p>Startet den Dienst <i>CodeMeter Lizenzserver</i> neu.</p> <p>Diesen Menü-Eintrag können Sie verwenden, wenn <i>CodeMeter Lizenzserver</i> als Dienst vorher gestoppt wurde, z.B. bei Änderungen von Netzwerkeinstellungen in <i>CodeMeter WebAdmin</i>, die einen Neustart des Dienstes erforderlich machen.</p> <p>Falls Sie Administratorrechte unter Windows besitzen, können Sie den Dienst <i>CodeMeter Lizenzserver</i> auch über die Arbeitsplatz-Verwaltung einstellen (Systemeinstellungen Verwaltung Dienste).</p>
Hardware-Konfiguration reparieren 	<p>Repariert die Hardware-Konfiguration der <i>CmDongle</i>-Bauformen SD-Cards und CF-Cards.</p> <p>Dieses Tool wird benötigt, wenn die <i>CmCard</i>-Hardware nicht in der Lizenzliste des <i>CodeMeter Kontrollzentrums</i> erscheint.</p>
CodeMeter Dienst stoppen 	Beendet den Dienst <i>CodeMeter Lizenzserver</i> .
Das Ansicht-Menü	
Element	Beschreibung
Fenster verstecken	Minimiert und verbirgt das <i>CodeMeter Kontrollzentrum</i> Fenster zurück in den Infobereich der Windows-Task-Leiste. Alternativ dazu können Sie die Tastenkombination <STRG+M> drücken.
Aktualisieren	Aktualisiert die Anzeige aller verbundenen <i>CmContainer</i> . Alternativ dazu können Sie die Taste <F5> drücken.
Schriftgrad vergrößern	Vergrößert die Anzeige im Ereignisse Karteireiter. Alternativ dazu können Sie die Tastenkombination <STRG>++ drücken.
Schriftgrad verkleinern	Verkleinert die Anzeige im Ereignisse Karteireiter. Alternativ dazu können Sie die Tastenkombination <STRG>-- drücken.
Kopiere Ereignisablauf	Kopiert den Ereignisablauf im Ereignisse Karteireiter in die Zwischenablage. Alternativ dazu können Sie die Tastenkombination <STRG>+C drücken.
Lösche Ereignisablauf 	Löscht den Ereignisablauf im Ereignisse Karteireiter. Alternativ dazu können Sie die Tastenkombination <ALT>+C drücken.
Zeige alle verbundenen CmContainer 	Zeigt alle angeschlossenen <i>CmContainer</i> mit Details im Ereignisse Karteireiter an. Alternativ dazu können Sie die Tastenkombination <ALT>+S drücken.
Zeige alle offenen Handles	Zeigt alle offenen Handles im Ereignisse Karteireiter an. Als Referenz ermöglicht

Element	Beschreibung
	chen Handles dem Entwickler weitere Programmierungen.
	Zeigt alle Lizenzinträge im <i>CmContainer</i> im Ereignisse Karteireiter an. Alternativ dazu können Sie die Tastenkombination <ALT>+E drücken.
	Wechselt die Ansicht des Ausleihe Karteireiters zwischen sichtbar und ausgeblendet.

Das Hilfe-Menü

Element	Beschreibung
Hilfe	Öffnet die <i>CodeMeter® Online-Hilfe</i> . Von dort aus erreichen Sie die Hilfe zu <i>CodeMeter Lizenzserver</i> und <i>CodeMeter Kontrollzentrum</i> .
CmDongle registrieren	Öffnet die gesicherte Webseite https://my.codemeter.com zur Registrierung von <i>CmDongles</i> .
Über	Informiert über die gestartete <i>CodeMeter Kontrollzentrum</i> Version.

12.3.3 Lizenz-Karteireiter

Dieser Karteireiter zeigt Ihnen Informationen über verbundene *CmContainer* an, und bietet einige Optionen zur Konfiguration von verbundenen *CmContainer*. Außerdem können Sie hier die über den [CmFAS Assistenten](#)⁴⁵⁵ die Lizenzen aktualisieren, die sich in *CmContainer* befinden.

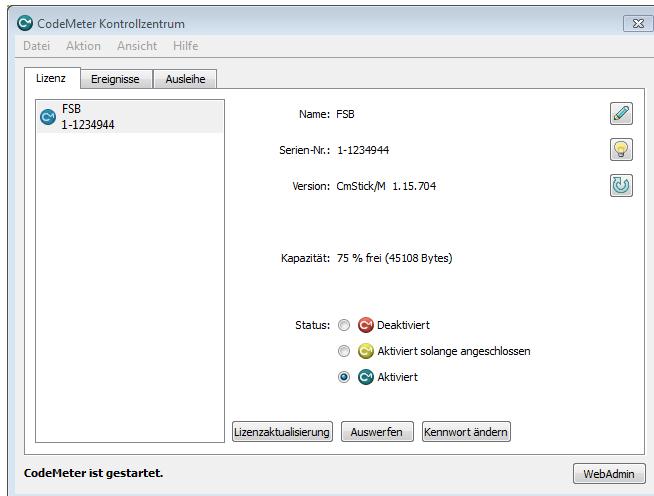
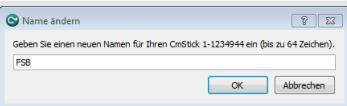


Abbildung 203: CodeMeter Kontrollzentrum – Karteireiter "Lizenz"

Element	Beschreibung
	Ändert den Namen des ausgewählten <i>CmContainers</i> und zeigt diesen an. Im nachfolgenden Dialog können Sie den Namen ändern.

Element	Beschreibung								
	 <p>Abbildung 204: CodeMeter Kontrollzentrum – Name des CmContainers ändern</p>								
	Lässt die LEDs eines ausgewählten CmSticks kurz blinken. Sind mehrere CmSticks angeschlossen, erleichtert dies die Identifikation eines bestimmten CmSticks.								
	<p>Aktualisiert die Firmware des ausgewählten CmDongles. Dies gewährleistet die korrekte Ausführung elementarer Funktionen und behebt gegebenenfalls auftretende Probleme.</p> <p> Um eine Firmware-Aktualisierung durchzuführen, ist eine Internet-Verbindung unbedingt erforderlich. CodeMeter Kontrollzentrum verbindet sich dann automatisch mit dem Firmware Update Server von Wibu-Systems. Sie werden dabei zur Eingabe Ihres CmDongle Kennwortes aufgefordert, um die Aktion zu bestätigen.</p> <p> Die Aktualisierung kann einige Minuten dauern. Bitte ziehen Sie den CmDongle <u>nicht</u> ab bevor der Vorgang beendet ist. Dies kann zu irreparablen Schäden im CodeMeter®-SmartCard Chip führen!</p>								
Kapazität	<p>Dieses Feld informiert über die Kapazität des ausgewählten CodeMeter®-SmartCard Chip eines CmDongles. Die Angaben sind in Prozent sowie in Zahl der absoluten Bytes.</p> <p> Bitte beachten Sie, dass dieser Wert nichts über die Belegung eines möglichen Flash-Speichers eines CmDongles aussagt.</p>								
Status	<p>Der Bereich Status zeigt den Aktivierungsstatus des ausgewählten CmContainer an.</p> <table border="1"> <thead> <tr> <th>Farbe</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td></td> <td>Der verbundene CmContainer ist deaktiviert und kann von keiner lizenzierten Anwendung genutzt werden.</td> </tr> <tr> <td></td> <td>Der CmDongle ist aktiviert solange er angeschlossen ist. Nach dem Entfernen des CmDongle vom PC wird automatisch der Zugriff von lizenzierten Anwendungen deaktiviert.</td> </tr> <tr> <td></td> <td>Der CmContainer ist voll aktiviert. Im Fall eines CmDongles ist der Zugriff von lizenzierten Anwendungen auch nach dem Abziehen eines CmDongles weiterhin möglich.</td> </tr> </tbody> </table> <p> Wibu-Systems <u>empfiehlt</u> die Verwendung des Aktivierungsstatus "Aktiviert solange angeschlossen". Dies stellt sicher, dass bei Verlust des CmDongles Unbefugte nicht auf die Lizenzen und persönliche Daten im CmDongle zugreifen können!</p> <p>Ändern des Aktivierungsstatus</p> <p>Zum Ändern des Aktivierungsstatus gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Klicken in das Auswahlfeld der gewünschten Statusoption. 	Farbe	Status		Der verbundene CmContainer ist deaktiviert und kann von keiner lizenzierten Anwendung genutzt werden.		Der CmDongle ist aktiviert solange er angeschlossen ist. Nach dem Entfernen des CmDongle vom PC wird automatisch der Zugriff von lizenzierten Anwendungen deaktiviert.		Der CmContainer ist voll aktiviert. Im Fall eines CmDongles ist der Zugriff von lizenzierten Anwendungen auch nach dem Abziehen eines CmDongles weiterhin möglich.
Farbe	Status								
	Der verbundene CmContainer ist deaktiviert und kann von keiner lizenzierten Anwendung genutzt werden.								
	Der CmDongle ist aktiviert solange er angeschlossen ist. Nach dem Entfernen des CmDongle vom PC wird automatisch der Zugriff von lizenzierten Anwendungen deaktiviert.								
	Der CmContainer ist voll aktiviert. Im Fall eines CmDongles ist der Zugriff von lizenzierten Anwendungen auch nach dem Abziehen eines CmDongles weiterhin möglich.								

Element	Beschreibung
	<p>2. Eingeben des <i>CmDongle</i>-Kennwortes in den nachfolgenden Dialog.</p> 
	<p>Abbildung 205: <i>CodeMeter Kontrollzentrum</i> – Kennwort eingeben</p> <p>3. Drücken der "OK"-Schaltfläche zur Bestätigung der Statusänderung</p>
Lizenz-Aktualisierung	<p>Über diese Schaltfläche fordern Sie neue, oder aktualisieren bestehende Lizenzen für den ausgewählten <i>CmContainer</i> an. Es öffnet sich der <i>CodeMeter Field Activation Service (CmFAS) Assistent</i> ⁴⁵⁵.</p> 
	<p>Abbildung 206: <i>CodeMeter Kontrollzentrum</i> – <i>CmFAS Assistent</i></p>
Auswerfen	<p>Über diese Schaltfläche geben Sie den ausgewählten <i>CmDongle</i> wieder frei. Der <i>CmDongle</i> meldet sich beim Betriebssystem ab und kann dann sicher vom PC abgezogen werden.</p>
Kennwort ändern	<p>Über diese Schaltfläche können Sie das Kennwort des ausgewählten <i>CmDongles</i> ändern. Füllen Sie die entsprechenden Felder im nachfolgenden "Kennwort ändern"-Dialog aus.</p>

Element	Beschreibung
	<p>Um nun das Kennwort für Ihren CmDongle 1-1153413 zu ändern füllen Sie bitte folgende Datenfelder vollständig aus. Wählen Sie Ja aus, falls Sie statt Ihrem Kennwort Ihr Master-Kennwort verwenden möchten.</p> <p>Master-Kennwort verwenden? <input checked="" type="radio"/> Ja <input type="radio"/> Nein</p> <p>Altes Kennwort: <input type="text"/></p> <p>Neues Kennwort: <input type="text"/></p> <p>Kennwort wiederholen: <input type="text"/></p> <p style="color: red;">* Datenfelder unvollständig *</p> <p>OK Abbrechen</p>

Abbildung 207: CodeMeter Kontrollzentrum – Kennwort ändern

1. Geben Sie im "Altes Kennwort"-Feld das aktuell verwendete *CmDongle* Kennwort ein.
2. Geben Sie im "Neues Kennwort"-Feld das gewünschte neue *CmDongle* Kennwort ein.
3. Geben Sie im "Kennwort wiederholen"-Feld noch einmal das gewünschte neue *CmDongle*-Kennwort ein.



Falls Sie das *CmDongle*-Kennwort vergessen haben, besteht hier die Möglichkeit durch Eingabe des Master-Kennworts ein neues Kennwort zu setzen.

4. Bestätigen Sie die Angaben mit der "OK"-Schaltfläche.
5. Aktivieren Sie die "Master-Kennwort verwenden"-Option und geben Sie im "Altes Kennwort"-Feld Ihr *CmDongle* Master-Kennwort ein.

	<p>Ein Master-Kennwort haben Sie erhalten, wenn Sie sich auf der Webseite my.codemeter.com registriert haben. Sie können die Registrierung über den "Hilfe CmDongle registrieren" Menü-Eintrag vornehmen. Die Registrierung bietet verschiedene Vorteile und dient der Sicherheit beim Einsatz von <i>CodeMeter</i>®. Nur bei erfolgreicher Registrierung kann beim Verlust des eigenen Kennworts ein Master-Kennwort angefordert und der <i>CmDongle</i> bei Verlust gesperrt werden.</p>
--	--

12.3.4 Ereignisse-Karteireiter

Über diesen Karteireiter stehen Ihnen beim Start und während der Laufzeit die folgenden Informationen zur Verfügung:

- Anzahl der verbundenen *CmContainer*
- Anzahl der *CmContainer*-Einträge
- Anzahl der gefundenen Firm Item-Ebenen (Lizenzcontainer)
- Zugriffe auf *CodeMeter Lizenzserver*

Sie konfigurieren die Ansicht der Ereignisliste über den "[Ansicht | ...](#)"-Menü-Eintrag.

Den Inhalt der Ereignisanzeige können Sie über den "[Datei](#) | [Protokollierung](#)"-Menü-Eintrag mit-schreiben lassen.

12.3.5 Ausleihe-Karteireiter

Dieser Karteireiter informiert Sie über ausleihbare Lizenzen im Rahmen der *CodeMeter®-Lizenzausleihe*.
[58](#). Dadurch werden Lizenzen auch nutzbar, wenn sie nicht mit *CodeMeter Lizenzserver* verbunden sind.

Diesen Karteireiter können Sie wahlweise über "[Ansicht | Ausleihe sichtbar](#)" ein- oder ausblenden.

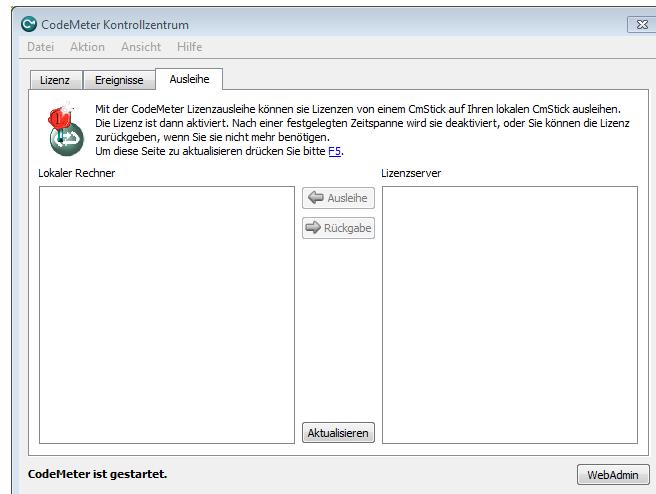


Abbildung 208: *CodeMeter Kontrollzentrum* – Karteireiter "Ausleihe"

Lizenzserver

Auf der rechten Seite werden alle für die Lizenzausleihe verfügbare Lizenzen angezeigt. Die Lizenzen sind nach vorhandenen Lizenzservern, Firm Items und Product Items geordnet. Die angezeigten Lizenzen sind entweder ausleihbar, oder inaktiv.

Sie können nur aktive Lizenzen ausleihen. Sie erkennen aktive Lizenzen am farblich unterlegten Symbol und an der aktivierten "**Ausleihe**"-Schaltfläche.



Abbildung 209: *CodeMeter Kontrollzentrum* – Lizenzen ausleihen

1. Klicken Sie auf die "**Ausleihe**"-Schaltfläche, um Lizenzen, die sich auf dem Lizenzserver befinden für den lokalen PC auszuleihen.

Lokaler PC

Auf der linken Seite werden alle für die Benutzung auf einem lokalen PC vom Lizenzserver ausgeliehenen Lizenzen angezeigt.

Diese Lizenzen werden nach dem definierten Ausleihzeitraum deaktiviert. Sie haben aber auch die Möglichkeit, die ausgeliehenen Lizenzen vor Ablauf der Ausleihdauer vorzeitig wieder zurückzugeben.

1. Klicken Sie auf die "**Rückgabe**"-Schaltfläche, um ausgeliehene Lizenzen zurückzugeben und sie damit wieder für *CodeMeter* Lizenzserver verfügbar zu machen.

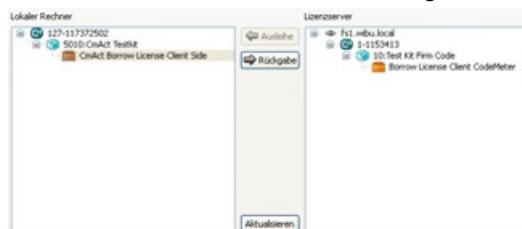


Abbildung 210: *CodeMeter Kontrollzentrum* – Lizenzen zurückgeben

Zur Aktualisierung der Anzeige des Karteireiters drücken Sie die Taste <F5> oder die **Aktualisieren**-Schaltfläche.

12.3.6 Status und Öffnen von *CodeMeter WebAdmin*

Status

Dieser Bereich gibt Ihnen Informationen über den *CodeMeter* Lizenzserver-Status, d.h. ob dieser Dienst gestartet ist oder nicht. Wollen Sie den Status ändern, wählen Sie die "**Aktion | CodeMeter Dienst stoppen**"- bzw. "**Aktion | CodeMeter Dienst starten**"-Menü-Einträge.

WebAdmin

Über diese Schaltfläche öffnen Sie *CodeMeter WebAdmin*. Alternativ können Sie auch den "**Datei | WebAdmin**"-Menü-Eintrag verwenden.

12.4 Einspielen und Aktualisierung von Lizzen

Der [CmFAS Assistent](#)⁴⁵⁵ unterstützt sie beim Einspielen und Aktualisieren von Lizenzdateien für Ihren *CmContainer*.

Über verschiedene Dialoge erstellen Sie manuell Lizenzanforderungen, spielen Lizenzaktualisierungen ein und erzeugen optional Quittungen über diese Vorgänge, die Sie dem Software-Hersteller zusenden können. Die Verwendung von Dateien ermöglicht es auch Lizizen auf einem PC zu aktivieren, der über keinen direkten Internetzugang verfügt. Die untenstehende Abbildung skizziert diesen Vorgang.

Bitte beachten Sie, dass das Einspielen von aktualisierten Lizenzdateien im laufenden Betrieb eines *CmContainers* derzeit nicht unterstützt wird.



Schließen Sie daher vor einer Aktualisierung alle anderen *CodeMeter*[®]-geschützten Anwendungen, die Lizizen aus dem *CmContainer* beziehen, der aktualisiert werden soll und speichern gegebenenfalls Ihre Daten ab.

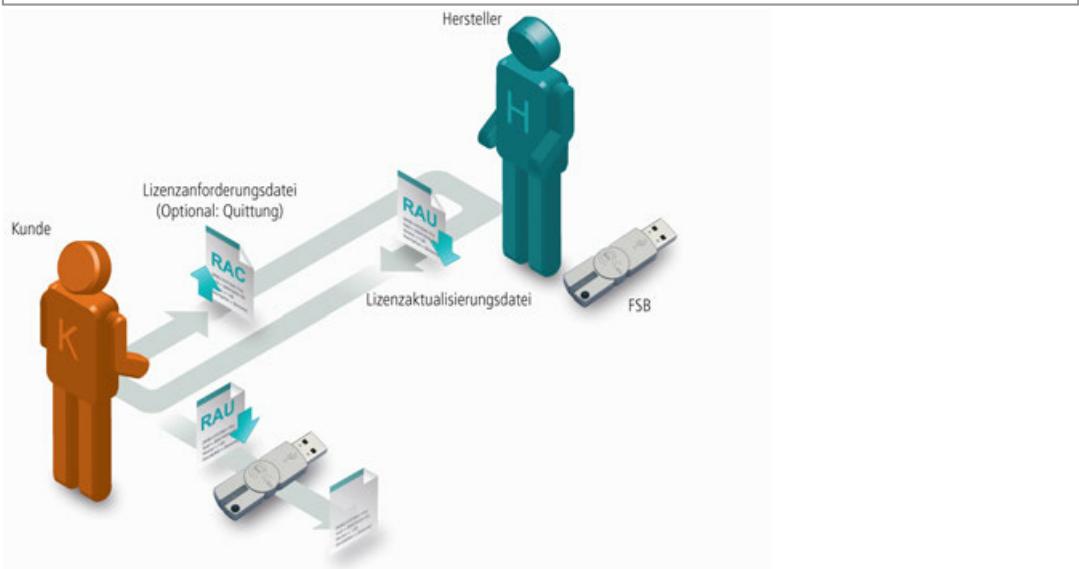


Abbildung 211: *CmFAS* - Dateibasierte Fernaktualisierung

12.4.1 Der CmFAS Assistent im CodeMeter Kontrollzentrum

Öffnen Sie *CodeMeter Kontrollzentrum*. Sollten Sie mehrere *CmContainer* angeschlossen haben, wählen Sie bitte den gewünschten *CmContainer* aus und klicken Sie dann auf die "**Lizenzaktualisierung**"-Schaltfläche.

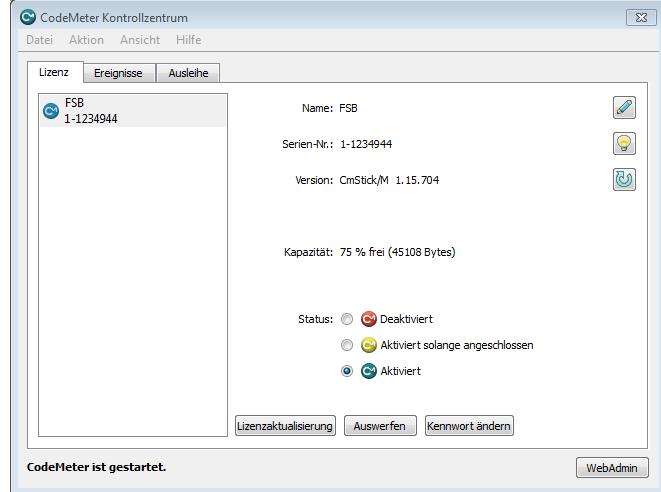


Abbildung 212: Lizenzaktualisierung - CodeMeter Kontrollzentrum

Bitte beachten Sie, dass das Einspielen von aktualisierten Lizenzdateien im laufenden Betrieb eines *CmContainers* derzeit nicht unterstützt wird.

 Schließen Sie daher vor einer Aktualisierung alle anderen *CodeMeter*®-geschützten Anwendungen, die Lizenzen aus dem *CmContainer* beziehen, der aktualisiert werden soll und speichern gegebenenfalls Ihre Daten ab.

Daraufhin öffnet sich der *CodeMeter Field Activation* (*CmFAS*)-Assistent mit dem Eröffnungsdialog. Klicken Sie auf die "**Weiter**"-Schaltfläche.



Abbildung 213: CmFAS Assistent

12.4.1.1 Erzeugen der Lizenzanforderungsdatei

Im Ausgangsdialog werden sie aufgefordert anzugeben, wie Sie weiter verfahren wollen. Sie können hier wahlweise eine Lizenzanforderung erzeugen, eine Lizenzaktualisierung einspielen, die Sie vom Software-Hersteller erhalten haben, oder optional nach der erfolgreichen Aktualisierung eine Quittung erzeugen, die sie dem Software-Hersteller zu senden. Klicken Sie auf die "Weiter"-Schaltfläche.



Abbildung 214: CmFAS - Lizenzanforderung erzeugen

12.4.1.1.1 Bestehende Lizenz erweitern

Bei der Erzeugung einer Lizenzanforderung können sie wählen, ob Sie eine bestehende Lizenz erweitern möchten, oder die Lizenz eines neuen Hersteller hinzufügen möchten. Klicken Sie auf die "Weiter"-Schaltfläche.



Abbildung 215: CmFAS – Erweitern einer bestehenden Lizenz

Erweitern Sie eine bestehende Lizenz, so wählen Sie den Software-Hersteller aus, für den Sie die Lizenzanforderung erstellen möchten. Klicken Sie auf die "**Weiter**"-Schaltfläche.



Abbildung 216: CmFAS – Lizenerweiterung – Hersteller auswählen

Der nächste Dialog erlaubt Ihnen das Abspeichern der Lizenzanforderungsdatei an einen von Ihnen ausgewählten Ort. Klicken Sie auf die "**Anwenden**"-Schaltfläche, um die Datei zu erzeugen. Diese Datei können Sie dann, zum Beispiel per e-Mail, an den Software-Hersteller schicken.



Abbildung 217: CmFAS – Lizenerweiterung – Datei speichern

Ein Dialog erscheint, der die erfolgreiche Erzeugung der Lizenzanforderungsdatei bestätigt. Über die "**Abschließen**"-Schaltfläche schließen Sie den Dialog. Senden Sie nun die Datei dem Software-Hersteller per e-Mail zu.

12.4.1.1.2 Lizenz eines neuen Herstellers hinzufügen

Bei der Erzeugung einer Lizenzanforderung können Sie wählen, ob Sie eine bestehende Lizenz erweitern möchten, oder die Lizenz eines neuen Herstellers hinzufügen möchten. Wählen Sie "**Lizenz eines neuen Herstellers hinzufügen**" aus. Klicken Sie auf die "**Weiter**" Schaltfläche.

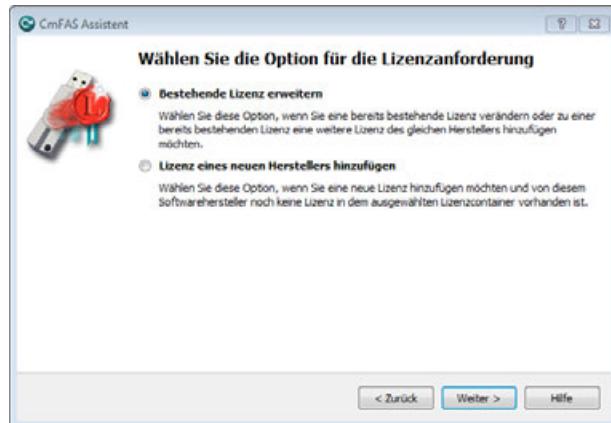


Abbildung 218: CmFAS – Neue Lizenz

Geben Sie im Dialog den Firm Code an, der Ihnen vom Software-Hersteller genannt wurde. Klicken Sie auf die "**Weiter**-Schaltfläche.

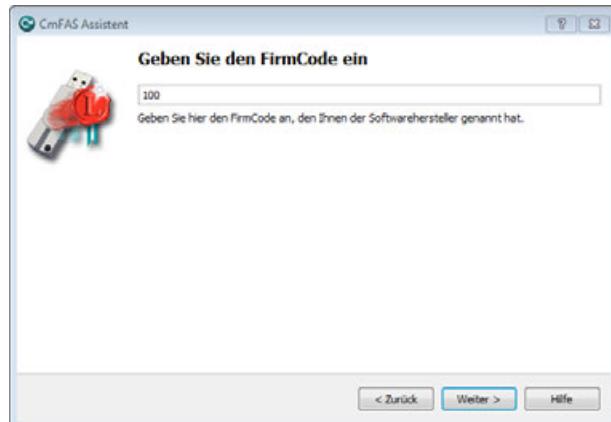


Abbildung 219: CmFAS – Lizenerweiterung – Firm Code

Der nächste Dialog erlaubt Ihnen das Abspeichern der Lizenzanforderungsdatei an einen von Ihnen ausgewählten Ort. Klicken Sie auf die "**Anwenden**-Schaltfläche, um die Datei zu erzeugen. Diese Datei können Sie dann, zum Beispiel per e-Mail, an den Software-Hersteller schicken.



Abbildung 220: *CmFAS* – Lizenerweiterung – Datei speichern

Sowohl bei der Erweiterung, als auch beim Hinzufügen erhalten Sie eine Bestätigung, dass die Lizenzanforderungsdatei erfolgreich erstellt wurde. Klicken Sie auf die "**Abschließen**"-Schaltfläche.

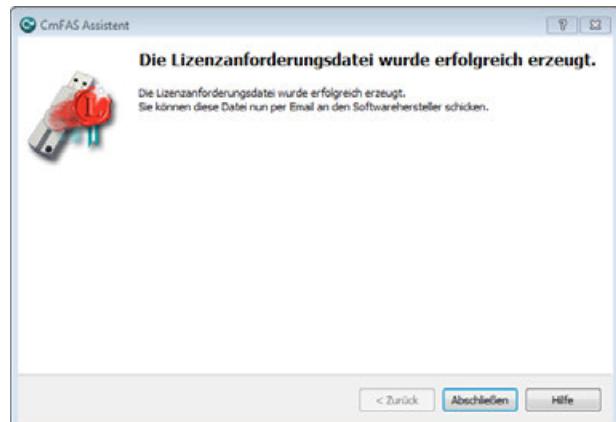


Abbildung 221: *CmFAS* – Lizenerweiterung – Bestätigung

12.4.1.2 Lizenzaktualisierung einspielen

Bitte beachten Sie, dass das Einspielen von aktualisierten Lizenzdateien im laufenden Betrieb eines *CmContainers* derzeit nicht unterstützt wird.

 Schließen Sie daher vor einer Aktualisierung alle anderen *CodeMeter®*-geschützten Anwendungen, die Lizenzen aus dem *CmContainer* beziehen, der aktualisiert werden soll und speichern gegebenenfalls Ihre Daten ab.

Um eine Lizenzaktualisierung einzuspielen, wählen Sie im Ausgangsdialog die betreffende Option. Klicken Sie auf die "**Weiter**"-Schaltfläche.

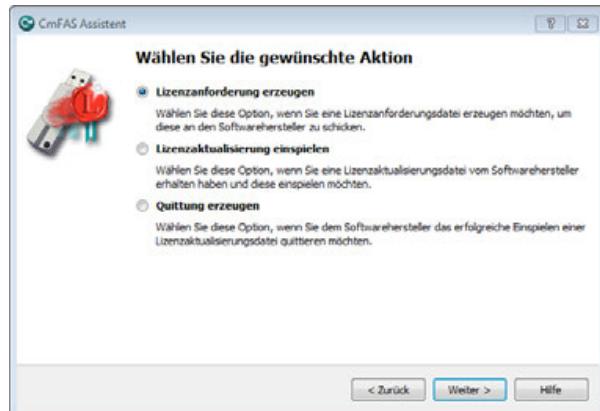


Abbildung 222: *CmFAS* – Lizenzaktualisierung

Im nächsten Dialog wählen Sie den Dateinamen, unter dem Sie die Lizenzaktualisierungsdatei, die Ihnen zugeschickt wurde, gespeichert haben. Klicken Sie die "**Anwenden**"-Schaltfläche, um die Lizenzaktualisierungsdatei einzuspielen.



Abbildung 223: *CmFAS* – Lizenzaktualisierung – Datei speichern

Der nachfolgende Dialog bestätigt das erfolgreiche Einspielen. Optional können Sie hier auch eine Quittungsdatei für den Software-Hersteller erzeugen. Diese Option haben Sie auch im Ausgangsmenü. Klicken Sie auf die "**Abschließen**"-Schaltfläche.

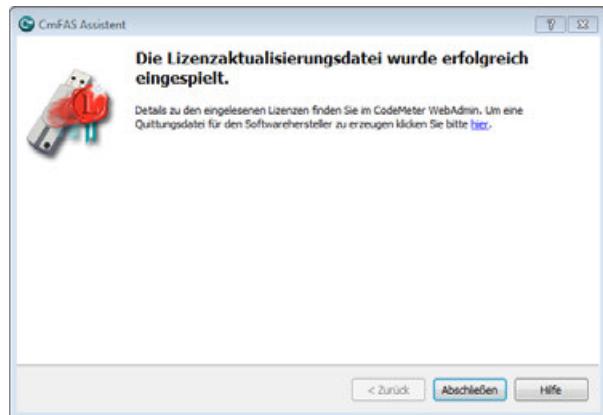


Abbildung 224: CmFAS – Lizenzaktualisierung – Bestätigung

12.4.1.3 Quittung erzeugen

Wählen Sie im Ausgangsmenü die betreffende "**Quittung erzeugen**"-Option. Klicken Sie auf die "**Weiter**"-Schaltfläche.

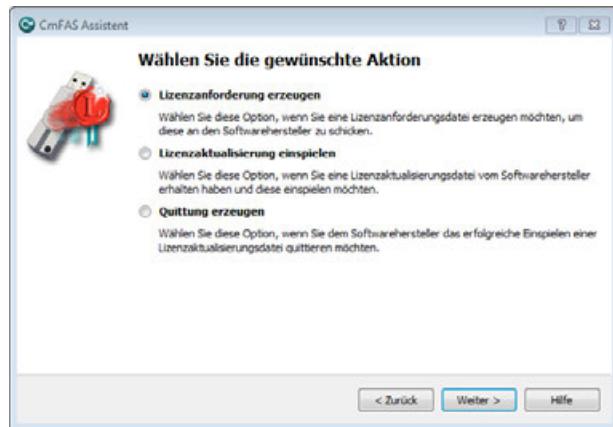


Abbildung 225: CmFAS – Quittung erzeugen

Wählen Sie im folgenden Dialog den Software-Hersteller, an den Sie die Quittungsdatei schicken möchten. Klicken Sie auf die "**Weiter**"-Schaltfläche.



Abbildung 226: CmFAS – Quittung erzeugen – Hersteller

Speichern Sie die Quittung über die "**Anwenden**"-Schaltfläche unter dem gewählten Dateinamen ab. Diese können sie nun den Software-Hersteller zukommen lassen.



Abbildung 227: CmFAS – Quittung erzeugen – Datei speichern

Die erfolgreiche Erstellung der Quittungsdatei wird Ihnen bestätigt. Klicken Sie auf die "**Abschließen**"-Schaltfläche, um den Vorgang zu beenden.



Abbildung 228: CmFAS – Quittung erzeugen – Bestätigung

12.5 CodeMeter WebAdmin

Mit *CodeMeter WebAdmin* erhalten Sie Information über verbundene *CmContainer* und die darin enthaltenen Lizenzentitäten. Darüber hinaus ist eine Konfiguration des Dienstes *CodeMeter License Server* möglich. Im Einzelnen bietet *CodeMeter WebAdmin* vielfältige Konfigurations- und Analysemöglichkeiten in den folgenden Bereichen:

- **Statusinformationen**⁴⁶⁷: Rechner, *CmContainer*
- **Konfiguration**⁴⁷¹: Verwendung als Netzwerk Server, Proxy-Einstellungen, Zugriffsschutz, Fernzugriff, Zeitserver, Datensicherung
- **Anzeige**: Anzeige aller vorhandener Lizzenzen lokal⁴⁸² und im Netzwerk⁴⁸⁷, Einsicht der Lizenzkonditionen, Session-Informationen
- **Verwaltung**⁴⁹⁰: Verwaltung von Netzwerklicenzen über manuelle Freigabe von Lizzenzen
- **Diagnose**⁴⁹⁶: Protokollierung
- **Datensicherung**⁴⁹⁷.

Die folgende Auflistung beschreibt kurz Begriffe, die auf einzelnen Seiten in *CodeMeter WebAdmin* immer wiederkehren.

Begriff	Beschreibung
Aktivierungsdatum	Informiert über den Aktivierungszeitpunkt einer Lizenz, d.h. ab wann die Lizenz einsetzbar.
Ausleihlizenzen	Informiert über vorhandene Ausleihlizenzen, den maximalen Ausleihzeitraum und einen eindeutigen Security Identifier (SID) für die Ausleihe im Netzwerk.
Extended Protected Data	Zusätzliches Eintragsfeld für den Lizenzgeber.
Feature Map	Informiert über Lizzenzen, die von Herstellern mit unterschiedlichen Funktionalitäten oder Modulen ausgeliefert werden. Diese sind über Feature Maps abgebildet, die einen bestimmtem Funktionsumfang beschreiben. Der hier angegebene Wert informiert über die gültige Funktionalität oder das freigeschaltete Modul der Lizenz.

Begriff	Beschreibung
Firm Code	Zahl, die den Lizenzcontainer eines Lizenzgeber identifiziert.
Hidden Data	Zusätzliches Eintragsfeld für den Lizenzgeber.
Implicit Firm Item (IFI)	Der Lizenzcontainer, der Lizizen enthält, die der Benutzer ausschließlich mit seinem <i>CmDongle</i> Passwort nutzen kann. Dieser Lizenzcontainer wird über die Zahl "0" identifiziert.
Lizenzzahl	Informiert über die Gesamtzahl der Lizizen, die für eine Anwendung zur Verfügung stehen.
Linger Time	Informiert über die Nachlaufzeit nach Freigabe oder Lizenz oder Beendigung der geschützten Anwendung.
Momentan ausgeliehene Lizizen	Zahl der momentan ausgeliehenen Lizizen.
n/a	Informiert darüber, dass es für diese Lizenz keinen entsprechenden Eintrag gibt (nicht eingetragen).
Nutzungseinheiten	Informiert über Lizizen, die nutzungsabhängig abgerechnet werden (pay-per-use, pay-per-print, etc.). Diese sind über Zähler realisiert, die bei der Nutzung eines Produkts heruntergezählt werden. Der hier angegebene Wert informiert über die verbleibenden Nutzungseinheiten einer Lizenz.
Nutzungszeitraum	Informiert über den Nutzungszeitraum einer Lizenz. Der hier angegebene Wert informiert über den Zeitraum der Nutzung einer Lizenz in Tagen. Er kann auch an einen Startzeitpunkt für die Gültigkeit einer Lizenz gebunden sein.
Product Code	Zahl, die den Lizenzeintrag eines Lizenzgebers identifiziert.
Protected Data	Zusätzliches Eintragsfeld für den Lizenzgeber.
Secret Data	Zusätzliches Eintragsfeld für den Lizenzgeber.
Status	Informiert über das Verhältnis, wie sich gestartete Instanzen einer geschützten Anwendung zur Belegung von Lizizen zueinander verhalten. User Limit: Hier belegt jede gestartete Instanz eine Lizenz. Shared: Hier belegen mehrere gestartete Instanzen auf demselben PC lediglich eine Lizenz. Exklusiv: Hier kann die geschützte Anwendung nur <u>einmal</u> auf einem PC gestartet werden. No User Limit: Hier können beliebig viele Instanzen der geschützten Anwendung im Netzwerk gestartet werden wobei keine zusätzlichen Lizizen belegt werden.
User Data	Zusätzliches Eintragsfeld für den Lizenzgeber.
Verfallsdatum	Informiert über das Verfallsdatum einer Lizenz, d.h. ab wann die Lizenz nicht mehr einsetzbar ist.
Wartungszeitraum	Informiert über den Zeitraum innerhalb dessen eine geschützte Version der Software erstellt sein muss, damit eine gültige Lizenzierung vorliegt. Es wird der Beginn und das Ende des Zeitraumes angezeigt.
Zugriffsmodus	Siehe: Status

Table 10: *CodeMeter WebAdmin* - Begriffe in der Lizenzanzeige

Sollte *CodeMeter WebAdmin* nicht starten, so gehen Sie wie folgt vor:

1. Prüfen Sie, ob sich der benutzte Web-Browser nicht im "Offline Modus" befindet.
2. Prüfen Sie die JavaScript-Unterstützung Ihres Web-Browser (JavaScript muss aktiviert sein!).

3. Tippen Sie die URLs: <http://localhost:22350> oder <http://127.0.0.1:22350> direkt in Ihren Web-Browser ein.

12.5.1 Voraussetzungen

TCP/IP basiert

Die Kommunikation zwischen *CodeMeter WebAdmin* und verbundenen *CmContainern* ist browserbasiert und fußt auf der Nutzung von Netzwerkkomponenten. Daher muss das Netzwerk-Protokoll TCP/IP installiert und der Zugriff auf den `localhost (127.0.0.1)` gestattet sein.



Eine tatsächliche Verbindung zum Internet wird jedoch nicht hergestellt.

Firewall-Einstellung

Achten Sie auch darauf, dass Einstellungen Ihrer Firewall die Kommunikation nicht blockieren.



CodeMeter Lizenzserver benutzt einen bestimmten IP-Port (voreingestellt: 22350), um mit Ihrem PC und Ihrem Netzwerk zu kommunizieren. Dieser Netzwerk Port ist bei der IANA (Internet Assigned Numbers Authority) registriert und eindeutig für die *CodeMeter®*-Kommunikation vergeben.

Dieser Port darf also nicht von Ihrer Firewall geblockt werden. Sorgen Sie dafür, dass der benutzte IP-Port 22350 für *CodeMeter®* frei verwendbar ist, d.h. geben Sie die Kommunikation für diesen IP-Port in Ihrer Firewall frei.

Kommunikationsmodus

Über Registry- bzw. Server-Einträge können sie zusätzlich festlegen, welchen Kommunikationsmodus *CodeMeter License Server* verwendet.

Die folgende Tabelle zeigt Ihnen wo sie für welches Betriebssystem im Profiling den Kommunikationsmodus setzen können.

Betriebssystem	Eintrag
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

Den Kommunikationsmodus setzen Sie im Parameter **ApiCommunicationMode**.

Die folgenden Einträge sind möglich:

CodeMeter®-Version	Einträge
kleiner als 4.40	'1' TCP/IP (Standard) '2' Shared Memory
ab 4.40	'1' Plattform-spezifisch (Standard) Plattform-spezifische Standards: • Windows: IPv6, IPv4; Shared Memory • Linux/Mac:IPv6, IPv4

CodeMeter®-Version	Einträge
	<ul style="list-style-type: none"> • WinCE: IPv4, Shared Memory '2' Shared Memory '4' IPv4 '8' IPv6 <p>Die einzelnen Modi können kombiniert werden.</p> <p> Wibu-Systems empfiehlt die jeweiligen Standard-Einstellungen beizubehalten, falls keine begründete Ausnahmen bestehen.</p>

12.5.2 Starten von CodeMeter WebAdmin

CodeMeter WebAdmin ist ein webbasiertes Tool, das mit jedem Standard Internet Browser angezeigt werden kann. Die folgende Tabelle zeigt die Startoptionen.

Betriebssystem	Start
 Windows	<ul style="list-style-type: none"> • über das <i>CodeMeter®</i>-Symbol in der Task-Leiste (rechte Maustaste) WebAdmin • über die Option WebAdmin im <i>CodeMeter Kontrollzentrum</i> • direkt in Ihrem Internet Browser, wenn Sie die URLs: http://localhost:22350 oder http://127.0.0.1:22350 eingeben.
 Mac OS / Linux	<ul style="list-style-type: none"> • über das <i>CodeMeter®</i>-Symbol in der Task-Leiste (rechte Maustaste) WebAdmin • über die Option WebAdmin im <i>CodeMeter Kontrollzentrum</i> • direkt in Ihrem Internet Browser, wenn Sie die URLs: http://localhost:22350 oder http://127.0.0.1:22350 eingeben.

Sollte CodeMeter WebAdmin nicht starten, versuchen Sie folgendes:

1. Prüfen Sie, ob sich der benutzte Internet Browser nicht im "Offline Modus" befindet.
2. Prüfen Sie die JavaScript-Unterstützung Ihres Internet Browsers.



JavaScript muss zur effektiven Verwendung von CodeMeter WebAdmin aktiviert sein.

3. Geben Sie die URLs: <http://localhost:22350> oder <http://127.0.0.1:22350> direkt in die Adresszeile Ihres Internet Browser ein.

12.5.3 Statusinfomation

Hier erhalten Sie erste Auskünfte über angeschlossene *CmContainer*:

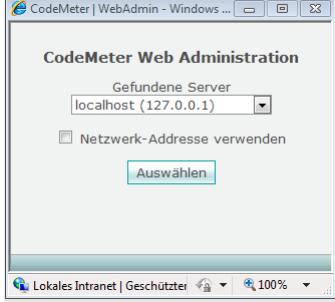
- [Generelle Informationen](#) ▾⁴⁶⁸
- [Informationen über CmContainer](#) ▾⁴⁶⁹

12.5.3.1 Generelle Informationen

Die "Home"-Seite gibt Auskunft über erste generelle Statusinformationen über Ihren Rechner, CodeMeter Lizenzserver sowie CodeMeter WebAdmin.



Abbildung 229: CodeMeter WebAdmin – "Home"

Element	Beschreibung
Rechnername	Die "Rechnername"-Schaltfläche zeigt den Namen des aktuellen Rechners an, auf dem der Dienst CodeMeter Lizenzserver gestartet ist. Es wird eine Suchanfrage über den Port 22350 über das Netzwerk gesendet. Zum Ändern des Rechners gehen Sie wie folgt vor: 1. Klicken Sie auf die "DNS-Namen"-Schaltfläche des Rechners. Der "CodeMeter Web Administration"-Dialog öffnet sich.
	 <p>Abbildung 230: CodeMeter WebAdmin – "Home Rechnername"</p> <p>2. Wählen Sie über das "Gefundene Server"-Auswahlfeld einen anderen Rechner aus, auf dem</p>

Element	Beschreibung
	ebenfalls CodeMeter® gestartet ist und der Dienst CodeMeter Lizenzserver läuft. 3. Aktivieren Sie die Netzwerk-Adresse verwenden -Option um die Netzwerk-Adresse des gefundenen Rechners zu verwenden. 4. Drücken Sie die " Auswählen "-Schaltfläche, um den ausgewählten Rechner zu verwenden.
Netzwerk Adresse	Zeigt die verwendete Netzwerkadresse an.
Betriebssystem	Zeigt Informationen über das verwendete Betriebssystem.
Server Startzeit	Gibt Information über die Server Startzeit aus.
Runtime Version	Zeigt Informationen über die verwendete CodeMeter®-Laufzeitumgebung.
Server Version	Zeigt Informationen über die verwendete CodeMeter®-Version auf dem Server
WebAdmin Version	Zeigt Informationen über die verwendete CodeMeter WebAdmin-Version.

12.5.3.2 Informationen über den CmContainer

Die "Inhalt | CmContainer"-Seite zeigt Ihnen Informationen über ausgewählte CmContainer an.

CmContainer:	1-1440495
Name:	FS
CmContainer Typ:	CmStick/M 8GB 1.18.900
Erstes Laufwerk:	E: (7840 MB)
Status:	<input checked="" type="radio"/> Gesperrt <input type="radio"/> Aktiviert solange angeschlossen <input type="radio"/> Aktiviert
System Zeit (PC):	2011-11-17 15:20:10
System Zeit (CmContainer):	2011-11-17 15:20:09
Zertifizierte Uhrzeit (CmContainer):	2011-05-19 09:13:18
Freier Speicher:	78 % (46.868 Bytes)
	<input type="button" value="Aktualisieren"/>
	<input type="button" value="Defragmentieren"/>

Abbildung 231: CodeMeter WebAdmin – "Inhalt | CmContainer"

Element	Beschreibung
CmContainer	Wählen Sie den CmContainer aus, auf den sich die Informationen beziehen. Falls

Element	Beschreibung
	mehrere <i>CmContainer</i> angeschlossen sind, wählen Sie über die Liste mit Hilfe der Seriennummer den gewünschten <i>CmContainer</i> aus.
Name	Zeigt den Namen des ausgewählten <i>CmContainers</i> an. Falls Sie den Namen Ihres <i>CmContainers</i> ändern möchten, können Sie dies über <i>CodeMeter Kontrollzentrum</i> tun.
CmContainer Typ	Zeigt den Typ des ausgewählten <i>CmContainers</i> an.
Erstes Laufwerk	Zeigt die Laufwerksinformationen des ausgewählten <i>CmDongles</i> .  Die Laufwerksgröße wird nur bei <i>CmDongles</i> mit Flash-Speicher angezeigt.
Status	Zeigt den aktuellen Aktivierungsstatus des ausgewählten <i>CmContainers</i> an. Die folgenden Statuszustände werden angezeigt: <ul style="list-style-type: none"> Deaktiviert: Der verbundene <i>CmContainer</i> ist deaktiviert und kann von keiner Anwendung verwendet werden. Aktiviert solange angeschlossen: Der <i>CmDongle</i> ist aktiviert, solange er angeschlossen ist und Strom zugeführt wird. Nach Entfernen vom PC wird ein <i>CmDongle</i> automatisch deaktiviert. Aktiviert: Der <i>CmContainer</i> ist voll aktiviert. Im Fall eines <i>CmDongles</i> ist der Zugriff aus Lizzenzen immer noch möglich, selbst wenn der <i>CmDongle</i> abgezogen wird. Der Aktivierungsstatus eines <i>CmContainers</i> kann über CodeMeter Kontrollzentrum geändert werden.  Wibu-Systems empfiehlt den " Aktiviert solange angeschlossen "-Aktivierungsstatus für <i>CmDongles</i> zu verwenden. Nur das stellt sicher, dass bei Verlust des <i>CmDongles</i> Unbefugte <u>nicht</u> auf Lizzenzen oder persönliche Daten im <i>CmDongle</i> zugreifen können.
System Zeit (PC)	Zeigt die System Zeit (lokale Zeit auf dem Computer) zum Startzeitpunkt des Dienstes <i>CodeMeter Lizenzserver</i> an.
System Zeit (CmContainer)	Zeigt die gespeicherte System Zeit (interne Zeit) des <i>CmContainers</i> an. Diese beiden Zeiten können voneinander abweichen im Falle, dass die System Zeiten des PC und des <i>CmContainers</i> noch nicht synchronisiert haben
Zertifizierte Uhrzeit (CmContainer)	Zeigt die im <i>CmContainer</i> gespeicherte, zertifizierte Uhrzeit an. Um die zertifizierte Uhrzeit Ihres <i>CmContainers</i> über einen <i>CodeMeter® Time Server</i> zu aktualisieren, klicken Sie auf die " Aktualisieren "-Schaltfläche. Diese Aktion wird durch einen Dialog bestätigt.
	 <p>Abbildung 232: <i>CodeMeter WebAdmin</i> – Zertifizierte Zeit aktualisieren</p>
Freier Speicher	Zeigt den Freien Speicher des SmartCard Chips des <i>CmDongles</i> an, d.h. wieviel Platz für die zusätzliche Programmierung von Lizenzinträgen noch verfügbar

Element	Beschreibung
	ist.
Defragmentieren	Durch Klicken der "Defragmentieren"-Schaltfläche wird der Speicher des <i>CmDongles</i> Chips defragmentiert.

12.5.4 Konfiguration

Hier nehmen Sie Einstellungen zur Verwendung als Netzwerk- (LAN- und/oder WAN-) Server, zu Proxy-Einstellungen, zum Zugriffsschutz, Fernzugriff, Zeitserver und zur Datensicherung vor.

12.5.4.1 Netzwerk

Zum Einrichten von *CodeMeter®* in einer Netzwerkumgebung führen Sie auf der "**Einstellungen | Netzwerk**"-Seite die folgenden Schritte durch.

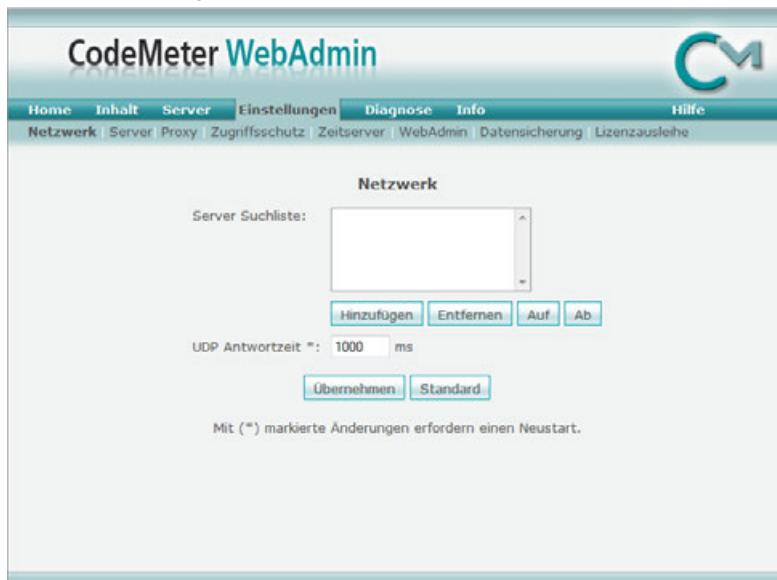


Abbildung 233: CodeMeter WebAdmin – "Einstellungen | Netzwerk"

Element	Beschreibung
Server Suchliste	Verwenden einer Serversuchliste , die Zugriffe auf und die Reihenfolge von eingerichteten <i>CodeMeter®</i> -Netzwerk- und WAN (Wide Area Network)-Servern bearbeitet. Die Serversuchliste bearbeiten Sie, indem Sie über die entsprechenden Schaltflächen Server "Hinzufügen", "Entfernen", aber auch in der Reihenfolge ändern ("Auf" und "Ab"). Mit der "Übernehmen"-Schaltfläche speichern Sie diese Einstellungen ab.

Element	Beschreibung										
	<p> Mit der "Standard"-Schaltfläche setzen Sie die Einstellungen der Server-Suchliste zurück.</p> <p>Alternativ, können Sie die Server Suchliste auch über die <code>CodeMeter.ini</code> bzw <code>Server.ini</code> setzen. Die untenstehende Tabelle zeigt Ihnen, wo sie die Dateien finden.</p> <table border="1"> <thead> <tr> <th>Betriebssystem</th><th>Konfigurationsdatei</th></tr> </thead> <tbody> <tr> <td>Windows</td><td><code>%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini</code></td></tr> <tr> <td>Mac OS</td><td><code>\Library\Preferences\com.wibu.CodeMeter.Server.ini</code></td></tr> <tr> <td>Linux</td><td><code>\etc\wibu\CodeMeter\Server.ini</code></td></tr> <tr> <td>Solaris</td><td><code>\etc\opt\CodeMeter\Server.ini</code></td></tr> </tbody> </table> <p>Setzen Sie im separaten Bereich [ServerSearchList] die Server wie im unteren Beispiel gezeigt.</p> <pre>[ServerSearchList] [ServerSearchList\Server1] Address=184.45.89.5 [ServerSearchList\Server2] Address=185.55.78.6</pre> <p>Das Setzen der Netzwerkeinstellungen macht für manche Änderungen einen Neustart des CodeMeter® Dienstes erforderlich. Dazu müssen Sie den CmContainer aber nicht auswerfen oder deaktivieren. Nachdem Sie die Änderungen durchgeführt haben, können Sie in CodeMeter Kontrollzentrum⁴⁴⁸ den CodeMeter® Dienst stoppen und danach wieder starten. Für Nicht-Windows-Betriebssysteme siehe hier⁴⁴³.</p> <p>Überprüfen Sie, ob eine Verbindung zustande gekommen ist, indem Sie auf der "Home"-Seite die "Rechnername"-Schaltfläche klicken und in der Liste der gefundenen Server überprüfen, ob der Rechner aufgenommen wurde. Sie können die Überprüfung ebenfalls vornehmen, indem Sie CodeMeter Kontrollzentrum auf den Clients und dem Server öffnen und im "Ereignisse" Karteireiter den Kommunikationsstatus einsehen.</p> <p> Falls keine Verbindung zustande gekommen, geben Sie auf den Client-Rechnern die Server IP-Adresse ein.</p> <p>Verwendung im Netzwerk (LAN):</p> <p>Über die Eingabe der Rechnernamen oder IP-Adressen bestimmen Sie, dass sich die Anfragen des Clients genau an den definierten CodeMeter®-Netzwerkserver richten. Dies erhöht die Performance im Netz.</p> <p> Falls sich der CodeMeter®-Netzwerkserver in einem anderen Teilnetz (Subnetz) befindet, sollten Sie immer die IP Adresse in die Serversuchliste eintragen, um UDP Broadcast-Problemen vorzubeugen.</p> <p>Standardmäßig bindet sich der CodeMeter Lizenzserver auf den ersten gefundenen Netzwerkadapter.</p>	Betriebssystem	Konfigurationsdatei	Windows	<code>%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini</code>	Mac OS	<code>\Library\Preferences\com.wibu.CodeMeter.Server.ini</code>	Linux	<code>\etc\wibu\CodeMeter\Server.ini</code>	Solaris	<code>\etc\opt\CodeMeter\Server.ini</code>
Betriebssystem	Konfigurationsdatei										
Windows	<code>%Program Files%\CodeMeter\Runtime\bin\CodeMeter.ini</code>										
Mac OS	<code>\Library\Preferences\com.wibu.CodeMeter.Server.ini</code>										
Linux	<code>\etc\wibu\CodeMeter\Server.ini</code>										
Solaris	<code>\etc\opt\CodeMeter\Server.ini</code>										

Element	Beschreibung
	<p>Verwendung im Weitverkehrsnetz (Wide Area Network, WAN): Über die Eingabe der IP-Adressen bestimmen Sie, dass sich die Anfragen des Clients genau an den definierten <i>CodeMeter License Server</i> in einem Wide Area Network richten.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Bitte beachten Sie, dass Sie bei der Angabe der IP-Adresse immer ein "https://\" voranstellen. Diese wird für die abgesicherte Kommunikation zu einem Reverse Proxy im WAN benötigt. </div>
UDP Antwortzeit	<p>Angabe einer UDP Antwortzeit, um den Zeitraum zu definieren, innerhalb der eine UDP-Anfrage nach im Netzwerk vorhandenen <i>CodeMeter Lizenzservern</i> beantwortet werden muss. Der Standardwert beträgt 1000 Millisekunden.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Über die Änderung dieser Zeit lässt sich die Performance des Dienstes anpassen. Wenn kein dringender Bedarf besteht, sollte diese Einstellung allerdings beibehalten werden. </div>

12.5.4.2 Server

Zum Einrichten von *CodeMeter*® in einer Netzwerkumgebung führen Sie auf der "**Einstellungen | Server**"-Seite die folgenden Schritte durch.



The screenshot shows the 'Server' configuration section of the CodeMeter WebAdmin interface. It includes fields for 'Bind Address *:' (set to 'All (Default)'), 'Starte Netzwerk Server:' (unchecked), 'Netzwerk Port *:' (set to 22350), 'Starte CmWAN Server:' (unchecked), and 'CmWAN Port *:' (set to 22351). At the bottom are 'Übernehmen' and 'Standard' buttons, with a note: 'Mit (*) markierte Änderungen erfordern einen Neustart.'

Abbildung 234: *CodeMeter WebAdmin – "Einstellungen | Server"*

Element	Beschreibung
Bind Adresse	Auswahl einer Netzwerk Adresse , auf die sich der Dienst <i>CodeMeter Lizenzserver</i>

Element	Beschreibung
	<p>binden soll.</p> <p> Dies ist vor allem dann notwendig, wenn der PC mehrere Netzwerkarten (virtuelle Adapter) besitzt und als Netzwerk-Server für die Lizenzen zur Verfügung stehen soll.</p>
Starte Netzwerk Server	Aktivieren der Starte Netzwerk Server -Option, um den Rechner als CodeMeter®-Netzwerkserver zu nutzen. Damit stellt dieser Rechner über den Dienst <i>CodeMeter Lizenzserver</i> seine CodeMeter®-Lizenzen im Netzwerk zur Verfügung.
Netzwerk Port	<p>Angabe eines Netzwerk Port. Der Port 22350 ist die Standardeinstellung für die CodeMeter®-Kommunikation. Dieser Netzwerk Port ist bei der IANA (Internet Assigned Numbers Authority) registriert und eindeutig für die CodeMeter®-Kommunikation vergeben.</p> <p> Sie können diesen Wert an Ihre Bedürfnissen anpassen. In diesem Fall sollten Sie allerdings sicherstellen, dass alle <i>CodeMeter Lizenzserver</i> diesen Port benutzen, falls die CodeMeter®-geschützte Anwendung über das Netzwerk benutzt werden soll.</p>
Starte CmWAN Server	Aktivieren der Starte CmWAN Server -Option, um den Rechner in einem Wide Area Network (WAN) zu nutzen und Lizenzzugriffe zu ermöglichen.
CmWAN Port	<p>Angabe eines CmWAN Port. Der Port 22351 ist die Standardeinstellung für die CodeMeter®-Kommunikation über WAN.</p> <p> Sie können diesen Wert an Ihre Bedürfnissen anpassen. In diesem Fall sollten Sie allerdings sicherstellen, dass:</p> <ul style="list-style-type: none"> • alle <i>CodeMeter Lizenzserver</i> diesen Port benutzen, wenn die CodeMeter®-geschützte Anwendung über das Wide Area Network (WAN) Netzwerk auf Lizenzen zugreift. • der notwendige Reverse Proxy dieselbe Port-Einstellung besitzt.

Mit der "**Übernehmen**"-Schaltfläche speichern Sie diese Einstellungen ab. Mit der "**Standard**"-Schaltfläche setzen Sie die Einstellungen der Serversuchliste zurück.

Das Setzen der Server-Einstellungen macht für manche Änderungen einen Neustart des *CodeMeter*®-Dienstes erforderlich. Dazu müssen Sie den *CmContainer* aber nicht auswerfen oder deaktivieren. Nachdem Sie die Änderungen durchgeführt haben, können Sie in [CodeMeter Kontrollzentrum](#) ▶⁴⁴⁸ den *CodeMeter*®-Dienst stoppen und danach wieder starten. Für Nicht-Windows-Betriebssysteme siehe [hier](#) ▶⁴⁴³.

12.5.4.3 Proxy Einstellungen

Auf der "Einstellungen | Proxy"-Seite nehmen Sie Einstellungen vor, wenn Sie einen Proxy Server verwenden.



Abbildung 235: CodeMeter WebAdmin – "Einstellungen | Proxy"

Element	Beschreibung
Proxy	Aktivieren Sie diese Option für die Proxy Server-Unterstützung. Falls Sie einen Proxy Server verwenden, tragen Sie hier die IP-Adresse, oder den DNS-Namen des Proxy Servers sowie die Port -Nummer ein. i Ein Proxy wird dann benötigt, wenn Sie zertifizierte Zeitaktualisierungen über den Wibu-Systems Zeitserver vornehmen, oder Produkte über einen Internet Shop erworben werden.{
Authentifizierung	Aktivieren Sie diese Option für eine benötigte Proxy Server-Authentifizierung. Tragen Sie hier die Benutzer kennung sowie das Benutzer- Passwort für den Proxy Server ein.

Ist die Auswahl verschiedener CodeMeter®-Client PCs in CodeMeter WebAdmin nicht möglich, dann versuchen Sie folgendes:

1. Schließen Sie die betreffenden CodeMeter Lizenzserver von der Proxy-Verwendung aus.
2. Schreiben Sie hierfür die IP-Adressen oder die DNS-Namen dieser CodeMeter®-Client-PCs in die Proxy-Exceptions-Liste Ihres Internet Explorers: [Tools - Internet Options.. | Connections | Lan Settings | Advanced | Exceptions]

12.5.4.4 Zugriffsschutz

Auf der "Einstellungen | Zugriffsschutz"-Seite nehmen Sie Einstellungen vor, die den Zugriff durch Clients auf CodeMeter Lizenzserver regeln.

Abbildung 236: *CodeMeter WebAdmin - "Einstellungen | Zugriffsschutz"*

Element	Beschreibung
Clients	<p>Zeigt eine Liste aller Client-Rechner an, die die Berechtigung haben, CodeMeter Lizenzserver zu benutzen, d.h. eine Lizenz zu belegen.</p> <p> Ist diese Liste leer kann jeder CodeMeter®-Client im Netzwerk CodeMeter Lizenzserver benutzen. Dies entspricht der Standardeinstellung.</p> <p>Zum Hinzufügen eines Clients gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Klicken Sie auf die "Hinzufügen"-Schaltfläche. Ein Eingabeaufforderungsdialog erscheint. <ol style="list-style-type: none"> 2. Geben Sie im den Rechnernamen oder die IP-Adresse des Client-Rechners in das Benutzereingabefeld ein. 3. Klicken Sie auf die "OK"-Schaltfläche. Der Rechner wird nun zur Liste hinzugefügt. <p>Zum Entfernen eines Clients gehen Sie wie folgt vor:</p> <ol style="list-style-type: none"> 1. Klicken Sie auf die "Entfernen"-Schaltfläche. Der Rechner wird von der Liste entfernt.

Element	Beschreibung
Zugriff auf die FSB	<p>Falls Sie im Besitz einer CodeMeter Firm Security Box (FSB) sind, aktiviert diese Option die Freigabe der FSB für die Netzwerknutzung. Dann kann die FSB kann dann von mehreren Benutzer verwendet werden, um z.B. <i>CmContainer</i> zu programmieren, oder Anwendungen automatisch zu schützen.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Diese Option ist nur für CodeMeter® Lizenznehmer mit eigenem CodeMeter®-Firm Code sinnvoll! </div> <p>Klicken Sie auf die "Übernehmen"-Schaltfläche, um die vorgenommenen Änderungen zu speichern. Durch vorheriges Klicken der "Standard"-Schaltfläche stellen Sie die Standardeinstellung wieder her. Die Client-Liste ist dann leer und eine FSB ist nicht im Netz verfügbar.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Das Setzen der Zugriffseinstellungen macht für manche Änderungen einen Neustart des CodeMeter®-Dienstes erforderlich. Dazu müssen Sie den <i>CmContainer</i> aber nicht auswerfen oder deaktivieren.  Nachdem Sie die Änderungen durchgeführt haben, können Sie in CodeMeter Kontrollzentrum den CodeMeter®-Dienst stoppen und danach wieder starten. Für Nicht-Windows-Betriebssysteme siehe hier. </div>

Zusätzliche Zugriffsregelung der Client-Liste über Whitelist und Blacklist

Alternativ besteht auch die Möglichkeit, eine Positiv- wie Negativ-Zugriffsliste von Clients (Whitelist und Blacklist) zu erstellen. Für die einzelnen Betriebssysteme führen Sie dieses Profiling an den folgenden Orten durch:

Betriebs-System	Profilerstellung
 Windows	Registry-Eintrag in <code>HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion</code>
 Mac OS	<code>/Library/Preferences/com.wibu.CodeMeter.Server.ini</code>
 Linux	<code>/etc/wibu/CodeMeter/Server.ini.</code>
 Solaris	<code>/etc/opt/CodeMeter/Server.ini</code>

Die Profilerstellung für **CodeMeter Lizenzserver** umfasst die folgenden Versionen (`CodeMeter.exe`, `CodeMeterMacX`, `CodeMeterLin`, `CodeMeterSun`),

	<p>Wenn Sie die <code>*.ini</code> Dateien im Fall von Mac OS, Linux und Sun editieren, muss CodeMeter Lizenzserver vorher beendet werden. Andernfalls werden die vorgenommenen Änderungen nicht übernommen.</p>
---	---

Parameter	Beschreibung
<code>Client<index>=<Subnet>[,<serial>[,FC[,PC]]]</code> (Whitelist)	<p>Whitelist: Diese Parameter enthalten die IP-Adressen von Client-PCs im Netzwerk, die eine Berechtigung besitzen auf den lokalen CodeMeter Lizenzserver zuzugreifen. Steht eine IP Adresse eines Clients nicht in dieser Liste so wird der Zugriff verweigert. Existiert keine Whitelist greifen keine Einschränkungen. Auch die Angabe von Subnetzen ist möglich. Die Syntax lautet wie folgt:</p>

Parameter	Beschreibung
	<p><code>Client<index>=<Subnetz>[,<serial>[,FC[,PC]]]</code> Die optionale serial muss dem Format MaskenByte-SerialNumber folgen (z.B. 1-1179681). <u>Beispiel:</u> <code>Client1=192.168.0.0/24,1-123456,10,13</code> dies adressiert alle Rechner von 192.168.0.0-192.168.0.255 (Class C). Üblich sind noch /8 (Class A) und /16 (Class B). Die Seriennummer, FC und PC sind (wie bisher) optional.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Diese Whitelist entspricht der Client-Liste in <i>CodeMeter WebAdmin</i>. </div>
<code>Client<index>=<Subnetz>[,<serial>[,FC[,PC]]]</code> [SZ, optional]	<p>Blacklist: Diese Parameter enthalten die IP-Adressen von Client-PCs im Netzwerk, die keine Berechtigung besitzen auf den lokalen <i>CodeMeter Lizenzserver</i> zuzugreifen. Steht eine IP Adresse eines Clients in dieser Liste so wird der Zugriff verweigert. Existiert keine Blacklist greifen keine Einschränkungen. Auch die Angabe von Subnetzen ist möglich. Die Syntax lautet wie folgt: <code>Client<index>=<Subnetz>[,<serial>[,FC[,PC]]]</code> Die optionale serial muss dem Format MaskenByte-SerialNumber folgen (z.B. 1-1179681). <u>Beispiel:</u> <code>Client1=192.168.0.0/24,1-123456,10,13</code> dies adressiert alle Rechner von 192.168.0.0-192.168.0.255 (Class C). Üblich sind noch /8 (Class A) und /16 (Class B). Die Seriennummer, FC und PC sind (wie bisher) optional.</p>

12.5.4.5 Zeitserver

Auf der "Einstellungen | Zeitserver"-Seite nehmen Sie Einstellungen vor, die die CodeMeter®-Zeitserver betreffen.

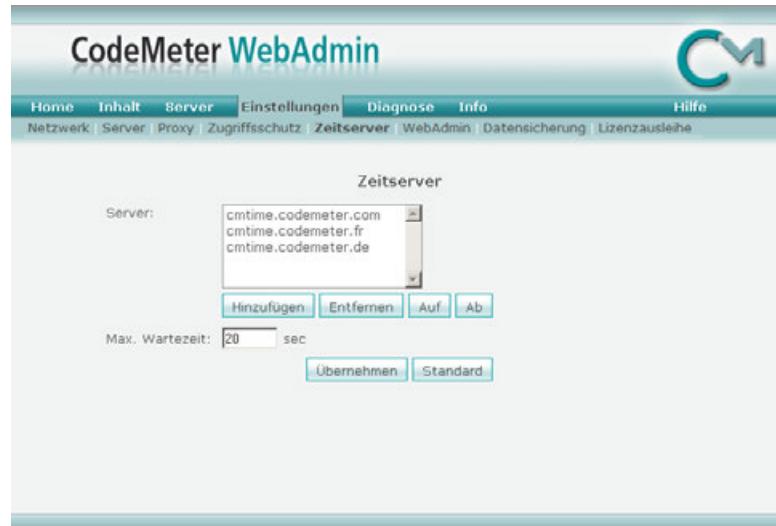


Abbildung 237: CodeMeter WebAdmin - "Einstellungen | Zeitserver"

Element	Beschreibung
Zeitserver	Zeigt eine Liste von Wibu-Systems CodeMeter® Zeitservern an, über die eine zertifizierte Zeit bezogen wird. Die Zeitserver sind entweder als Internet-Adresse oder als IP Adresse angegeben. Die Zeitserver-Liste bearbeiten Sie, indem Sie über die entsprechenden Schaltflächen Zeitserver "Hinzufügen" oder "Entfernen", aber auch in der Reihenfolge ändern ("Auf" und "Ab").
Max Wartezeit	Definiert den maximalen Antwortzeitraum der CodeMeter®-Zeitserver. Klicken Sie auf die "Übernehmen"-Schaltfläche, um die vorgenommenen Änderungen zu speichern. Durch vorheriges Klicken der "Standard"-Schaltfläche stellen Sie die Standardeinstellung wieder her.

12.5.4.6 WebAdmin

Auf der "Einstellungen | WebAdmin"-Seite nehmen Sie Einstellungen vor, die den lokalen bzw. den Fernzugriff auf CodeMeter WebAdmin regeln und die Spracheinstellungen betreffen.



Abbildung 238: CodeMeter WebAdmin - "Einstellungen | WebAdmin"

Element	Beschreibung
Nur lokaler Zugriff (unbeschränkt)	Aktivieren Sie diese Option, um den Zugriff auf CodeMeter WebAdmin unbeschränkt lokal zuzulassen.
Authentifizierung verwenden	Aktivieren Sie diese Option, um per Fernzugriff auf CodeMeter WebAdmin auch schreibend zugreifen zu können. Damit kann von einem Client aus auf den Server per HTTP zugegriffen werden. Dazu ist eine Authentifizierung erforderlich. Geben Sie dazu in die Felder Benutzername , Passwort und Passwort wiederholen die Authentifizierungsdaten ein.
Sprache	<p>Stellen Sie über dieses Feld die Sprache der CodeMeter WebAdmin-Oberfläche ein. Sie können zwischen deutsch, englisch, französisch, italienisch, japanisch und chinesisch wählen.</p> <p>Klicken Sie auf die "Übernehmen"-Schaltfläche, um die vorgenommenen Änderungen zu speichern. Durch vorheriges Klicken der "Standard"-Schaltfläche stellen Sie die Standardeinstellung wieder her. Eine Leseberechtigung ist gesetzt und deutsch als Standardsprache gesetzt.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p>i Das Setzen der Zugriffseinstellungen macht für manche Änderungen einen Neustart des CodeMeter®-Dienstes erforderlich. Dazu müssen Sie den CmContainer aber nicht auswerfen oder deaktivieren. Nachdem Sie die Änderungen durchgeführt haben, können Sie in CodeMeter Kontrollzentrum den CodeMeter®-Dienst stoppen und dann wieder starten. Für Nicht-Windows-Betriebssysteme siehe hier⁴⁴³.</p> </div>

12.5.4.7 Datensicherung

Auf der "Einstellungen | Datensicherung" Seite nehmen Sie Einstellungen vor, die Speicherort und -intervall von Datensicherungen des *CmDongles* bestimmen.



Abbildung 239: *CodeMeter WebAdmin* - "Einstellungen | Datensicherung"

Element	Beschreibung
Ordner	Geben Sie im Ordner Feld den Ort an, an dem das Ergebnis der Datensicherung des <i>CmDongles</i> gespeichert werden soll. Der Standardspeicherort ist hierbei vom Betriebssystem abhängig.
Intervall	Geben Sie im Intervall Feld den Zeitrahmen innerhalb dessen eine Datensicherung automatisch ausgeführt werden soll. Standardmäßig und automatisch wird alle 24 Stunden eine Datensicherung durchgeführt. Sie können aber auch jederzeit eine Sicherung des <i>CmDongles</i> durchführen.
Zertifizierte Uhrzeit	Aktivieren Sie diese Einstellung, wenn vor jeder Sicherung die zertifizierte Zeit über die Zeitserver aktualisiert werden soll. Klicken Sie auf die " Übernehmen "-Schaltfläche, um die vorgenommenen Änderungen zu speichern. Durch vorheriges Klicken der " Standard "-Schaltfläche stellen Sie die Standardeinstellung wieder her.

12.5.4.8 Lizenzausleihe

Auf der "Einstellungen | Lizenzausleihe"-Seite können Sie eintragsspezifische Einstellungen einer ausgeliehenen Lizenz überschreiben, um die Anzahl der ausgeliehenen Lizenzen sowie deren Ausleihdauer abweichend von der Programmierung abzuändern.



Abbildung 240: CodeMeter WebAdmin – "Einstellungen | Lizenzausleihe"

Zur Einstellung der Lizenzausleihsparameter gehen Sie wie folgt vor:

1. Aktivieren Sie die "**Eintragsspezifische Einstellungen überschreiben**"-Option, um die Abänderung der Lizenzbedingungen der Ausleihlizenz zuzulassen.
2. Geben Sie im "**Maximale Ausleihzeit**"-Feld den maximalen Zeitrahmen in Minuten an, die die Lizenz ausgeliehen werden darf.
3. Geben Sie im "**Maximale Ausleihe**"-Feld die maximale Anzahl der Ausleihlizenzen an, die ausgeliehen werden dürfen.
4. Wählen Sie im Feld "**Server Identifizierung**"-Feld aus wie die Identifizierung des Servers erfolgen soll: Entweder über den Server Namen oder die IP-Adresse.
5. Klicken Sie auf die Schaltfläche "**Übernehmen**" um die vorgenommenen Änderungen zu speichern. Durch vorheriges Klicken der "**Standard**"-Schaltfläche stellen Sie die Standardeinstellung wieder her.

12.5.5 Lizenzanzeige

CodeMeter WebAdmin zeigt Informationen über lokal verfügbare⁴⁸³ Lizenzen oder Netzwerklicensen⁴⁸⁷ an.

12.5.5.1 Lokale Lizenzen

Die "Inhalt | Lizenzen"-Seite zeigt Ihnen alle lokale Lizenzen an, die ein ausgewählter, oder alle verbundenen CmContainer enthalten. Wählen Sie über **CmContainer** den gewünschten, oder alle verbundenen CmContainer aus.

The screenshot shows the 'CodeMeter WebAdmin' interface with the 'Inhalt' tab selected. A dropdown menu shows 'CmContainer | Lizenzen' is selected. The main content area displays a table of licenses categorized by manufacturer:

10 Hersteller 1					
Product Code	Name	Nutzungs-einheiten	Verfalls-datum	Aktivierungs-datum	Lizenz-anzahl
10	Textanwendung	200	n/a	n/a	10
13	Tabellenanwendung	400	2010-03-02 17:48:22	n/a	20
14	Chartanwendung	200	n/a	n/a	23

228 Hersteller 2					
Product Code	Name	Nutzungs-einheiten	Verfalls-datum	Aktivierungs-datum	Lizenz-anzahl
67	Druckanwendung	1000	n/a	2009-03-02 17:48:45	50
1000	Fax Add-on	n/a	2009-04-13 11:50:05	[ausgeliehen]	1

100003 Bundling Articles					
Product Code	Name	Nutzungs-einheiten	Verfalls-datum	Aktivierungs-datum	Lizenz-anzahl
1	SecuriKey Lite	n/a	n/a	n/a	1

Abbildung 241: CodeMeter WebAdmin – "Inhalt | Lizenzen"

Die Anzeige der lokalen Lizenzen ist nach unterschiedlichen Lizenzgebern gegliedert. Ein Lizenzgeber ist durch einen Zahlenwert, den Firm Code, und einen Namen eindeutig gekennzeichnet. In der obigen Abbildung ist dies z.B. der Firm Code "10" von "Hersteller 1".



Alle zugehörigen Produkte, und damit die Lizenzen, sind unterhalb der einzelnen Lizenzgeber mit ihren jeweiligen Product Codes, einem eindeutigen Zahlenwert, aufgelistet.

In der obigen Abbildung ist dies zum einen das Produkt "Druckanwendung" mit einem Product Code 67. Zum anderen das Produkt "Fax Add-on" mit einem Product Code 1000, das bis zum Verfallsdatum als lokale Lizenz ausgeliehen ist. Darüber hinaus erhalten Sie weitere [Informationen über die Lizenz](#)⁴⁸⁴, wie vorhandene **Nutzungseinheiten**, **Verfallsdatum**, **Aktivierungsdatum** und **Lizenzanzahl**.

Klicken Sie auf den [Firm Code](#)⁴⁸⁴, um detailliertere Information über die Lizenzierung von Produkten eines bestimmten Anbieters angezeigt zu bekommen.

Klicken Sie auf den [Product Code](#)⁴⁸⁴, um detailliertere Information über die Lizenzierung der Produkte eines bestimmten Lizenzgebers angezeigt zu bekommen.

12.5.5.1.1 Lizenzgeber-Informationen

Diese Seite zeigt detailliertere Information über die Lizenzierung von Produkten eines bestimmten Anbieters.

In der folgenden Abbildung sehen Sie z.B. alle Lizenzen des Herstellers 1. Zusätzliche [Informationen](#)⁴⁶⁴ umfassen den **Product Code**, **CmContainer-Seriennummer**, **Namen**, **Nutzungseinheiten**, **Aktivierungsdatum**, **Verfallsdatum**, **Lizenzzahl** und **Feature Map**.

The screenshot shows the CodeMeter WebAdmin interface with the following details:

- Header:** CmContainer, CodeMeter WebAdmin, CM logo, navigation menu (Home, Inhalt, Server, Einstellungen, Diagnose, Info, Hilfe), sub-navigation (CmContainer, Lizenzen, Benutzerdaten, Datensicherung).
- Title:** Product Items für Firm Code 10 des CmSticks 1-1123634
- Table:** A table listing three product items. The columns are: Product Code, CmStick, Name, Nutzungen-einheiten, Aktivierungs-datum, Verfalls-datum, Lizenz-anzahl, Feature Map.

Product Code	CmStick	Name	Nutzungen-einheiten	Aktivierungs-datum	Verfalls-datum	Lizenz-anzahl	Feature Map
10	1-1123634	Textanwendung	200	n/a	n/a	10	0x2
13	1-1123634	Tabellenanwendung	30	n/a	n/a	20	n/a
14	1-1123634	Chartanwendung	200	n/a	n/a	23	n/a

Abbildung 242: CodeMeter WebAdmin – "Inhalt | Lizenzen – Firm Item"

12.5.5.1.2 Produkt-Informationen

Diese Seite zeigt detailliertere Information über die Lizenzierung der Produkte eines bestimmten Lizenzgebers.

In der folgenden Abbildung sehen Sie z.B. alle Informationen über das Produkt mit Product Code "13" des Lizenzgebers auf der Firm Item-Ebene mit dem Firm Code "10".

The screenshot shows the CodeMeter WebAdmin interface with the following details:

Product Item Details

Product Item 228:1000 des CmSticks 1-1123634

Product Item Option	Typ	Größe (Bytes)	Abhängigkeiten	Wert
Text		22		Fax Add-on
Feature Map		4	data, serial, counter	0000 0000 0000 0000 0000 0000 0000 0000 (0x0)
Nutzungseinheiten		4	data, serial, counter	0
Aktivierungsdatum		4	data, serial, counter	2008-04-03 13:09:32
Verfallsdatum		4	data, serial, counter	2009-09-13 13:09:32
Nutzungszeitraum		8	data, serial, counter	0 Tage - Startzeitpunkt: n/a
Lizenzzahl		4	data, serial, counter	lokal
Lizenzinformation		10		Leihlizenz
User Data		10		0x00
Protected Data		20	data, serial, counter	0x00
Ausleihzeit Ende	130	64	data, serial, counter	2009-04-13 11:50:05
Ausleihserver (CmStick)	131	20		192.168.0.134 (1-1123622)
Ausleihserver Eintrag	130	64	data, serial, counter	CodeMeter 228:1200
Ausleihe SID	130	64	data, serial, counter	0x00000000000000011
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>

Abbildung 243: CodeMeter WebAdmin – "Inhalt | Lizenzen – Product Item"

Element	Beschreibung
Product Item Options	In der ersten Spalte sehen sie die Product Item Options . Dies sind Lizenz-eigen-schaften, die durch den Lizenzgeber gesetzt worden sind. Zur Verdeutlichung sind in der Abbildung sämtliche Optionen gesetzt. Bei der Auflistung im konkre-

Element	Beschreibung
	ten Fall werden nicht alle Optionen ⁴⁶⁴ aufgelistet . In obigen Abbildung sehen Sie darüber hinaus, dass die Lizenz lokal vom Lizenzserver ausgeliehen ist. Genauere Informationen enthalten die über die Zeileneinträge Ausleihzeit Ende, Ausleihserver (CmContainer), Ausleihserver Eintrag und Ausleihe SID.
Typ	Handelt es sich bei diesen Lizenzenschaften um Datenfelder gibt die Spalte Typ darüber Auskunft, in welchem Bereich des <i>CmContainer</i> diese abgelegt sind.
Größe	Die Spalte gibt die Byte-Zahl an, die eine aufgeföhrte Lizenzenschaft belegt.
Abhängigkeit	Die Spalte informiert darüber, ob der Lizenzgeber in der Programmiersequenz für <i>CmContainer</i> Abhängigkeiten gesetzt hat.
Wert	Die letzte Spalte gibt den eingetragenen Wert der einzelnen Lizenzenschaft an.



Die Lizenzenschaften so wie sie in der obigen Abbildung erscheinen, müssen nicht alle gesetzt sein. Die Anzeige Ihrer Lizzen kann gegebenenfalls abweichen.

12.5.5.2 Benutzerdaten

Die "**Inhalt | Benutzerdaten**"-Seite zeigt Ihnen detaillierte Informationen über Produkte (Lizenzen), die der Benutzer ausschließlich mit seinem *CmDongle* Passwort nutzen kann. Dieser Lizenzcontainer wird über die Zahl "0" identifiziert.

Wählen Sie über **CmDongle** den gewünschten *CmDongle* aus, um Informationen über "Ihre" Lizenzen einzusehen. Navigations- und Eintragsstrukturen der Lizenzen sind analog zur [Anzeige lokaler Lizenzen](#)⁴⁸³.

The screenshot shows the CodeMeter WebAdmin interface with the 'Inhalt' tab selected. At the top, there's a navigation bar with links for Home, Inhalt, Server, Einstellungen, Diagnose, Info, and Hilfe. Below the navigation is a sub-navigation bar with links for CmContainer, Lizenzen, Benutzerdaten, and Datensicherung. A search bar labeled 'CmContainer:' contains the value '1-1440495'. The main content area is titled '0 | User Data' and contains a table with the following data:

Product Code	Name	Nutzungseinheiten	Verfallsdatum	Aktivierungsdatum	Lizenzanzahl
0	-	n/a	n/a	n/a	1
1000	-	n/a	n/a	n/a	1
100000	CM Password Manager	n/a	n/a	n/a	1
100106	Steganos	n/a	n/a	n/a	1

Abbildung 244: CodeMeter WebAdmin – "Inhalt | Benutzerdaten"

12.5.6 Lizenzanzeige im Netzwerk

Die "Server | ..."-Seiten zeigt Ihnen Informationen über vorhandene Netzwerklicenzen sowie deren aktuellen Belegung an.



Netzwerklicenzen in CmContainern sind nur dann von anderen PCs nutzbar, wenn CodeMeter Lizenzserver als [Netzwerkserver](#) gestartet wurde.

Die Anzeige von Netzwerklicenzen ist in zwei Kategorien unterteilt:

- geordnet nach Lizenzgeber und Lizenzen ([Cluster](#)⁴⁸⁸).
- geordnet nach Benutzern von Lizenzen ([Benutzer](#)⁴⁹⁰).

12.5.6.1 Cluster - Lizenzen zusammengefasst

The screenshot shows the 'Server | Cluster' section of the CodeMeter WebAdmin. The table lists licenses for 'fs1.wibu.local' categorized by manufacturer (Hersteller) and article number (Artikel).

Product Code	Name	Feature Map	Lizenzen	Status						
				User Limit (Ausgeliehen)	No User Limit	Exklusiv	Shared	Frei		
10 Hersteller 1										
10	Textanwendung	0x2	10	0 (-)	1	1	0	9	Details	
13	Tabellenanwendung	-	20	5 (-)	3	0	1	14	Details	
14	Chartanwendung	0x6	23	2 (1)	2	0	1	20	Details	
228 Hersteller 2										
67	Druckanwendung	0x8	50	1 (-)	0	0	0	49	Details	
100003 Bundling Articles										
1	SecuriKey Lite	0x1	1	0 (-)	0	0	0	1	Details	

Stand 27.Feb.2009 11:10:57

Abbildung 245: CodeMeter WebAdmin – "Server | Cluster"

Die **"Server | Cluster"**-Seite zeigt Ihnen alle vorhandenen Netzwerklicenzen sowie deren Belegung an, geordnet nach Lizenzgebern und zugehörigen Lizzenzen. Neben beschreibenden Informationen zu **Product Code**, **Name** und **Feature Map**, zeigt die **Lizenzen**-Spalte die jeweilige Gesamtanzahl von vorhandenen Netzwerklicenzen an.

Belegte und freie Lizenzen

Zusätzlich gliedert der **Status**-Bereich die Lizzenzen nach Zugriffsmodi (**User Limit**, **No User Limit**, **Exklusiv**, **Shared**) und zeigt verfügbare, **freie Lizenzen**⁴⁶⁴ an .

Ausgeliehene Lizzenzen

Darüber hinaus sehen Sie hier im **Status**-Bereich auch, ob und wenn ja welche Lizzenzen in welchem Umfang vom Lizenzserver für die lokale Benutzung ausgeliehen sind.

In der obigen Abbildung sehen Sie z.B. dass von den insgesamt 20 Lizzenzen des Herstellers 1 für die Tabellenanwendung noch 14 frei sind. Insgesamt greifen 9 Instanzen der Anwendung auf Lizzenzen zu, aber es werden nur 6 gezählt, da die 3 vom Status **No User Limit** zusammen keine Lizenz belegen.

Klicken Sie auf die "[Details](#)⁴⁶⁹" Schaltfläche, um in der folgenden Abbildung detailliertere Information über die Belegung einer Lizenz zu bekommen.

12.5.6.1.1 Session Details

Die folgende Abbildung zeigt detailliertere Information über die Belegung einer Lizenz an.

The screenshot shows the CodeMeter WebAdmin interface with the following details:

- Header:** CodeMeter WebAdmin, CM logo, navigation menu (Home, Inhalt, Server, Einstellungen, Diagnose, Info, Hilfe), user information (Cluster | Benutzer).
- Title:** Lizenz Details CmContainer 1-1123634
- Table Headers:** ID, Client, Client Prozess ID, Anwendungs Information, Zugriffsmodus, Erster Zugriff, Letzter Zugriff, Ablaufzeit, Aktion.
- Data:** A table listing 8 entries of license usage. The first 7 entries are for Client 192.168.0.134 and the last one is for Client 192.168.0.33. The table includes columns for the number of users, the type of access (User Limit or Station Share), and the start and end dates of the license.
- Bottom Text:** Stand 03.Apr.2009 10:42:22

Abbildung 246: CodeMeter WebAdmin – "Server | Cluster - Details"

In der Beispiel-Abbildung sehen Sie:

- die Lizenzen der Anwendung stammen vom Lizenzgeber mit dem Firm Code 10 und umschreiben das Produkt mit dem Product Code 14 als Modul (0x6 als Feature Map).
- die Lizenzen befinden sich im *CmContainer* mit der Seriennummer 1-1123634.
- insgesamt 2 Clients, identifiziert über die **ID**, **Client** (192.168.0.134 und 192.168.0.33) und **Client Prozess ID** greifen auf die Anwendung Charts zu.
- von den insgesamt 23 verfügbaren Lizenzen sind 5 belegt, 18 sind frei und verfügbar.
- Client 192.168.0.33 belegt 1 exklusive Lizenz, Client 192.168.0.134 nutzt die Anwendung in 6 Instanzen, belegt aber aufgrund der **Zugriffsmodi** nur 4 Lizenzen
- Client 192.168.0.134 hat eine Lizenz im *CmContainer* [1-1123634] eine Lizenz ausgeliehen, die bis zum 12. April 2009 gültig ist.
- Client 192.168.0.33 hat das erste Mal auf die Anwendung zugegriffen (**Erster** und **Letzter Zugriff** sind datumsgleich).
- Client 192.168.0.134 hat laut der Spalte **Erster Zugriff** bereits vorher auf die Anwendung zugegriffen.

Löschen

Über die "Löschen" Schaltfläche der **Aktion**-Spalte können Sie einzelne Zugriffe löschen und dadurch belegt Lizenzen wieder freigeben.

 Ausgeliehene Lizenzen können Sie durch Löschen nicht freigeben. Sie müssen dazu erst wieder zurückgebucht sein.

Das ist beispielsweise notwendig, wenn alle Lizenzen belegt sind, aber noch eine weitere Anwendung gestartet werden soll.

 Nach Löschen eines Zugriffs wird die Lizenz freigegeben, ist somit verfügbar, und der Client der Anwendung erhält eine entsprechende Fehlermeldung.

12.5.6.2 Aktuell angemeldete Benutzer

Die "**Server | Benutzer**"-Seite zeigt Ihnen alle vorhanden Netzwerklicensen geordnet nach aktuell angemeldeten Benutzern (**Clients**).



CMStick	Firm Item	Product Item	Client	Zugriffsmodus	
1-1123634	10: Hersteller 1	10: Textanwendung	192.168.0.33	No User Limit	Details
1-1123634	10: Hersteller 1	10: Textanwendung	192.168.0.134	Exclusive	Details
1-1123634	10: Hersteller 1	13: Tabellenanwendung	192.168.0.134	No User Limit	Details
1-1123634	10: Hersteller 1	13: Tabellenanwendung	192.168.0.134	User Limit	Details
1-1123634	10: Hersteller 1	13: Tabellenanwendung	192.168.0.134	User Limit	Details
1-1123634	10: Hersteller 1	13: Tabellenanwendung	192.168.0.134	Station Share	Details
1-1123634	10: Hersteller 1	13: Tabellenanwendung	192.168.0.134	Station Share	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	No User Limit	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	No User Limit	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	User Limit	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	User Limit	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	Station Share	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.134	User Limit	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	192.168.0.33	Exclusive	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	127.0.0.1	Station Share	Details
1-1123634	10: Hersteller 1	14: Chartanwendung	127.0.0.1	Station Share	Details
1-1123634	228: Hersteller 2	67: Druckanwendung	192.168.0.33	User Limit	Details

Stand 03.Mär.2009 08:52:42

Abbildung 247: CodeMeter WebAdmin – "Server | Benutzer"

Hier erhalten Sie [beschreibenden Informationen](#) zu **CmContainer**, Lizenzgeber (**Firm Item**), Lizenz (**Product Item**) und **Zugriffsmodus**. Klicken Sie auf die "[Details](#)"-Schaltfläche, um in der obigen Abbildung detailliertere Information über die Belegung einer Lizenz zu erhalten.

12.5.6.3 Lizenzverfolgung

Die "**Server | Lizenzverfolgung**"-Seite erlaubt Ihnen nachzuverfolgen, wer, wann, von wo aus, wie oft Lizenzen CodeMeter-geschützter Anwendungen über einen Server nutzt bzw. wieviel Lizenzanfragen zurückgewiesen wurden.

Das Mitprotokollieren von Lizenzierungsdaten muss zusammen mit *CodeMeter License Server* aktiviert werden. Dies bewerkstelligen Sie über die direkte Aktivierung in der *CodeMeter® Profiling-Umgebung*.

Für Windows Betriebssysteme finden Sie die Profiling-Einträge in der Registry abgespeichert. Für andere Betriebssysteme werden diese Einträge in der Datei *server.ini* gesetzt. Die folgende Tabelle listet die entsprechenden Orte bzw. Dateien auf.

Betriebssystem	Registry / Server.ini-Eintrag
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

Es existieren die folgenden zwei relevanten Profiling-Einträge für das License Tracking.

Eintrag	Format	Wert
LogLicenseTracking	[DWord] [0 ; 1]	<p> Die Standardeinstellung besitzt den Wert 0 und die Protokollierung für das License Tracking ist abgeschaltet.</p>
LogLicenseTrackingPath	[SZ]	<p> Der Standard-Pfad für Windows Betriebssysteme ist %ProgramData%\CodeMeter\LicenseTracking.</p> <p> Für andere Betriebssysteme besitzt der Standard-Pfad den gleichen Wert wie der Eintrag für die allgemeine Protokollierung LogPath.</p>

 Bitte beachten Sie das die vorgenommenen Änderungen an den Einstellung erst dann wirksam werden, wenn Sie *CodeMeter License Server* neu starten.

Die Lizenzanfragen werden auf Grundlage auswählbarer Protokollierungsdateien und Lizenzen in einem Bericht grafisch und im Detail angezeigt. Der Bericht kann dazu dienen, aus Informationen zu Lizenzanfragen und -zugriffen mögliche Lizenzkosten zu senken sowie Prognosen zu erstellen.

Die Zahl und der Ursprung belegter, zurückgewiesener sowie freigegebener Lizenzen lässt sich über eine separate Navigation zeitlich in verschiedenen Ansichtsmodi (monatlich, täglich, stündlich) nachverfolgen. Das Klicken auf die in der Grafik angezeigten Balken zeigt weitere Details der Lizenzverwendung.

Zur Nutzung der Lizenzverfolgung gehen Sie wie folgt vor:

1. Wählen der Protokolldatei über das "**Auswahl der Log-Datei**"-Feld.

Auswahl der Log-Datei

2013-10-10T14:41 - 2013-10-10T14:46

Klicken der "Neu laden"-Schaltfläche aktualisiert die angezeigten Log-Dateien.

2. Wählen der Lizenz, die nachverfolgt werden soll, über das "Lizenzen auswählen"-Feld.

Lizenzen auswählen

1-1234944-10-13 LQ:11 (Lizenzdemo)

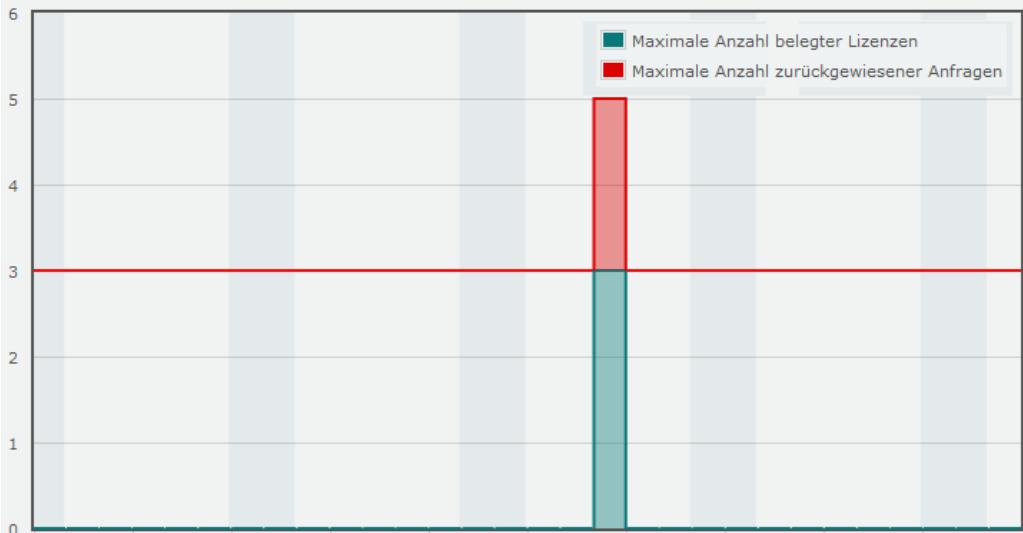
3. Klicken der "Bericht erstellen"-Schaltfläche.

Der separate Bereich **Navigation**:



- informiert über den Ansichtsmodus (Monat, Tag, Stunde),
- zeigt den beobachteten Zeitraum (Von-Bis) an,
- erlaubt über Pfeiltasten zeitlich nach vorne und zurück zu blättern und zum jeweilig vorhergehenden Ansichtsmodus zu wechseln.

Unterhalb des Eingabebereiches erscheint eine **Balkendiagramm**-Grafik, die die maximale Anzahl an belegten und zurückgewiesenen Anfragen anzeigt.



Die Standard-Einstellung steht auf dem Ansichtsmodus Monat.

- Überfahren der farbigen Balken öffnet einen überlagernden Dialog, der zusätzliche Informationen anzeigen.



- Links klicken wechselt in den Ansichtsmodus Tag.

Zum Zurückkehren in den Ansichtsmodus Monat kann das Pfeilssymbol des Bereiches Navigation verwendet werden.

- Erneutes Überfahren und Linksklicken wechselt in den Ansichtsmodus Stunde.



- Erneutes Überfahren und Linksklicken öffnet einen separaten Bereich Details.



Tabellarisch werden hier Detailinformationen zu Aktive Benutzer, Zurückgewiesene Anfragen und

Alle Ereignisse aufgelistet.

Details

Zeitraum: **2013-10-10T15:40:00 - 2013-10-10T15:40:59**

Maximale Anzahl gleichzeitig belegter Lizenzen: **3**

Maximale Anzahl zurückgewiesener Anfragen verschiedener Benutzer: 1

Aktive Benutzer (Lizenz, ID, Client, Benutzer)

Aktive Benutzer

ID	Client	Benutzer
37	10.49.12.17	wv
38	10.49.12.17	wv
39	10.49.12.17	wv
40	10.49.12.17	wv
41	10.49.12.17	wv
42	10.49.12.17	wv
43	10.49.12.17	wv
44	10.49.12.17	wv
45	10.49.12.17	wv
46	10.49.12.17	wv
47	10.49.12.17	wv
48	10.49.12.17	wv
49	10.49.12.17	wv
50	10.49.12.17	wv
51	10.49.12.17	wv
52	10.49.12.17	wv
53	10.49.12.17	wv
54	10.49.12.17	wv
55	10.49.12.17	wv
56	10.49.12.17	wv
57	10.49.12.17	wv
58	10.49.12.17	wv

Zurückgewiesene Anfragen (Sekunde, Ereignistyp, Lizenz, Client, Benutzer)

Zurückgewiesene Anfragen

Sekunde	Event-Typ	Client	Benutzer
3	Ablehnung	192.168.0.18	fs
7	Ablehnung	192.168.0.18	fs
26	Ablehnung	192.168.0.18	fs
33	Ablehnung	192.168.0.18	fs
37	Ablehnung	192.168.0.18	fs
41	Ablehnung	192.168.0.18	fs
44	Ablehnung	192.168.0.18	fs
48	Ablehnung	192.168.0.18	fs
52	Ablehnung	192.168.0.18	fs

Alle Ereignisse (Sekunde, Ereignistyp, Lizenz, ID, Client, Benutzer)

Alle Ereignisse

Sekunde	Ereignistyp	ID	Client	Benutzer
1	Freigabe	37		
1	Freigabe	38		
3	Zugriff	39	10.49.12.17	wv
4	Zugriff	40	10.49.12.17	wv
4	Freigabe	39		
4	Freigabe	40		
12	Zugriff	41	10.49.12.17	wv
13	Zugriff	42	10.49.12.17	wv
13	Zugriff	43	10.49.12.17	wv
18	Freigabe	41		
19	Freigabe	42		
19	Freigabe	43		
20	Zugriff	44	10.49.12.17	wv
21	Zugriff	45	10.49.12.17	wv
21	Zugriff	46	10.49.12.17	wv
26	Freigabe	44		
27	Freigabe	45		
27	Freigabe	46		
28	Ablehnung		10.49.12.17	fs
29	Zugriff	47	10.49.12.17	wv

Die Detail-Ansicht verwendet die folgenden Elemente:

Element	Beschreibung
ID	kennzeichnet eindeutig und unterscheidet Anfrage- und Zugriffsprozesse.
Client	identifiziert die IP Adresse der anfragenden / zugreifenden Maschine.
Benutzer	identifiziert den anfragenden / zugreifenden Benutzer.

Element	Beschreibung
Sekunde	informiert über den Sekunden-Wert der Anfrage / des Zugriffs.
Ereignistyp	<p>Ablehnung zeigt, dass ein Benutzer eine Lizenzzugriff-Anfrage hat, die Lizenz aber nicht zugewiesen werden konnte, weil keine Lizenzen auf dem Server verfügbar sind. Er zeigt nicht an, dass auf Lizenzen zugegriffen werden soll, die auf diesem Server nicht vorhanden sind. Der Ablehnungseintrag wird in dem Moment geschrieben, indem ein Lizenzzugriff fehlgeschlagen ist.</p> <p>Zugriff Eintrag zeigt, dass ein Benutzer eine Lizenz auf dem Server belegt.</p> <p>Freigabe Eintrag zeigt, dass ein Benutzer eine zuvor auf dem Server belegte Lizenz freigegeben hat.</p>

12.5.7 Diagnose

Protokollierung

Die "Diagnose | Protokoll"-Seite erlaubt Ihnen alle Vorgänge, die den Dienst *CodeMeter License Server* betreffen in ein Protokoll mitschreiben zu lassen. Dies bietet Informationen, die Sie bei einer eventuellen Suche nach möglichen Fehlern unterstützt.



Damit *CodeMeter WebAdmin* auf dieser Seite ein Protokoll anzeigt, muss vorher in *CodeMeter Kontrollzentrum* diese Funktion [aktiviert](#)⁴⁴⁷ werden. Dort finden Sie auch weitere Hinweise zum Speichern einer Protokollierungsdatei.

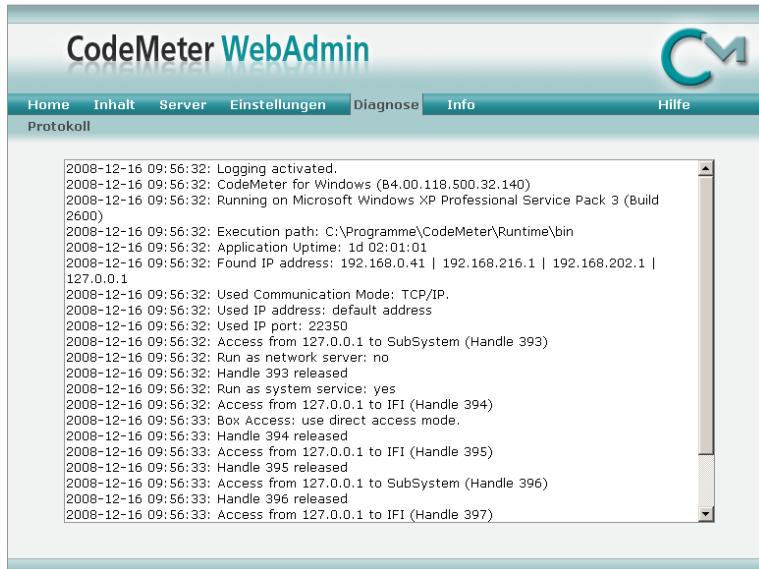


Abbildung 248: CodeMeter WebAdmin – "Diagnose | Protokoll"

12.5.8 Datensicherung

Die "**Inhalt | Datensicherung**"-Seite ermöglicht Ihnen die Sicherung persönlicher Daten, die sich auf einem *CmDongle* befinden, oder gesicherte Daten wieder auf den *CmDongle* zurückzuspielen. Bitte beachten Sie, dass damit nur die Benutzerdaten des *CmDongles* gesichert und zurückgeschrieben werden, nicht aber Lizenzinformation von anderen Lizenzgebern. Die Sicherung bezieht sich ausschließlich auf den Lizenzcontainer mit dem Firm Code "0".

Um Lizenzen wiederherzustellen, die nicht im persönlichen Bereich liegen (Firm Item-Ebenen, die nicht den Firm Code "0" haben), kontaktieren Sie bitte den WIBU Support.



Abbildung 249: CodeMeter WebAdmin – "Inhalt | Datensicherung"

Element	Beschreibung
CmDongle	Wählen Sie über CmDongle den gewünschten <i>CmDongle</i> aus, für den die Sicherung erstellt werden bzw. wiederhergestellt werden soll.
Jetzt sichern	<ol style="list-style-type: none"> Klicken Sie auf die "Jetzt sichern"-Schaltfläche, um eine sofortige Sicherung der persönlichen <i>CmDongle</i> Daten (Benutzerdaten) durchzuführen. Es wird Ihnen zusätzlich das Datum und die Uhrzeit der letzten Sicherung angezeigt. Bestätigen Sie im folgenden Dialog die Erstellung der Sicherheitskopie. 
Durchsuchen / Wiederherstellen	<ol style="list-style-type: none"> Klicken Sie auf die "Durchsuchen"-Schaltfläche, um die Sicherheitskopie, die wiederhergestellt werden soll, auszuwählen. Der Speicherort des Sicherungsordners wird angezeigt. Klicken Sie auf die "Wiederherstellen"-Schaltfläche, um den Einspielvorgang zu starten. Bestätigen Sie den folgenden Dialog mit der "OK"-Schaltfläche. 

Element	Beschreibung
	<p>Wenn Sie eine Sicherheitskopie des <i>CmDongles</i> einspielen, gehen alle Änderungen vom Zeitpunkt der Sicherung an verloren.</p> <p>Geben Sie das Kennwort des <i>CmDongles</i> ein, in den die Sicherheitskopie überspielt werden soll.</p>  <p>Sie können Ihre gesicherten Daten auch auf einen anderen <i>CmDongle</i> überspielen, beachten Sie aber dabei, dass der zweite <i>CmDongle</i> das "gleiche Kennwort" haben muss!</p>

12.5.9 Info

Die "**Info**"-Seite gibt einen Produktüberblick sowie Überblick über wichtige Wibu-Systems Adressen.

12.5.10 Hilfe

Die "**Hilfe**"-Seite lässt sich von jeder anderen Seite erreichen und gibt kontextsensitive Hilfestellungen zu *CodeMeter WebAdmin*.

12.6 CmDust

Manchmal ist es notwendig, dass Sie unseren Support um Hilfe beim Einsatz von *CodeMeter®* benötigen. Um es unserem Support zu erleichtern Ihr Problem zu identifizieren, wurde das Programm *CmDust* (**CodeMeter Enduser Support Tool**) für die Kommandozeilen-Eingabe-Aufforderung entwickelt.



Es werden keine geheimen Informationen zu Wibu-Systems übertragen. Sie können alle im Klartext gespeicherten Informationen überprüfen.

CmDust unter Windows

CmDust kann durch Nutzung des "**Start | Alle Programme | CodeMeter | Tools**"-Eintrages im Startmenü aufgerufen werden. Danach öffnet sich der Windows Explorer mit der Datei *CmDust-Result.log*. Die Textdatei *CmDust-Result.log* liegt im jeweiligen Benutzerverzeichnis, das sich nach Ausführen von *CmDust* automatisch öffnet.

Alternativ können Sie die Datei auch über das Kommandozeilenwerkzeug [cmu](#)⁵⁰³ erzeugen.

Die erstellte Datei kann für Analysen zu Wibu-Systems geschickt werden.

CmDust unter Mac OS

Für Mac OS erstellen Sie die *CmDust*-Datei über das [cmu](#)⁵⁰²-Kommandozeilen-Programm. Das Aufruf von *cmu* ist im Suchpfad hinterlegt.

Um ein *CmDust*-Log zu erstellen, gehen Sie wie folgt vor:

1. Öffnen Sie die *cmu*-Kommandozeile
2. Geben Sie den folgenden Befehl ein

```
cmu --cmdust
```

Über den Zusatz `--file` ist die Benennung und der Speicherort möglich.

Standardmäßig wird ohne den Zusatz die Datei in das Verzeichnis geschrieben, von dem aus Sie den Befehl ausgeführt haben.

3. Diese Datei kann für Analysen zu Wibu-Systems geschickt werden.

CmDust unter Linux, Sun

Für die Betriebssysteme Linux und Sun erstellen Sie die *CmDust*-Datei über das [cmu](#)⁵⁰²-Kommandozeilen-Programm. Das Aufruf von *cmu* ist im Suchpfad hinterlegt.

1. Öffnen Sie die *cmu*-Kommandozeile

2. Geben Sie den folgenden Befehl ein

```
cmu --cmdust
```

Über den Zusatz `--file` ist die Benennung und der Speicherort möglich.

Standardmäßig wird ohne den Zusatz die Datei in das Verzeichnis geschrieben, von dem aus Sie den Befehl ausgeführt haben.

3. Diese Datei kann für Analysen zu Wibu-Systems geschickt werden.

Die folgenden Einstellungen werden von *CmDust* ausgelesen:

- Informationen über das Betriebssystem: Version, Installierte Service Packs, Spracheinstellungen.
- *CodeMeter®*-relevante Registry-Einträge: Installationspfad, Einstellungen zu *CodeMeter Lizenzserver* und *CodeMeter WebAdmin*, Sicherungs- und HTTP Einstellungen.

- AddOns: Informationen über alle installierte *CodeMeter®* AddOns.
- Informationen über *CodeMeter®* und *CmContainer*: Software und Hardware Version und alle Einträge der verbundenen *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: German
Current User: German

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"
```

12.7 CMU - CodeMeter Universal Support Tool

Sie haben auch die Möglichkeit, einige CodeMeter Kontrollzentrum Funktionen alternativ über das kommandozeilenbasierte *CodeMeter Universal Support Tool* (*cmu*) ausführen zu lassen.

cmu unterstützt Sie beim:

- Auflisten von Inhalten in *CmContainer*n
- Erstellen einer einfachen Testumgebung für *CmContainer*
- Durchführen einer Zeitaktualisierung und Erstellen und Import von Lizenzanforderungs- und Lizenzaktualisierungsdateien (Remote Context und Update-Dateien, *.WibuCmRac; *.WibuCmRaU)

Sie rufen *cmu* im Verzeichnis %\Program Files%\CodeMeter\Runtime\bin über den Befehl *cmu [32].exe* auf.

Alternativ rufen Sie *cmu* über den Systemmenü-Eintrag **"Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt"** auf.

Für die Betriebssysteme Mac OS, Linux und Sun ist dieser Befehl im Suchpfad hinterlegt.

Die folgende Liste zeigt vorhandenen *cmu*-Befehle.

Befehl	Beschreibung
/h oder --help	zeigt diese Hilfe im Eingabefenster an.
/v oder --version	zeigt die Versionen aller verfügbaren CodeMeter® Komponenten an.
/l oder --list	listet alle verbundenen <i>CmContainer</i> in Form der Seriennummer auf.
/x oder --list-content	listet die Inhalte aller verbundenen <i>CmContainer</i> auf.
/k oder --list-server	listet alle verfügbaren Netzwerk Lizenzserver ⁴⁸⁸ auf.
/n oder --list-network	listet die gesamten Netzwerk-Lizenzinformation ⁴⁸⁸ auf.
/c <FI> oder --context <FI>	erzeugt eine Lizenzanforderung über CmFAS ⁴⁵⁷ für eine Lizenzaktualisierung zur Erstellung einer Lizenzaktualisierungsdatei für den Lizenzcontainer, das Firm Item <FI>. Mit der Option --file wird die Ausgabedatei angegeben. Ohne Angabe der Option wird die Standardausgabe verwendet (stdout).
/i oder --import	importiert eine über CmFAS empfangene Lizenzaktualisierungsdatei ⁴⁶¹ für die verfügbare CodeMeter®-Lizenz. Mit der Option --file wird der Dateiname angegeben. Die Aktualisierung kann für einen <i>CmDongle</i> oder eine <i>CmActLicense</i> Lizenzdatei durchgeführt werden.
/d oder --firmware-update	startet die Aktualisierung der Firmware eines <i>CmContainers</i> .
/u oder --time-update	startet die Aktualisierung der zertifizierten Zeit ⁴⁷⁹ in jedem verbundenem <i>CmContainer</i> .
/e <s> oder --enable <s>	erlaubt die Aktivierung oder Deaktivierung (Enabling) des ausgewählten <i>CmContainers</i> . Dabei muss das CodeMeter® Passwort angegeben werden. Der erforderliche neue Enabling-Status wird über den Parameter <s> angegeben wobei die Werte s die Werte 1 (disable), 2 (temporär enable), 3 (enable) annehmen kann.
/t <no> oder --test<no>	startet für jeden verbundenem <i>CmContainer</i> einige einfache Tests. Die Anzahl der Tests wird über den Parameter <no> angegeben. Dazu muss der <i>CmContainer</i> (temporär) enabled sein.

Befehl	Beschreibung
/vv oder --cmdust	erstellt einen <i>CmDust</i> Bericht. Dieser Bericht macht Sinn und ist erforderlich bei Anfragen beim Support. Es wird empfohlen einen <i>CmDust</i> Bericht zu erstellen bevor den Wibu-Systems Support kontaktiert. Mit der Option --file kann das Ergebnis in eine Textdatei geschrieben werden.
--borrow	erlaubt die Lizenzausleihe von einem Lizenzserver auf den lokalen PC. Sie müssen den Firm Code und den Product Code der Lizenz über die Optionen -firmcode und -productcode angeben. Als zusätzliche Option kann die Feature Map über die Option --featuremap angegeben werden. Darüberhinaus müssen Sie die Seriennummer des Client-CmContainers und den Server-Namen über die Optionen --serial und --server angeben.
--return	gibt die ausgeliehene Lizenz an den Lizenzserver zurück. Sie müssen den Firm Code und den Product Code der Lizenz über die Optionen --firmcode und -productcode angeben sowie die Seriennummer des Client-CmContainers über die Option --serial .
--borrowlist	listet die ausgeliehenen Lizenzen für den Client und den Server auf.
--enabling	listet die Enabling Status-Zustände aller verbundenen CmContainer auf. Kombiniert mit dem Befehl -x können aus dem CmContainer-Inhalt zusätzliche Enabling-Informationen angezeigt werden.
--create-io	wird in Kombination mit der Option --file verwendet und macht Sinn nur im Zusammenhang mit den Hardware-Varianten CmCard/SD oder CmCard/CF. Es wird eine neue codemtr.io Datei erstellt. Rufen Sie diesen Befehl nur dann auf, wenn die codemtr.io Datei gelöscht ist.
--detect-proxy	ermittelt die Proxy-Einstellungen des Systems. Ohne Angabe von Optionen wird die Standardausgabe verwendet (stdout). Die Option --write sichert die Einstellungen über das CodeMeter® Profiling.
--delete-cmact-license	löscht eine CmActLicense-Lizenz, die sie über die Angabe von --serial angeben.  Eine CmActLicense-Lizenz, die Sie einmal gelöscht haben, kann nicht wiederhergestellt werden.

Die folgende Liste zeigt vorhandene cmu-Optionen:

Optionen	Beschreibung
/f <file> oder --file <file>	Zusätzliche Option, die das Befehlsresultat in eine Datei <file> schreibt. Diese Option macht z.B. Sinn in Kombination mit den Befehlen --context , --import , --cmdust .
/s <serial> oder --serial <serial>	Zusätzliche Option, die bestimmt, dass ein Befehl nur für einen CmContainer gilt, dessen Seriennummer <serial> angegeben wird, z.B. "1-10234242".
/p <pwd> oder --password <pwd>	Zusätzliche Option in Kombination mit den Befehlen --enable und --firmware-update . Die Option definiert das notwendige CodeMeter® Passwort für diesen Befehl.
--firmcode <fc>	Zusätzliche Option in Kombination mit den Befehlen --borrow oder --return , die den Firm Code der ausgeliehenen Lizenz definieren.
--productcode <pc>	Zusätzliche Option in Kombination mit den Befehlen --borrow oder --return , die den Product Code der ausgeliehenen Lizenz definieren.
--featuremap <fm>	Zusätzliche Option in Kombination mit den Befehlen --borrow oder --return ,

Optionen	Beschreibung												
	die die Feature Map der ausgeliehenen Lizenz definiert.												
--server <servername>	Zusätzliche Option um eine Lizenz von einem anderen Server auszuleihen. Wird zusammen mit dem Befehl --borrow benutzt.												
--write	Zusätzliche Option, die zusammen mit dem Befehl --detect-proxy verwendet wird und die Einstellungen über das CodeMeter®-Profiling abspeichert. Diese Einstellungen werden nur geschrieben, wenn zuvor kein Proxy im Profiling gesetzt wurde. Zum Überschreiben der Einstellungen wird die Option --force verwendet.												
--force	Zusätzliche Option, die zusammen mit dem Befehl --detect-proxy verwendet wird und bereits bestehende Proxy-Einstellungen im CodeMeter-Profiling überschreibt.												
--show-config-disk	Zeigt die aktuellen Einstellungen zum Beispiel zu wechselbaren/festen (removable/fixed) Laufwerken oder zum Typ des definierten Master Boot Records (MBR). Diese Option betrifft das Verhalten von virtuellen Flash-Speicher-Partitionen. Verwendbar nur für <i>CmStick</i> und <i>CmStickIM</i> .												
--set-config-disk <parameter>	Erlaubt Ihnen ein spezielles Verhalten der virtuellen Flash-Speicher-Partitionen zu bestimmen, z.B. Laufwerkseinstellungen, Boot Code oder Aktivierungen (nur <i>CmDongle</i>).  Bitte beachten Sie, dass hier ein erneutes Anstecken des <i>CmDongles</i> erforderlich ist. <table border="1"> <thead> <tr> <th>Beschreibung</th><th>Parameter</th></tr> </thead> <tbody> <tr> <td>Laufwerkseinstellungen</td><td>RemovableDisk,LocalDisk</td></tr> <tr> <td>Boot Code</td><td>Int18Boot ,ZeroBoot,LoopBoot,SwapBoot,VbrBoot</td></tr> <tr> <td>Aktivierung</td><td>ActivePartition,InactivePartition</td></tr> <tr> <td>FAT</td><td>Fat16,Fat32</td></tr> <tr> <td>USB-Kommunikationsgeräteklaasse</td><td>HidCommunication; MsdCommunication</td></tr> </tbody> </table>	Beschreibung	Parameter	Laufwerkseinstellungen	RemovableDisk,LocalDisk	Boot Code	Int18Boot ,ZeroBoot,LoopBoot,SwapBoot,VbrBoot	Aktivierung	ActivePartition,InactivePartition	FAT	Fat16,Fat32	USB-Kommunikationsgeräteklaasse	HidCommunication; MsdCommunication
Beschreibung	Parameter												
Laufwerkseinstellungen	RemovableDisk,LocalDisk												
Boot Code	Int18Boot ,ZeroBoot,LoopBoot,SwapBoot,VbrBoot												
Aktivierung	ActivePartition,InactivePartition												
FAT	Fat16,Fat32												
USB-Kommunikationsgeräteklaasse	HidCommunication; MsdCommunication												
--check-cm-integrity	Erlaubt Ihnen die CodeMeter®-Signatur zu überprüfen.												

Anwendungs-Beispiele

Aktion	Parameter
Anzeigen der <i>cmu</i> -Optionen	<i>Cmu[32].exe -h</i>
Erstellen einer CodeMeter®-Remote Activation-Kontextdatei (hier:1-1040870.WibuCmRaC) für den Firm Code 10 (Firm Item Ebene)	<i>Cmu[32].exe -c10 -f1-1040870.WibuCmRaC</i>
Importiert eine CodeMeter®-Remote-Activation-Update-Datei (hier:1-1040870.WibuCmRaU) -> programmiert den verbundenen <i>CmContainer</i> um	<i>Cmu[32].exe -i -f1-1040870.WibuCmRaU</i>
Zeigt die Versionen der aktuellen CodeMeter®-Komponenten.	<i>cmu32 --version</i>
Listet alle verfügbaren CodeMeter-Netzwerk-Lizenzserver auf und falls vorhanden auch eine Liste der verbundenen Lizzenzen.	<i>cmu32 --list-server --list-content</i>

Aktion	Parameter
Startet 100 einfache Tests. Die Tests werden nur ausgeführt für den <i>CmContainer</i> mit der angegebenen Seriennummer 1-233232.	cmu32 --test 100 --serial 1-233232
Ändert für den <i>CmContainer</i> 1-2345 den Enabling Status auf "temporär enabled" und nutzt dazu das <i>CodeMeter®-Password</i> "SECRET".	cmu32 --enable2 --serial 1-2345 --password SECRET

12.8 CodeMeter License Tracking

Beginnend mit der Version 4.50 führt *CodeMeter®* ein License Tracking ein, das die Auswertung von Lizenzierungsdaten auf der Grundlage von strukturierten Protokolldateien (logfiles) erlaubt. Damit lässt sich feststellen, wie Lizenzen benutzt werden.

Wibu-Systems bietet jedoch keine eigene separate Anwendung zur Auswertung des License Tracking an. Kunden, die eine Auswertung wünschen, haben entweder die Option, die strukturierten Daten selbst auszuwerten, oder diese Daten in ein Format zu überführen, das den Datenimport in Werkzeuge von Drittanbietern ermöglicht.

Derzeit werden die Inhalte der Protokolldateien lokal gespeichert. Für künftige Versionen ist geplant, dass die Inhalte auch über HTTP-Zugriffe und Aufrufe protokolliert werden (Echtzeit-Historie).

 Wenn die Notwendigkeit besteht, die Protokolldateien von einem anderen System aus zu lesen, dann muss das Verzeichnis, in dem die Protokolldateien gespeichert werden, innerhalb des lokalen Netzwerkes für den Lesezugriff freigegeben werden.

Die folgenden Abschnitte:

- [zeigen die Konfiguration des License Tracking](#)⁵⁰⁵
- [führen in die Definition und Wertebereiche ein, die in den Protokolldateien verwendet werden](#)⁵⁰⁶
- [beschreiben die einzelnen Protokoll-Eintragstypen](#)⁵⁰⁸

12.8.1 Voraussetzungen

Um das *CodeMeter®*-Feature License Tracking nutzen zu können, wird mindestens die *CodeMeter License Server*-Version 4.50 benötigt.

12.8.2 Konfiguration

Das Mitprotokollieren von Lizenzierungsdaten muss zusammen mit *CodeMeter License Server* aktiviert werden. Dies bewerkstelligen Sie über die direkte Aktivierung in der *CodeMeter® Profiling*-Umgebung.

12.8.2.1 Profiling

Für Windows Betriebssysteme finden Sie die Profiling-Einträge in der Registry abgespeichert. Für andere Betriebssysteme werden diese Einträge in der Datei *server.ini* gesetzt. Die folgende Tabelle listet die entsprechenden Orte bzw. Dateien auf.

Betriebssystem	Registry / Server.ini-Eintrag
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

Es existieren die folgenden zwei relevanten Profiling-Einträge für das License Tracking.

Eintrag	Format	Wert
LogLicenseTracking	[DWord] [0;1]	 Die Standardeinstellung besitzt den Wert 0 und die Protokollierung für das License Tracking ist abgeschaltet.
LogLicenseTrackingPath	[SZ]	 Der Standard-Pfad für Windows Betriebssysteme ist %ProgramData%\CodeMeter\LicenseTracking. Für andere Betriebssysteme besitzt der Standard-Pfad den gleichen Wert wie der Eintrag für die allgemeine Protokollierung LogPath.

 Bitte beachten Sie das die vorgenommenen Änderungen an den Einstellung erst dann wirksam werden, wenn Sie CodeMeter License Server neu starten.

Rotierendes System für die Protokollierung

Derzeit ist ein solches System nicht implementiert.

 Jedes Mal wenn CodeMeter License Server gestartet wird, wird eine neue Protokollierungsdatei mit einem Zeitstempel angelegt und mit den entsprechenden Daten gefüllt.

12.8.3 Format der Protokollierungsdatei

Dem Format der Protokollierungsdatei liegt die folgende Logik zugrunde.

1. Jede Zeile der Protokollierungsdatei kann getrennt behandelt werden. Es existieren unterschiedliche [Eintragstypen](#)⁵⁰⁸.
2. Jede Zeile, die nicht dem hier beschriebenen Format entspricht, wird ignoriert.
Dies erlaubt Wibu-Systems die Ausgabeoptionen für künftige Versionen auszuweiten, ohne die bereits schon in Funktion befindliche Protokollierung zu beeinträchtigen.

 Wibu-Systems empfiehlt ebenso die Segmentierung der verschiedenen Argumente in einer Zeile (parsing) und dabei nicht bekannte Formate zu ignorieren.
Auch dies erlaubt Wibu-Systems die Ausgabeoptionen für künftige Versionen auszuweiten, ohne die bereits schon in Funktion befindliche Protokollierung zu beeinträchtigen.

12.8.3.1 Definitionen und Wertebereiche

Für die Protokollierungsdatei und die Eintragstypen werden die folgenden Definitionen und Wert (-bereiche) verwendet:

Definition	Werte (Bereich)
access id	Zeichenfolge (string)  Die <access id> wird vom Server vergeben und erweitert die <license id> um einen Index, der den Steckplatz (slot) beschreibt, d.h. <license id>-<slot id>.
application id	[0..4294967295]
application text	Zeichenfolge (string)
borrow id	Zeichenfolge (string)  Die <borrow id> wird abgeleitet aus <mask>-<serial number>-<firm code>-<enabling block index>. Dabei sind dies alle Werte des Ausleih-Clients.
enabling block index	[0..31]
expiration time	["never" UTC Timestamp]
feature map	[0..4294967295]
firm code	[0..4294967295]
license id	Zeichenfolge (string)  Die <license id> wird automatisch abgeleitet aus <mask>-<serial number>-<firm code>-<product item reference>, z.B. "2-1500002-100532-18". Die <license id> ist ein eindeutiger Bezeichner für einen Lizenzeintrag.
license quantity	[0..4294967295]
mask	[0..65535]
product code	[0..4294967295]
product item reference	[0..4294967295]
product item text	Zeichenfolge (string)
serial	[0..4294967295]
server	Zeichenfolge (string)
slot id	[0..4294967295]
timestamp	UTC Timestamp UTC Timestamp-Beispiel: "2012-12-24T08:32:59".
 Da Zeichenfolgen Anführungszeichen ("") enthalten können, aber Anführungszeichen auch die gesamte Zeichenfolge umgeben, werden Anführungszeichen, die Teil der Zeichenfolge sind über umgekehrte Schrägstriche (Backslash - \) markiert. Zum Beispiel, die Definition des application text <i>Das Beste von "Erika Mustermann"</i> . wird ausgegeben als: <pre>...AppText: "Das Beste von \"Erika Mustermann\"."</pre>	

12.8.4 Eintragstypen (Entry Types)

Die *CodeMeter® License Tracking*-Protokollierungsdatei kennt die folgenden aufgelisteten Eintragstypen (entry types).

12.8.4.1 List of Licenses-Eintrag

Eintrag	List of Licenses-Eintrag
Beschreibung	Einer Liste von License-Einträgen steht der List of Licenses-Eintrag vor. Dieser Eintrag zeigt an, dass in den folgenden Zeilen alle existierenden Lizenzen eines Servers aufgelistet werden. Eine zuvor geladene Liste von License-Einträgen wird ungültig.
Zeitpunkt des Schreibens	Der List of Licenses-Eintrag wird unmittelbar vor dem Schreiben der Liste von License-Einträgen geschrieben.
Syntax	<timestamp> ListOfLicenses

12.8.4.2 License-Eintrag

Eintrag	License-Eintrag
Beschreibung	Der License-Eintrag beschreibt eine existierende Lizenz.
Zeitpunkt des Schreibens	Alle License-Einträge werden in die Protokollierungsdatei geschrieben: <ul style="list-style-type: none"> • beim Starten von <i>CodeMeter License Server</i> • jede Mal wenn ein Eintrag sich ändert, z.B. bei An- und Ausstecken oder Fernprogrammierung.  In diesen Fällen sind den License-Einträgen der aktuellen Servers ein List of Licenses-Eintrag <small>508</small> vorangestellt.
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>"

Bevor im Falle einer Eintragsänderung alle License-Einträge erneut geschrieben werden, werden alle belegten Lizenzen durch einen Release-Eintrag freigegeben. Unmittelbar nach der Ausgabe der License-Einträge werden die zuvor freigegebenen Lizenzen wieder mit einem Access-Eintrag belegt.

Dies ist notwendig, da sich bei Umprogrammierungen license ids ändern können sowie beim Aussacken und folgenden automatischen Umbuchen. Außerdem kann sich durch automatisches Umbuchen nach Aussacken die access id ändern.

Lizenzen mit einer License Quantity (Lizenzanzahl) von einem Wert von 0 (nur lokale Lizenz) werden nicht gelistet.

 Das Verfallsdatum (Expiration Time) enthält das Minimum der Product Item Option Verfallsdatum (Expiration Time) und den Wert einer aktivierten Product Item Option Nutzungszeitraum (Usage Period). Wenn weder ein Verfallsdatum (Expiration Time) gesetzt ist, noch ein Nutzungszeitraum (Usage Period) existiert oder aktiviert ist, beträgt der Wert "never".

12.8.4.3 Access-Eintrag

Eintrag	Access-Eintrag
Beschreibung	Der Access-Eintrag zeigt, dass ein Benutzer eine Lizenz auf dem Server belegt.
Zeitpunkt des Schreibens	Der Access-Eintrag wird in dem Moment geschrieben, indem auf eine Lizenz zugegriffen wird.
Syntax	<pre><timestamp> Access Server:<server>, LicenseId:<license id>, AccessId:<access id>, Client:<computer name>, User:<user name>, AppId:<application id>, AppText:<application text>"</pre>
 Die Definitionen application id und application text leiten sich aus der CMCRECREDENTIAL-Struktur ab und nutzen die Parameter mulUserDefinedId und mszUserDefinedText.	

12.8.4.4 Release-Eintrag

Eintrag	Release-Eintrag
Beschreibung	Der Release-Eintrag zeigt, dass ein Benutzer eine zuvor auf dem Server belegte Lizenz freigegeben hat.
Zeitpunkt des Schreibens	Der Release-Eintrag wird in dem Moment geschrieben, indem eine Lizenz freigegeben wird.
Syntax	<pre><timestamp> Release Server:<server>, AccessId:<access id></pre>

12.8.4.5 Borrow Access-Eintrag

Eintrag	Borrow Access-Eintrag
Beschreibung	Der Borrow Access-Eintrag zeigt, dass ein Benutzer eine Lizenz von einem Server ausgeliehen hat.
Zeitpunkt des Schreibens	Der Borrow Access-Eintrag wird in dem Moment geschrieben, indem eine Lizenz ausgeliehen wird. Zusätzlich wird ein Borrow Access-Eintrag geschrieben, wenn <i>CodeMeter License Server</i> gestartet wird und bereits ausgeliehene Lizenz vorliegen.
Syntax	<pre><timestamp> Borrow Server:<server>, LicenseId:<license id>, BorrowId:<borrow id>, Client:<computer name>, User:<user name>, Expires:<expiration time>, BorrowSn:< mask>-<serial></pre>

12.8.4.6 Borrow Return-Eintrag

Eintrag	Borrow Return-Eintrag
Beschreibung	Der Borrow Return-Eintrag zeigt entweder, dass ein Benutzer eine zuvor von einem Server ausgeliehene Lizenz zurückgegeben hat oder, dass eine Ausleihdauer abgelaufen ist und die Lizenz automatisch zurückgegeben wurde.
Zeitpunkt des Schreibens	Der Borrow Return-Eintrag wird in dem Moment geschrieben, indem eine ausgeliehene Lizenz zurückgegeben wird.
Syntax	<pre><timestamp> Return Server:<server>, BorrowId:<borrow id></pre>

12.8.4.7 Denial-Eintrag

Eintrag	Denial-Eintrag
Beschreibung	<p>Der Denial-Eintrag zeigt, dass ein Benutzer eine Lizenzzugriff-Anfrage hat, die Lizenz aber nicht zugewiesen werden konnte, weil keine Lizenzen auf dem Server verfügbar sind.</p> <p>Der Eintrag zeigt nicht an, dass auf Lizenzen zugegriffen werden soll, die auf diesem Server nicht vorhanden sind.</p>
Zeitpunkt des Schreibens	Der Denial-Eintrag wird in dem Moment geschrieben, indem ein Lizenzzugriff fehlgeschlagen ist.
Syntax	<pre><timestamp> Denial Server:<server>, LicenseId:<license id>, Client:<computer name>, User:<user name>, AppId:<application id>, AppText:<application text></pre>

	Ein Denial-Eintrag wird nur mitprotokolliert, wenn der Fehler 212 auftritt (CMERROR_NO_MORE_LICENSES).
--	--

12.8.4.8 Administrative-Eintrag

Eintrag	Administrative-Eintrag
Beschreibung	Der Administrative-Eintrag zeigt ein Ereignis von <i>CodeMeter License Server</i> an.
Zeitpunkt des Schreibens	Der Administrative-Eintrag wird in dem Moment geschrieben, indem das beschriebene Ereignis stattfindet.
Syntax	<pre><timestamp> Admin Server:<server> CodeMeter_started <timestamp> Admin Server:<server> CodeMeter_stopped</pre>

	<p>Wenn <i>CodeMeter License Server</i> angehalten wird, werden automatisch alle Access-Einträge abgebrochen. Nur Borrow Access-Einträge bleiben gültig und werden beim nächsten Start von <i>CodeMeter License Server</i> wieder hergestellt.</p> <p>Üblicherweise werden Release-Einträge automatisch der Protokollierungsdatei hinzugefügt. In manchen Fällen kann diese u.U. nicht möglich sein, z.B. beim Absturz von <i>CodeMeter License Server</i>.</p>
---	---

12.9 HID-Unterstützung

CodeMeter® unterstützt ab Version 5.0 die Gerätekasse Human Interface Device (HID) des USB-Standards.

Eine Installation spezieller USB Host-Treiber ist nicht notwendig, da die Kommunikation über die USB HID-Klasse standardisiert ist und von Betriebssystemen Treiber zur Verfügung gestellt werden. Derzeit werden die Betriebssysteme Windows, Mac OS und Linux unterstützt.

Damit können sich *CmDongles* alternativ zur Massenspeicher-Anzeige (Mass Storage Device) auch als HID (Human Interface Device) am System anmelden und es wird kein Laufwerk angezeigt.

	HID ist derzeit verfügbar für alle <i>CmDongles</i> , 1001-02-xxx (ohne FlashDisk).
--	---

Voraussetzungen

- *CmContainer* mit der Kennung "2-xxxxxxx" (Samsung Chips)
- mindestens *CodeMeter®* Firmware 2.02
- mindestens *CodeMeter®* Runtime 5.0

Der Standard der USB-Kommunikation kann jederzeit in beide Richtungen zwischen Massenspeicher (Mass Storage Device) oder Human Interface Device (HID) umgestellt werden.

12.9.1 Umstellen: Massenspeicher zu HID

Zum Umstellen des Standard zur USB-Kommunikation vom Massenspeicher (Mass Storage Device) zu Human Interface Device (HID) gehen Sie wie folgt vor:

1. Einsehen des Status im *CodeMeter WebAdmin* Seite "**Inhalt | CmContainer**".

Ein Laufwerk ist zugeordnet und kein Flash-Speicher vorhanden.

The screenshot shows the 'CodeMeter WebAdmin' interface with the 'Inhalt' tab selected. In the 'CmContainer' section, the dropdown 'CmContainer:' is set to '2-2251132'. The 'Name:' field is empty. The 'CmContainer Typ:' field is set to 'CmStick/C 2.01'. The 'Erstes Laufwerk:' field is circled in red and contains the value 'E: (Kein Flash)'. Below it, the 'Status:' field has three radio button options: 'Gesperrt' (Locked), 'Aktiviert solange angeschlossen' (Activated while connected), and 'Aktiviert' (Activated), with 'Aktiviert' being selected. At the bottom, there are buttons for 'Aktualisieren' (Update) and 'Defragmentieren' (Defragment).

2. Aufrufen von [cmu](#)⁵⁰².

Für Windows rufen Sie *cmu* über den Systemmenü-Eintrag "**Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt**" auf. Für die Betriebssysteme Mac OS, Linux und Sun ist dieser Befehl im Suchpfad hinterlegt.

3. Eingeben der folgenden Kommandozeile:

```
cmu32 /s [Boxenmaske-Seriennummer] --set-config-disk HidCommunication
```

Der derzeitige Status wird in der folgenden Ausgabe der Kommandozeile angezeigt:

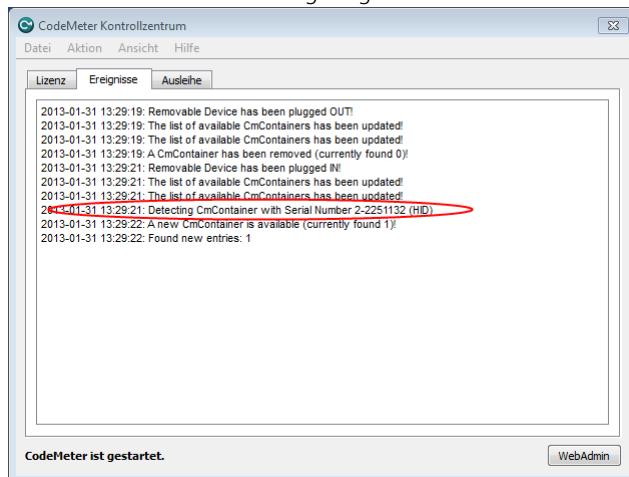
cmu32 - CodeMeter Universal Support Tool.

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
```

4. Abziehen und Wiederanstecken des *CmDongles*.
5. Protokollierung im *CodeMeter Kontrollzentrum*-Karteireiter "Ereignisse".
Der Wechsel auf HID wird angezeigt.



5. Überprüfen im *CodeMeter WebAdmin* Seite "Inhalt | CmContainer".
Kein Laufwerk ist zugeordnet.

The screenshot shows the 'CodeMeter WebAdmin' interface. The top navigation bar includes links for Home, Inhalt, Server, Einstellungen, Diagnose, Info, Hilfe, CmContainer, Lizizenzen, Benutzerdaten, and Datensicherung. The main content area displays the following information for a CmContainer:

- CmContainer: 2-2251132
- Name: <no name>
- CmContainer Typ: CmStick/C 2.01
- Erstes Laufwerk: Kein Laufwerk zugeordnet (HID) (This field is circled in red)
- Status:
 - Gesperrt
 - Aktiviert solange angeschlossen
 - Aktiviert
- System Zeit (PC): 2013-01-31 13:25:49
- System Zeit (CmContainer): 2013-01-31 13:25:49
- Zertifizierte Uhrzeit (CmContainer): 2012-11-26 15:37:36
- Freier Speicher: 94 % (367.696 Bytes)

Buttons for Aktualisieren and Defragmentieren are visible on the right.

12.9.2 Umstellen: HID zu Massenspeicher

Zum Umstellen des Standard zur USB-Kommunikation vom Human Interface Device (HID) zum Massenspeicher (Mass Storage Device) gehen Sie wie folgt vor:

1. Einsehen des Status im *CodeMeter WebAdmin* "Seite "Inhalt | CmContainer".
Ein Laufwerk ist nicht zugeordnet.

The screenshot shows the CodeMeter WebAdmin interface. At the top, there's a navigation bar with tabs: Home, Inhalt, Server, Einstellungen, Diagnose, Info, and Hilfe. Below the navigation bar, there are links for CmContainer, Lizizenzen, Benutzerdaten, and Datensicherung. The main area contains the following configuration details:

- CmContainer: 2-2251132
- Name: <no name>
- CmContainer Typ: CmStick/C 2.01
- Erstes Laufwerk: Kein Laufwerk zugeordnet (HID) (This field is circled in red)
- Status:
 - Gesperrt
 - Aktiviert solange angeschlossen
 - Aktiviert
- System Zeit (PC): 2013-01-31 13:25:49
- System Zeit (CmContainer): 2013-01-31 13:25:49
- Zertifizierte Uhrzeit (CmContainer): 2012-11-26 15:37:36
- Freier Speicher: 94 % (367.696 Bytes)

Buttons at the bottom right include 'Aktualisieren' and 'Defragmentieren'.

2. Aufrufen von `cmu`

Für Windows rufen Sie `cmu` über den Systemmenü-Eintrag "**Start | Alle Programme | CodeMeter | Tools | CodeMeter Command Prompt**" auf. Für die Betriebssysteme Mac OS, Linux und Sun ist dieser Befehl im Suchpfad hinterlegt.

3. Eingeben der folgenden Kommandozeile:

```
C:\Users\fs>cmu32 /s [Boxenmaske-Seriennummer] --set-config-disk MsdCommunication
```

Der derzeitige Status wird in der folgenden Ausgabe der Kommandozeile angezeigt:

```
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.
```

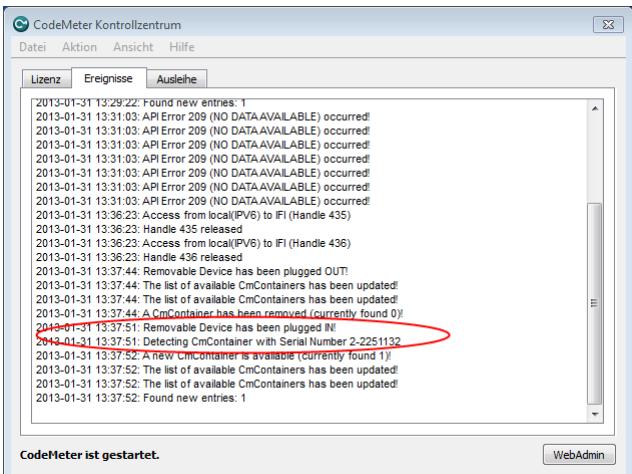
```
- CmStick/C with Serial Number 2-2251132 and version 2.01
Version: 2.01
Flash Size: no real flash available
Virtual Drive: No drive assigned (HID)
Communication: Human Interface Device (HID)
```

Please replug your CmDongle to apply the changes.

4. Abziehen und Wiederanstecken des *CmDongles*.

5. Protokollierung im *CodeMeter Kontrollzentrum*-Karteireiter "Ereignisse".

Der Wechsel auf MSD wird angezeigt.



5. Überprüfen im CodeMeter WebAdmin Seite "Inhalt | CmContainer". Ein Laufwerk ist zugeordnet und kein Flash-Speicher vorhanden.

CodeMeter WebAdmin

Inhalt | CmContainer

CmContainer: 2-2251132

Name: <no name>

CmContainer Typ: CmStick/C 2.01

Erstes Laufwerk: E: (Kein Flash)

Status:

- Gesperrt
- Aktiviert solange angeschlossen
- Aktiviert

System Zeit (PC): 2013-01-31 13:39:52

System Zeit (CmContainer): 2013-01-31 13:39:51

Zertifizierte Uhrzeit (CmContainer): 2012-11-26 15:37:36

Aktualisieren

Freier Speicher: 94 % (367.696 Bytes)

Defragmentieren

13 Glossar

Begriff	Erklärung
AxProtector	Grafische Anwendung zum automatischen Schutz von Software und digitaler Inhalte als Basissschutz ohne Eingriffe in den Quelltext. Dieser umfasst u.a. Überprüfungen der Lizenzenschaften zur Laufzeit der Anwendung, effektive Anti-Debugging-Maßnahmen, Modifikation von Ressourcen sowie Sperren des <i>CmContainers</i> bei Erkennung von Hack-Versuchen. Verfügbar für viele Projekttypen und als Kommandozeilen-Variante.
CmActLicense	Rein software-basierte Variante der Schutz- und Lizenzierungstechnologie <i>CodeMeter®</i> . Die Lizenzen sind an einen individuellen Rechner gebunden.
CmBoxPgm	Kommandozeilen-basierte Anwendung, mit der Sie Lizenzen und deren Bestandteile (Firm Item, Product Item und Product Item Options) in <i>CmContainer</i> anlegen, bearbeiten und löschen. Verwendung von Skripten und Batch-Dateien zur Programmierung von Abläufen in einem Vorgang für mehrere <i>CmContainer</i> .
CmContainer	Übergeordneter Begriff für den Lizenzträger beider <i>CodeMeter®</i> -Varianten: <i>CmDongle</i> im Fall des hardwarebasierten Lizenzierungssystems und <i>CmActLicense</i> für das softwarebasierte Lizenzierungssystem.
CmDongle	Hardware-basierte Variante der Schutz- und Lizenzierungstechnologie <i>CodeMeter®</i> . Verfügbar in verschiedenen Formfaktoren für verschiedene Schnittstellen.
CmDust	Das CodeMeter Enduser Support Tool liest wichtige System- und <i>CodeMeter®</i> -Einstellungen aus und erleichtert dem Support das Beheben von möglich auftretenden Fehlern.
CmFAS	siehe <i>CodeMeter Field Activation Service</i>
cmu	Kommandozeilen-basierte Alternative zur Ausführung vieler <i>CodeMeter Kontrollzentrum</i> -Funktionen (<i>CodeMeter Universal Support Tool</i>).
CodeMeter API Guide	Grafische Anwendung zur Erstellung von Quelltext-Fragmenten. Sie erstellen und testen API Funktionen mit allen dazugehörigen Parametern und notwendigen Strukturen für Ihre Programmiersprache. Unterstützte Programmiersprachen umfassen derzeit C, C++, C#, CB6, VB.NET, Delphi und Java.
CodeMeter Field Activation Service	siehe dateibasierte Fernprogrammierung
CodeMeter Kontrollzentrum	Das <i>CodeMeter Kontrollzentrum</i> erlaubt softwareseitig auf die <i>CodeMeter®</i> -Laufzeitumgebung zuzugreifen. Es zeigt Informationen über verbundene <i>CmContainer</i> an, und bietet einige Optionen zur Konfiguration von verbundenen <i>CmContainer</i> . Außerdem können über einen Assistenten Lizenzanforderungsdateien erzeugt und Lizenzaktualisierungsdateien eingespielt werden.
CodeMeter License Central	Ticketbasiertes System zum Erstellen, Verwalten und Ausliefern von Lizenzen für Software und digitale Inhalte. Verfügbar in einer <i>Desktop</i> und einer <i>Internet</i> Edition.
CodeMeter License Editor	Grafische Anwendung, mit der Sie Lizenzen und deren Bestandteile (Firm Item, Product Item und Product Item Options) in <i>CmDongles</i> anlegen, bearbeiten und löschen. Unterstützt neben der Programmierung von lokal mit dem PC verbundenen <i>CmCongles</i> auch die dateibasierte Fernprogrammierung (<i>CodeMeter Field Activation Service</i> , <i>CmFAS</i>). Geeignet zum Testen von Lizenzstrategien.
CodeMeter Lizenzserver	Laufzeitumgebung der Schutz- und Lizenzierungstechnologie <i>CodeMeter®</i> .
CodeMeter Start Center	Startanwendung zum Öffnen der <i>CodeMeter®</i> -Anwendungen und -Werkzeuge.

Begriff	Erklärung
<i>CodeMeter WebAdmin</i>	Grafische Anwendung zur Anzeige von Informationen über verbundene <i>CmContainer</i> und die darin enthaltenen Lizenzinstitute. Ermöglicht auch vielfältige Konfigurations- und Analyse-Optionen des <i>CodeMeter Lizenzservers</i> .
<i>CodeMeter®</i>	Wibu-Systems Technologie zum Schutz und zur Lizenzierung von Software und digitaler Inhalte.
Dateibasierte Fernprogrammierung	Um aus der Ferne die Aktualisierung eines <i>CmContainers</i> vornehmen zu können, benötigt man einige Informationen über den <i>CmContainer</i> , der umprogrammiert werden soll. Diese Informationen werden sicher in einer Kontext-Datei, der *.wibuCmRaC-Datei (Lizenzanforderungsdatei) abgespeichert und transportiert. Auf der Grundlage der Lizenzanforderungsdatei wird über die <i>CodeMeter®</i> -Programmierwerkzeuge eine Update-Datei (*.wibuCmRaU) erzeugt (Lizenzaktualisierung). Diese Datei wird anschließend sicher in den <i>CmContainer</i> transportiert. Zusätzlich wird beim Erstellen der *.wibuCmRaU-Datei automatisch auch eine *.wibuCmRaM-Datei erzeugt, mit der ein Abbild des <i>CmContainer</i> zur Verfügung steht, das der Lizenznehmer zum Zeitpunkt der Lizenzaktualisierung besitzt. Über den CmFAS-Assistenten im <i>CodeMeter Kontrollzentrum</i> wird der Lizenznehmer bei der Aktualisierung der Lizzen unterstützt.
Enabling	Verfahren über die Verwendung eines ZugriffsCodes gezielt, den gesamten <i>CmContainer</i> oder auch einzelne Firm Item-Ebenen oder Lizenzinstitute zu aktivieren oder zu deaktivieren.
Firm Code	Eine Zahl, die von Wibu-Systems jedem Lizenzgeber individuell und nur einmal vergeben wird und zur eindeutigen Identifizierung von Lizzen dient.
Firm Item	Logisch-hierarchische Eintragsebene im <i>CmContainer</i> . Auf der Firm Item-Ebene befinden sich Einträge, die für jeden einzelnen Lizenzgeber separat den jeweiligen Firm Code enthalten.
Firm Key	Geheimer Schlüssel der fast alle Ver- und Entschlüsselungsvorgänge von Lizzen, deren Authentifizierung sowie das Anlegen, Aktualisieren und Löschen von Lizenzinhalten auf der Ebene der Product Items beeinflusst. Der Firm Key wird initial in einer Firm Security Box ausgeliefert.
Firm Security Box	Sicherer Master-Dongle zum Programmieren von Lizzen in den <i>CmContainer</i> . Die FSB ist für jeden Lizenzgeber eindeutig.
FSB	siehe Firm Security Box
HIP	<i>High Level Programming-API</i> siehe <i>Programmier-API</i>
IFI	siehe Implicit Firm Item
Implicit Firm Item	Die Implicit Firm Item-Ebene im <i>CmContainer</i> verhält sich in der Regel genauso wie die üblichen Ebenen (Firm Items). Er weist lediglich einige Besonderheiten auf. Während sich alle anderen Ebenen durch das Vorhandensein eines exklusiven Firm Codes auszeichnen, der für jeden Lizenzgeber einzigartig ist, besitzt die Implicit Firm Item-Ebene den Firm Code 0. Dies bedeutet, dass jeder Besitzer eines <i>CmContainers</i> für die Implicit Firm Item-Ebene Lizenzgeber-Eigenschaften besitzt, und damit nicht nur lesend, sondern auch schreibend auf "seine" Ebene, das Implicit Firm Item zugreifen kann.
<i>IxProtector</i>	Individueller, erweiterter Schutz während der Anwendungsentwicklung. Es werden "echte" Quelltext-Fragmente über Schnittstellen (<i>Softwareschutz-API</i> , WUPI) und Sicherheitsmechanismen ver- und entschlüsselt. Geeignet zur Implementierung von modularem Softwareschutz.

Begriff	Erklärung
<i>Kern-API</i>	Mächtige Schnittstelle zur Kommunikation mit <i>CmContainern</i> zur Laufzeit von <i>CodeMeter</i> Lizenzserver. Alle anderen APIs und Schutzmechanismen (<i>AxProtector</i> , <i>IxProtector</i> , <i>Softwareschutz-API</i> WUPI) setzen letzten Endes auf <i>Kern-API</i> Funktionen auf. Daher eignet sich diese Schnittstelle zum Einsatz ergänzend zu den anderen Schutzmöglichkeiten (Ver- und Entschlüsselung von Daten, Personalisierung, Auslesen weiterer Daten).
Lizenzforschaltung	siehe dateibasierte Fernprogrammierung
Lizenzaktualisierungsdatei (*.wibuCmRaU)	Die Aktualisierungsdatei eines <i>CmContainers</i> , die nur für einen bestimmten <i>CmContainer</i> gültig ist und nur einmal eingespielt werden kann.
Lizenzanforderungsdatei (*.wibuCmRaC)	Die Kontextdatei eines <i>CmContainers</i> , der den Ist-Zustand von Lizenzeinträgen widerspiegelt und als Grundlage für eine Lizenzaktualisierung dient.
<i>SmartBind</i>	Bindungsart beim <i>CmActLicense</i> -Lizenzierungssystem, die die fortbestehende Gültigkeit von Lizenzen optimiert, wenn sich Hardware-Eigenschaften des Rechners ändern, an den die Lizenzen gebunden sind.
PIO	siehe Product Item Options
Product Code	Der Product Code ist eine Zahl, die Sie frei wählen können. Damit identifizieren Sie die Produkte, die Sie schützen und lizenziieren möchten.
Product Item Options (PIO)	Lizenzeinträge auf der Product Item-Ebene. Sie enthalten die Festlegungen der eigentlichen Eigenschaften einer Lizenz z.B. wieviele Lizenzen gleichzeitig in einem Netzwerk genutzt werden dürfen, wie lange eine Lizenz gültig ist, welche Funktionen benutzt und abgerechnet werden, usw. Außerdem stehen unterschiedliche Dateneinträge zur Verfügung, die zusätzliche Informationen enthalten und sich jeweils in ihren Zugriffsberechtigungen unterscheiden. Diese optionalen Eigenschaften lassen sich für jede Lizenz beliebig miteinander kombinieren und schaffen damit die Grundlage für die Abbildung von Lizenzmodellen und Lizenzstrategien.
Product Item	Logisch-hierarchische Eintragsebene im <i>CmContainer</i> unterhalb der Firm Item-Ebene. Auf der Product Item-Ebene befinden sich die einzelnen Lizenzeinträge, die Product Codes und weitere Product Item Options.
Programmier-API	Klassenorientierte Schnittstelle für den Programmierungs- und Organisationszugriff auf Lizenzeinträge in <i>CmContainer</i> . Erlaubt ein weitgehendes Customizing für die Integration in eigene Anwendungen.
<i>Softwareschutz-API</i>	Schnittstelle, um mit <i>IxProtector</i> geschützte Bereiche während der Laufzeit zu entschlüsseln, steht in Form von WUPI (WIBU Universal Protection Interface) zur Verfügung. Dieses schlanke, nur wenige, aber elementare Funktionen umfassende API ist universell für viele Programmiersprachen einsetzbar.
<i>Wibu Universal Protection Interface</i>	siehe <i>Softwareschutz-API</i>
WUPI	siehe <i>Wibu Universal Protection Interface</i>

Index

- * -

*.lif (license information file) 30

- . -

.NET

Obfuscierung 305, 306

- A -

Activation Time

Product Item Option 41

Aktivierungsdatum

siehe: Activation Time 41

Anschließen des CmDongle 433

ASM-Programmbibliothek

AxProtector Java 13

Lizenzbedingungen 13

Auslieferung 395

"stille" Installation der Runtime 399

Anpassungen vorkonfigurierte Installationspakete (Windows)
398

gezieltes Installieren von Features (Windows)

Kommandozeile 400

Kopieren der Laufzeitumgebung 404

Merge-Module (Windows) 397

Merge-Module Konfigurationsparameter (Windows) 400

Mobile Installation auf CmDongle 402

Nicht-Windows Betriebssysteme 396

Vorkonfigurierte Installationspakete (Windows) 397

Windows Betriebssysteme 396

AxProtector .NET

.NET Einstellungen 129

Ablaufdatum 119, 121

Abweichung CmContainer-/PC-Zeit 123

Abweichung CmContainer-/PC-Zeit nach hinten 123

Abweichung CmContainer-/PC-Zeit nach vorne 123

Aktivierungsdatum 121

Angepasste Fehlermeldungen 128

Auch zur Laufzeit 119

Automatisch Fällen generieren 125

Automatische Methodenverschlüsselung 126

Begrenzungszähler 119, 120

Dateiauswahl 112

Einfacher Debugger Check 125

Erstelldatum 115

Erweiterte Kommandozeilenoptionen 130

Erweiterte Optionen 130, 131

Exclusive Mode 117

Feature Code 114

Fehlermeldungen 127, 128

Firm Code 114

Herunterzählen um 119

Host-Anwendung beenden 123

Inline Meldungen (nur .NET) 128

Intervall 118

Intervall ohne Überprüfung 122

IxProtector aktivieren 130

IxProtector Bytes 137

IxProtector Methoden 137

IxProtector Name 136

IxProtectorAnsichten 136

Kein Strong Name 129

Laufzeiteinstellungen 118, 119

Laufzeiteinstellungen, erweitert 120, 121, 122, 123

Laufzeitüberprüfung 118

Laufzeitüberprüfungen 118

Linger Time 117

Lizenzbelegung 116, 117, 146

Lizenzierungssysteme 114, 115

Lizenzliste Beschreibung 132

Lizenzliste Erstelldatum 134

Lizenzliste Feature Code 134

Lizenzliste Firm Code 133

Lizenzliste Id 132

Lizenzliste Lizenzierungssysteme 133

Lizenzliste Minimale Firmware 134

Lizenzliste Minimale Treiberversion 134

Lizenzliste Product Code 133

Lizenzliste Subsystem 134

Lizenzliste WupiReadData 134

Lizenzliste WupiWriteData 135

Lizenzoption 117, 146

Lizenzprüfung bei Plug Out (CmDongle) 118

Lizenz-Zugriffssperre (Konfiguration) 125

Log-Datei erzeugen 131

Max. Alter Zeitzertifikat 122

Max. erlaubtes Ignorieren 118

Minimale Firmware 115

Minimale Treiberversion 114

Mobile Anwendung erzeugen 123

No User Limit 117

Normal User Limit 117

Obfuscierung 124

Optimierung 131

Probing 129

Product Code 114

Protokollierung 131

Prüfen der Verschlüsselungszeit 122

Quelldatei 112

Reflector defence 125

Schwellenwerte 119

AxProtector .NET	Lizenzoption 281
Sicherheitsoptionen, erweitert 125	Log-Datei erzeugen 282
Sperren des Lizenz-Zugriffs 125	Minimale Firmware 279
Standard-Fehlermeldungen 128	Minimale Treiberversion 278
Station Share 117, 146	No User Limit 281
Strong Name aus Container 129	Normal User Limit 281
Strong Name aus Datei 129	Product Code 278
Subsystem Lokal 116	Protokollierung 282
Subsystem Netzwerk 117	Quelldatei 276
Unterdrücke IxProtector-Fehlermeldungen 127	Station Share 281
User Message DLL 128	Subsystem Lokal 280
Verschlüsselungsalgorithmus 114	Subsystem Netzwerk 280
Wartungszeitraum 121	Verschlüsselungsalgorithmus 278
WibuKey Kompatibilitätsmodus 117	WibuKey Kompatibilitätsmodus 281
Zeitzertifikat setzen 122	Zieldatei 276
Zertifikat prüfen 122	Zusammenfassung Ausführen 291
Zieldatei 112	Zusammenfassung Fertigstellen 290
Zusammenfassung Ausführen 139	Zusammenfassung Zurück 290
Zusammenfassung Fertigstellen 138	
Zusammenfassung Zurück 138	
AxProtector Dateiverschlüsselung	AxProtector Java
Dateiauswahl 276	Ablaufdatum 177, 179
Dateiverschlüsselung Dateiendung 287	Abweichung CmContainer-/PC-Zeit 181
Dateiverschlüsselung Dateizugriffsmodus 288	Abweichung CmContainer-/PC-Zeit nach hinten 181
Dateiverschlüsselung Name 287	Abweichung CmContainer-/PC-Zeit nach vorne 181
Dateiverschlüsselung Player Überprüfung 287	Aktivierungsdatum 179
Dateiverschlüsselung Schreiben existierende Datei 289	Angepasste Fehlermeldungen 183
Dateiverschlüsselung Schreiben neue Datei 289	Auch zur Laufzeit 177
Erstelldatum 279	Aufruf von system.exit() 184
Erweiterte Kommandozeilenoptionen 282	Ausgabe aufteilen 185
Erweiterte Optionen 282	Begrenzungszähler 177, 178
Exclusive Mode 281	Blacklist 185
Feature Code 278	Callback Manipulationsüberprüfung 182
Firm Code 278	Class Name 183
Linger Time 281	Dateiauswahl 170, 171
Lizenzbelegung 280, 281	Einstellung main class 184
Lizenzierungssysteme 278, 279	Erstelldatum 173
Lizenzliste Beschreibung 283	Erweiterte Kommandozeilenoptionen 186
Lizenzliste Erstelldatum 285	Erweiterte Optionen 186
Lizenzliste Feature Code 284	Exclusive Mode 175
Lizenzliste Firm Code 284	Feature Code 172
Lizenzliste Id 283	Fehlermeldungen 183
Lizenzliste Ignoriere Linger Time 285	Firm Code 172
Lizenzliste Lizenzierungssysteme 284	Herunterzählen um 177
Lizenzliste Minimale Firmware 285	Intervall 176
Lizenzliste Minimale Treiberversion 285	Intervall ohne Überprüfung 180
Lizenzliste Product Code 284	IxProtector 316
Lizenzliste Subsystem 285	IxProtector aktivieren 186
Lizenzliste WupiReadData 285	Java Runtime Einstellung 184
Lizenzliste WupiWriteData 285	JVMPI Erkennung 182
	Klassen auswählen 185
	Klassen umbenennen 185

- AxProtector Java
- Laufzeiteinstellungen 176, 177
 - Laufzeiteinstellungen, erweitert 178, 179, 180, 181
 - Laufzeitüberprüfung 176
 - Linger Time 175
 - Lizenzbelegung 174, 175
 - Lizenzierungssysteme 172, 173
 - Lizenzoption 175
 - Log-Datei erzeugen 186
 - Max. Alter Zeitzertifikat 180
 - Max. erlaubtes Ignorieren 176
 - Methodenverschlüsselung 316
 - Minimale Firmware 173
 - Minimale Java Version 184
 - Minimale Treiberversion 172
 - No User Limit 175
 - Normal User Limit 175
 - Parameter main class 184
 - Product Code 172
 - Protokollierung 186
 - Prüfen der Verschlüsselungszeit 180
 - Quelldatei 170
 - Schwellenwerte 177
 - Sicherheitsoptionen 182
 - Standard-Fehlermeldungen 183
 - Station Share 175
 - Subsystem Lokal 174
 - Subsystem Netzwerk 174
 - User Message Class 183
 - Verschlüsselungsalgorithmus 172
 - VM Verifikation 182
 - Wartungszeitraum 179
 - Whitelist 185
 - WibuKey Kompatibilitätsmodus 175
 - Zeitzertifikat setzen 180
 - Zertifikat prüfen 180
 - Zieldatei 171
 - Zusammenfassung Ausführen 189
 - Zusammenfassung Fertigstellen 188
 - Zusammenfassung Zurück 188
- AxProtector Linux
- Ablaufdatum 198, 200
 - Abweichung CmContainer-/PC-Zeit 202
 - Abweichung CmContainer-/PC-Zeit nach hinten 202
 - Abweichung CmContainer-/PC-Zeit nach vorne 202
 - Aktivierungsdatum 200
 - Angepasste Fehlermeldungen 207
 - API statisch linken 205
 - Auch zur Laufzeit 198
 - Begrenzungszähler 198, 199
 - Dateiauswahl 191
 - Einfacher Debugger Check 203
 - Erstelltdatum 194
 - Erweiterte Kommandozeilenoptionen 208
 - Erweiterte Optionen 208
 - Erweiterter Debugger Check 203
 - Exclusive Mode 196
 - Feature Code 193
 - Fehlermeldungen 206, 207
 - Firm Code 192
 - Herunterzählen um 198
 - Intervall 197
 - Intervall ohne Überprüfung 201
 - IxProtector aktivieren 208
 - IxProtector Funktion Beschreibung 214
 - IxProtector Funktion Id 214
 - IxProtector Funktion Länge 214
 - IxProtector Funktion Lizenzliste 214
 - IxProtector Funktion Trap 214
 - Laufzeiteinstellungen 197, 198
 - Laufzeiteinstellungen, erweitert 199, 200, 201, 202
 - Laufzeitüberprüfung 197
 - Laufzeitüberprüfungen 198
 - Linger Time 196
 - Lizenzbelegung 195, 196
 - Lizenzierungssysteme 192, 193, 194
 - Lizenzliste Beschreibung 210
 - Lizenzliste Erstelltdatum 211
 - Lizenzliste Feature Code 211
 - Lizenzliste Firm Code 210
 - Lizenzliste Id 210
 - Lizenzliste Ignoriere Linger Time 211
 - Lizenzliste Lizenzierungssysteme 210
 - Lizenzliste Minimale Firmware 211
 - Lizenzliste Minimale Treiberversion 211
 - Lizenzliste Product Code 210
 - Lizenzliste Subsystem 211
 - Lizenzliste WupiReadData 212
 - Lizenzliste WupiWriteData 212
 - Lizenzoption 196
 - Lizenzprüfung bei Plug Out (CmDongle) 198
 - Lizenz-Zugriffssperre (Konfiguration) 204
 - Log-Datei erzeugen 208
 - Max. Alter Zeitzertifikat 201
 - Max. erlaubtes Ignorieren 197
 - Minimale Firmware 194
 - Minimale Treiberversion 193
 - No User Limit 196
 - Normal User Limit 196
 - Product Code 193

- AxProtector Linux
Protokollierung 208
Prüfen der Verschlüsselungszeit 201
Quelldatei 191
Schwellenwerte 198
Sicherheitsoptionen, erweitert 203, 204, 205
Sperren des Lizenz-Zugriffs 203
Standard-Fehlermeldungen 206
Station Share 196
Subsystem Lokal 195
Subsystem Netzwerk 195
User Message DLL 206
Verschlüsselungsalgorithmus 193
Virtuelle Maschinen 203
Virusprüfung hinzufügen 205
Wartungseitzraum 200
WibuKey Kompatibilitätsmodus 196
Zeitzertifikat setzen 201
Zertifikat prüfen 201
Zieldatei 191
Zu verschlüsselnder Code 205
Zusammenfassung Ausführen 217
Zusammenfassung Fertigstellen 216
Zusammenfassung Zurück 216
- AxProtector MAC OS
Ablaufdatum 148, 150
Abweichung CmContainer-/PC-Zeit 152
Abweichung CmContainer-/PC-Zeit nach hinten 152
Abweichung CmContainer-/PC-Zeit nach vorne 152
Aktivierungsdatum 150
Angepasste Fehlermeldungen 153
API statisch linken 156
Auch zur Laufzeit 148
Begrenzungszähler 148, 149
Dateiauswahl 141
Einfacher Debugger Check 154
Erstelldatum 144
Erweiterte Kommandozeilenoptionen 157
Erweiterte Optionen 157, 158
Erweiterter Debugger Check 154
Exclusive Mode 146
Feature Code 143
Fehlermeldungen 153
Firm Code 143
Herunterzählen um 148
Intervall 147
Intervall ohne Überprüfung 151
IxProtector aktivieren 157
IxProtector Funktion Beschreibung 164
IxProtector Funktion Id 164
IxProtector Funktion Länge 164
IxProtector Funktion Lizenzliste 164
IxProtector Funktion Trap 164
Laufzeiteinstellungen 147, 148
Laufzeiteinstellungen, erweitert 149, 150, 151, 152
Laufzeitüberprüfung 147
Laufzeitüberprüfungen 147
Linger Time 146
Lizenzbelegung 145, 146
Lizenzierungssysteme 143, 144
Lizenzliste Beschreibung 160
Lizenzliste Erstelldatum 161
Lizenzliste Feature Code 161
Lizenzliste Firm Code 160
Lizenzliste Id 159
Lizenzliste Ignoriere Linger Time 161
Lizenzliste Lizenzierungssysteme 160
Lizenzliste Minimale Firmware 161
Lizenzliste Minimale Treiberversion 161
Lizenzliste Product Code 160
Lizenzliste Subsystem 161
Lizenzliste WupiReadData 162
Lizenzliste WupiWriteData 162
Lizenzoption 146
Lizenzprüfung bei Plug Out (CmDongle) 147
Lizenz-Zugriffssperre (Konfiguration) 155
Log-Datei erzeugen 158
Max. Alter Zeitzertifikat 151
Max. erlaubtes Ignorieren 147
Minimale Firmware 144
Minimale Treiberversion 143
No User Limit 146
Normal User Limit 146
Product Code 143
Protokollierung 158
Prüfen der Verschlüsselungszeit 151
Quelldatei 141
Schwellenwerte 148
Sicherheitsoptionen, erweitert 154, 155, 156
Sperren des Lizenz-Zugriffs 154
Standard-Fehlermeldungen 153
Subsystem Lokal 145
Subsystem Netzwerk 145
Verschlüsselungsalgorithmus 143
Virtuelle Maschinen 154
Virusprüfung hinzufügen 156
Wartungseitzraum 150
WibuKey Kompatibilitätsmodus 146
Zeitzertifikat setzen 151
Zertifikat prüfen 151

- AxProtector MAC OS
 Zieldatei 141
 Zu verschlüsselnder Code 156
 Zusammenfassung Ausführen 167
 Zusammenfassung Fertigstellen 166
 Zusammenfassung Zurück 166
- AxProtector Windows
 Ablaufdatum 84, 86
 Abweichung CmContainer-/PC-Zeit 88
 Abweichung CmContainer-/PC-Zeit nach hinten 88
 Abweichung CmContainer-/PC-Zeit nach vorne 88
 Aktivierungsdatum 86
 Angepasste Fehlermeldungen 94
 API statisch linken 92
 Auch zur Laufzeit 84
 Begrenzungszähler 84, 85
 Dateiauswahl 77
 Dateiverschlüsselung aktivieren 96
 Dateiverschlüsselung Dateiendung 104
 Dateiverschlüsselung Dateizugriffsmodus 105
 Dateiverschlüsselung Name 104
 Dateiverschlüsselung Player Überprüfung 104
 Dateiverschlüsselung Schreiben existierende Datei 106
 Dateiverschlüsselung Schreiben neue Datei 106
 Dynamische Modifikation 89
 Dynamisches Laden der Wibu-Systems Bibliotheken 95
 Einfacher Debugger Check 90
 Erstelldatum 80
 Erweiterte Kommandozeilenoptionen 95
 Erweiterte Optionen 95, 96
 Erweiterte statische Modifikation 89
 Erweiterter Debugger Check 90
 Feature Code 79
 Fehlermeldungen 93, 94
 Firm Code 79
 Generischer Debugger Check 90
 Herunterzählen um 84
 Host-Anwendung beenden 88
 IDE Debugger Check 90
 Intervall 83
 Intervall ohne Überprüfung 87
 IxProtector aktivieren 95
 IxProtector Funktion Beschreibung 102
 IxProtector Funktion Id 102
 IxProtector Funktion Länge 102
 IxProtector Funktion Lizenzliste 102
 IxProtector Funktion Name 102
 IxProtector Funktion Trap 102
 Kernel Debugging Check 90
 Laufzeiteinstellungen 83, 84
- Laufzeiteinstellungen, erweitert 85, 86, 87, 88
 Laufzeitüberprüfung 83
 Laufzeitüberprüfungen 84
 Linger Time 82
 Lizenzbelegung 81, 82
 Lizenzierungssysteme 79, 80
 Lizenzliste Beschreibung 98
 Lizenzliste Erstelldatum 99
 Lizenzliste Feature Code 99
 Lizenzliste Firm Code 98
 Lizenzliste Id 97
 Lizenzliste Ignoriere Linger Time 99
 Lizenzliste Lizenzierungssysteme 98
 Lizenzliste Minimale Firmware 99
 Lizenzliste Minimale Treiberversion 99
 Lizenzliste Product Code 98
 Lizenzliste Subsystem 99
 Lizenzliste WupiReadData 100
 Lizenzliste WupiWriteData 100
 Lizenzoption 82
 Lizenzprüfung bei Plug Out (CmDongle) 84
 Lizenz-Zugriffssperre (Konfiguration) 90
 Log-Datei erzeugen 96
 Max. Alter Zeitzertifikat 87
 Max. erlaubtes Ignorieren 83
 Minimale Firmware 80
 Minimale Treiberversion 79
 Mobile Anwendung erzeugen 88
 No User Limit 82
 Normal User Limit 82
 Product Code 79
 Protokollierung 96
 Prüfen der Verschlüsselungszeit 87
 Quelldatei 77
 Ressourcenverschlüsselung 89
 Schwellenwerte 84
 Sicherheitsoptionen 89
 Sicherheitsoptionen, erweitert 90, 92
 Sperren des Lizenz-Zugriffs 90
 Standard-Fehlermeldungen 93
 Station Share 82
 Statische Modifikation 89
 Subsystem Lokal 81
 Subsystem Netzwerk 81
 Systemmenü erweitern 88
 Unterdrücke IxProtector-Fehlermeldungen 93
 User Message DLL 93
 Verschlüsselungsalgorithmus 79
 Virtuelle Maschinen 90
 Virusprüfung hinzufügen 92

- AxProtector Windows
Wartungszeitraum 86
WibuKey Kompatibilitätsmodus 82
Zeitzertifikat setzen 87
Zertifikat prüfen 87
Zieldatei 77
Zu verschlüsselnder Code 92
Zusammenfassung Ausführen 109
Zusammenfassung Fertigstellen 108
Zusammenfassung Zurück 108
- AxProtector-Kommandozeile
-! (Erzeugung wbc-Datei) 319
-# (Protokollierung) 319
-? -h (Optionen und Hilfe) 319
-@cmds.wbc (Parameter in ausführbarer Datei) 319
-A (Sicherheits-Code einfügen) 292
-A[AES] (Verschlüsselungsalgorithmus) 293
-ANF (Meldung wenn Assembly nicht gefunden, .NET) 312
-CAA (Sicherheitsoptionen) 296
-CACT (CmContainer Systemzeit) 297
-CAD (Datei-Zugriffsmodus) 297
-CAE (Plug-Out) 297
-CAG (Anti-Debugging .NET) 298, 299
-CAG (Anti-Debugging Java) 299, 300
-CAG (Anti-Debugging) 297, 298
-CAL (Bereiche der Verschlüsselung) 300
-CAM (Systemmenü) 300
-CAR (Runtime Check) 300
-CAS (Prozent Verschlüsselung) 300
-CAT (Zertifizierte und System-Zeit) 300, 301
-CAV (Virus-Überprüfung) 301
-CAZ (Verschlüsselungszeit) 301
-CCA (Zielplattform und -CCX) 301
-CCB (.reloc Section-Übersetzung) 301
-CCC (ActiveX / OCX Images) 302
-CCD (Shared Objects Optionen) 303
-CCE (PE wird nicht vergrößert) 302
-CCH (verhindert globales Hooking) 302
-CCI (Ladereihenfolge) 302
-CCK (Shared Objects entladen) 303
-CCM (Ladereihenfolge wibucrt.dll) 302
-CCQ (Lizenzen aufräumen Exit Process) 302
-CCR (Deaktivierung Umbenennung Sektionen) 302
-CCS (Lizenzen alle vom ersten verbundenen CmContainer) 302, 303
-CCX (Mixed Mode Assemblies) 303
-CDC (Dateinamenerweiterung Dateiverschlüsselung) 303
-CDH (Zugriff auf Lizenz Dateiverschlüsselung) 304
-CDK (Lizenzierungssystem Dateiverschlüsselung) 304
-CFx (Feature Code) 294
-CI (IxProtector) 305
-CID (IxProtector) 305
-CIH (WUPI und Hooking) 304
-CIN (Keine Fehlermeldungen IxProtector) 304
-CK (RID Schlüssel Cache, .NET) 305
-CMD(n) (Wiederverschlüsseln von Methoden, .NET) 306
-CML (Mindestgröße Methoden, .NET) 306
-CO (Obfuscierung) 305, 306
-CP (Aufräummechanismus Windows) 306
-CPA (Verschlüsselung Eigenschafts-Accessoren, .NET) 306
-D (Treiber-Version) 294
-EA (Activation Time Überprüfung) 307
-EC (MSIL Code Klassenkonstruktoren, .NET) 306
-EE (Expiration Time Überprüfung) 307
-EF (Verringerung Firm Access Counter) 307
-EM (Maintenance Period/Wartungszeitraum erforderlich)
307
-ET (erzwingt Certified Time Aktualisierung) 307, 308
-EU (Unit Counter/Begrenzungszähler Überprüfung) 308
-EXTRACT (Assembly Ausdruck, .NET) 319
-FW (Firmware-Version) 294
-FW (minimale Firmware bei Verschlüsselung) 309
-Fx (Firm Code) 293
-G (Bereiche von Verschlüsselung ausgenommen) 308
-I (Fehlerbehandlung Plugin DLLs) 310
-ja (Main Class Argumente Laufzeit Java) 313
-jb (Fehlerbehandlungsverfahren Java) 313
-jcl (alternativer ClassLoader Java) 313, 314
-jd (Min.- Max. Version Java) 314
-jff:[c|w] (externe Class-Dateien) 318
-jh (Verstecken und umbenennen von Klassen Java) 314
-jl (white- blacklist Klassen Java) 315
-jm (startende Main Class Java) 314
-jn (Klassen-Ladevorgang Java) 314
-jo (Ausgabeoptionen *.jar Java) 315
-jpc (Definieren zusätzlicher JAR-Dateien) 319
-jvm (Option Virtual Machine Java) 315
-jx (Beenden der Anwendung Java) 315, 316
-K (Lizenzierungssystem) 293
-L (Sprache Meldungstexte) 310
-M (Ausgabetexte für Meldungstexte) 310
-N (Netzwerkzugriff-Modus) 295, 296
-O (Name und Pfad der verschlüsselten Zieldatei) 313
-prio (Prozesspriorität festlegen) 313
-PROBING (Pfadangaben der Assembly nicht gefunden, .NET)
312
-Px Product Code) 293
-RD (Erstelldatum / Release Date) 294
-RID (Anzahl RID-Varianten) 308
-RIDI (Anzahl RID- u- Trap-Varianten IxProtector) 308
-S (Suchreihenfolge für Lizenzen) 294, 295

AxProtector-Kommandozeile
 -Silverlight (Verschlüsselung) 309
 -SNK (Strong Name Key, .NET) 312
 Syntax 292
 -TRAP (Hackerfallen, .NET) 313
 -U (Aufruf Meldungs-DLL) 312
 -UI (Inline-Meldungs-Assembly, .NET) 312
 -UM (Aufruf Meldungs-Assembly, .NET) 312
 -V (Verbose, ausführlicher Modus) 319
 -WC (Schwellenwert Certified Time) 309
 -WE (Schwellenwert Expiration Time) 309
 -WP (Schwellenwert Usage Period) 309
 -WU (Schwellenwert Unit Counter) 309
 -X (statisches Linken) 292
 -XAP (XAP Datei bei Silverlight) 309

- B -

Bauformen
 CodeMeter 26
 Human Interface Device (HID) 26

Begrenzungszähler

siehe: Unit Counter 44

Betriebssysteme

CodeMeter 31

Binding Extension 29

Bindungsschema 28

- C -

CmActLicense
 Aktivieren von Lizenzen 30, 433
 Binding Extension 29
 Bindungsschema 28
 None-Bind 29
 Protection Only-Lizenz 29
 SmartBind 28, 369
 SmartBind-Verhalten in VM 369
 Trial-Lizenz 29
 zusätzliche Aktivierungsoptionen 29

CmActLicense-Lizenz
 Programmieren (CmBoxPgm) 372

CmBoxPgm 355, 358
 Abfolge Ausgabe -sqd 382
 Activation Time -pat 361
 Activation Time, absolut -pata 361
 Activation Time, relativ -patr 362
 Aktualisiieren -cu 357
 Anfügen Enabling Block -eatt 379
 Anlegen RaC Datei -crac 381
 Anzeige CmActLicense Bindungsschemata -lfs 369
 Anzeige CmActLicense Installations ID -ldi 369
 Anzeige CmActLicense Lizenzdatei -ldf 369

Auflisten -l 357
 Ausführlicher Modus -v 382
 Backup Datei -b kp 381
 Box Index -qb 358
 Box Index Range -qnx 358
 Box Passwort -pwd 358
 CmActLicense Aktivierungscode -lac 368
 CmActLicense Aktivierungsdatei -laf 368
 CmActLicense beliebig oft einspielen VM -lopt:reimport 371
 CmActLicense Bindungswert-lbind 368
 CmActLicense in VM -lopt:vm 371
 CmActLicense License ID -lpid 371
 CmActLicense Lizenz-Informationsdatei (file) -lif 370
 CmActLicense Lizenz-Informationsdatei (phone) -lip 370
 CmActLicense Lizenzoptionen -lopt 371
 CmActLicense Name -lpn 371
 CmActLicense Ziel-Betriebssystem -los 371
 Customer Owned License Information -pcoli 362
 Dateibasierte Aktivierung CmActLicense 370
 Detach Enabling Block -edet 380
 Disable Time -edt 380
 Disable Time, absolut -edta 380
 Disable Time, relativ -edtr 381
 Enabling Access Code -eac 379
 Enabling -e 379
 Enabling Mode -em 381
 Enabling Text -et 381
 Expiration Time -pet 362
 Expiration Time, absolut -peta 363
 Expiration Time, relativ -petr 363
 Extended Protected Data -ped 362
 Feature Map -pfm 363
 Firm Access Counter -fac 359
 Firm Code -f 359
 Firm Item Text -ft 360
 Firm Key -fk 378
 Firm Precise Time, absolut -fpta 359
 Firm Precise Time, relativ -fptr 359
 Firm Update Counter -fuc 360
 FSB Entry -fsb 378
 Hidden Data -phd 363
 Hilfe aufrufen -? 382
 Hinzufügen / Aktualisieren -cau 357
 Hinzufügen -ca 357
 License Quantity -plq 364
 Linger Time -plt 364
 Lizenzausleihe Client -blc 377
 Lizenzausleihe Server -bls 376
 Logging -log 382
 Löschen -cd 357

CmBoxPgm 355, 358
Löschen wenn möglich -cdx 357
Maintenance Period -pmp 364
Maintenance Period, Datum -pmpd 365
Maintenance Period, Ganzzahl -ppmi 365
Minimal erforderliche Runtime (CmActLicense) 370
Network License Counter -pnwc 365
Product Code -p 361
Protected Data -ppd 365
Registry aufräumen -rcl 382
Rekursives Löschen -r 358
Remote Activation -ra 382
Secret Data -psd 366
Seriennummer -qs 358
Smart Bind 369
Telefonische Aktivierung CmActLicense 370
Text -pt 366
Unit Counter -puc 366
Unit Counter, absolut -puca 366
Unit Counter, relativ -pucr 367
Usage Period -pup 367
Usage Period, absolute -pupa 367
Usage Period, relative -pupr 367
User Data -pud 367
Validierungsmodus -val 383
WUPI Data -pwupidata 368

CmDongle
Erstes Anschließen 433

CmDust 500

CmFAS Assistant 455

CmFirm.wbc 18

CmLicense Editor 346
Arbeiten mit 351
Ausgabefenster 350
Baumansichtsfenster 349
Darstellungsfenster 350
Menüleiste 347
Oberfläche und Navigation 347
Remote Programming 351
Symbolleiste 348
WibuCmRaC 351
WibuCmRaM 351
WibuCmRaU 351

cmu
CodeMeter Universal Support Tool 502

CmWAN
AxProtector (Verschlüsselung) 429
CmBoxPgm (Programmierung) 426
CodeMeter API Guide (Lizenzzugriff) 426

CodeMeter WebAdmin (Konfiguration) 429
Profiling 429
Registry; server.ini (Konfiguration) 429

CodeMeter 32
Bauformen 26
Betriebssysteme 31
Installation 435
Konzept 35
Token-Funktionen 32
X.509 Zertifikate 32

CodeMeter API Guide 337
Aufbau und Navigation 338
Baumansichtsfenster 340
Blöcke Karteireiter 340
Funktionen Karteireiter 340
Handle-Anzeigefenster 340
Interaktiv-Bereich 341
Menüleiste 338
Quellcode-Bereich 341
WUPI Karteireiter 340

CodeMeter auf Embedded Systemen
Compact Driver 34

CodeMeter Beispieldatenanwendungen
CmCalculator 343
CmDemo 341
WupiCalculator 343

CodeMeter Compact Driver 34

CodeMeter Dienst
starten (Linux) 444
starten (Mac OS) 444
starten (Sun Solaris) 445
starten (Windows) 443
stoppen (Linux) 444
stoppen (Mac OS) 444
stoppen (Sun Solaris) 445
stoppen (Windows) 443
Verhalten bei Systemstart 433

CodeMeter Kern-API 333
Authentifizierungs-API 334
Enabling API 334
Fehlermanagement-API 335
Funktionsbereiche 334
Management-API 336
Programming-API 336
Remote-API 336
Verschlüsselungs-API 335
Zeitmanagement-API 337
Zugriffs-API 334

CodeMeter Kontrollzentrum 443

- CodeMeter Kontrollzentrum 443
 - Aktivierungsstatus 446
 - Ausleihe-Karteireiter 453
 - CmDongle registrieren 449
 - CodeMeterDienst starten 448
 - Ereignisse-Karteireiter 453
 - Firmware aktualisieren 450
 - Lizenz importieren 446
 - Lizenz-Karteireiter 449
 - Menüleiste 446
 - Protokollierung einschalten 447
 - Status und Öffnen 454
 - Struktur und Navigation 445
 - CodeMeter License Central 384
 - Admin-Interface 391
 - Architektur 386
 - Connectoren 387
 - Depot-Interface 390
 - Einsatzszenarios 391
 - Gateway 388
 - Prinzip 384
 - CodeMeter License Editor
 - Firm Code 352
 - PIO Aktivierungsdatum 353
 - PIO Feature Map 353
 - PIO License Quantity 353
 - PIO Text 353
 - PIO Unit Counter 353
 - PIO Usage Period 353
 - PIO Verfallsdatum 354
 - PIO Wartungszeitraum 354
 - Product Code 353
 - CodeMeter License Tracking 505
 - Access-Eintrag 509
 - Administrative-Eintrag 510
 - Borrow Access-Eintrag 509
 - Borrow Return-Eintrag 509
 - Denial-Eintrag 510
 - Format 507
 - Konfiguration 505
 - License-Eintrag 508
 - List of Licenses-Eintrag 508
 - Release-Eintrag 509
 - Voraussetzungen 505
 - CodeMeter Lizenzserver 68
 - Starte als Cm WANServer (WebAdmin) 474
 - Starte Netzwerk Server (WebAdmin) 474
 - CodeMeter Start Center 66
 - CodeMeter WebAdmin 464
 - Authentifizierung 480
 - Diagnose | Protokoll 496
 - Diagnose Protokollierung 496
 - Durchführung Datensicherung 497
 - Einstellungen | Datensicherung 481
 - Einstellungen | Lizenzausleihe 482
 - Einstellungen | Netzwerk 471
 - Einstellungen | Proxy 475
 - Einstellungen | Server 473
 - Einstellungen | WebAdmin 480
 - Einstellungen | Zeitserver 479
 - Einstellungen | Zugriffsenschutz 476
 - Fernzugriff 480
 - Firewall 466
 - Hilfe 499
 - Home 468
 - Info 499
 - Inhalt | Benutzerdaten 486
 - Inhalt | CmContainer 469
 - Inhalt | Datensicherung 497
 - Inhalt | Lizizenzen 483
 - LAN-Server 473
 - Lizenzanzeige auf CmContainer 483
 - Lizenzanzeige Benutzer (IFI) 486
 - Lizenzanzeige im Netz 487
 - Lizenzanzeige Netz (Benutzer) 490
 - Lizenzanzeige Netz (zusammengefasst) 488
 - Lizenzen freigeben 490
 - Netzwerk-Port 466
 - Profiling 477
 - Server | Benutzer 490
 - Server | Cluster 488
 - Server | Lizenzverfolgung 491
 - Server Lizenzverfolgung 491
 - Server Suchliste 471
 - Server-Zugriff 480
 - Starte CmWAN Server 474
 - Starte Netzwerk Server 474
 - Starten 467
 - WAN-Server 473
 - White und Blacklist 477
 - Zertifizierte Zeit aktualisieren 470
- CodeMeter Zeitserver
 - Systemzeit (CmContainer. PC) 417
 - Zertifizierte Zeit 417
- CodeMeter.ini-Datei 402
- CodeMeterCSSI 32
- CodeMeterDienst starten (Windows) 448
- Customer Owned License Information (COLI)
 - Product Item Option 43

- D -	Umrüsten auf Massenspeicher 513
Disable Time 408	HID (Human Interface Device) 26, 433, 510
- E -	Hidden Data Product Item Option 50
Enabling 406 Beispiel 412	Human Interface Device (HID) 26, 510
Disable Time 408	- I -
Enabling Access Code 408	IFI (Implicit Firm Item) 406
Enabling Block 408	Implicit Firm Item 35
Enabling Level 410	Implicit Firm Item (IFI) 406
Enabling Mode 409	Individueller Softwareschutz 320
Enabling Status 409	Installation 32/64-Bit Windows 436
Lookup 410	Linux Betriebssysteme 440
Pflichtkennzeichner 411	MAC OS Betriebssysteme 438
Required Flag 411	Sun Solaris Betriebssysteme 441
Simple PIN 408	IPv4, IPv6 466
Time PIN 408	IxProtector Modularer Softwareschutz 320
Erstelldatum siehe Release Date 47	IxProtector .NET
Erstelldatum / Release Date AxProtector-Kommandozeile 294	.NET Einstellungen 236 Aktivieren WupiReadData 237 Angepasste Fehlermeldungen 235 Dateiauswahl 233 Erweiterte Kommandozeilenoptionen 237 Erweiterte Optionen 237, 238 Fehlermeldungen 234, 235 Inline-Meldungen 235 IxProtector aktivieren 237 IxProtector Bytes 244 IxProtector Methoden 244 IxProtector Name 243 IxProtectorAnsichten 243 Kein Strong Name 236 Lizenzliste Beschreibung 239 Lizenzliste Erstelldatum 241 Lizenzliste Feature Code 240 Lizenzliste Firm Code 240 Lizenzliste Id 239 Lizenzliste Lizenzierungssysteme 240 Lizenzliste Minimale Firmware 241 Lizenzliste Minimale Treiberversion 241 Lizenzliste Product Code 240 Lizenzliste Subsystem 241 Lizenzliste WupiReadData 241 Lizenzliste WupiWriteData 241 Log-Datei erzeugen 237 Optimierung 238 Probing 236 Protokollierung 237
- F -	
Feature Code 45 AxProtector-Kommandozeile 294	
Feature Map Product Item Option 45 Versionsverwaltung 45	
Fernprogrammierung -rau 358	
FIO (Firm Item Options) 35	
Firm Code 35 AxProtector-Kommandozeile 293	
Firm Item 35	
Firm Item Options (FIO) 35	
Firm Item Text setzen 357	
Firm Key 35	
Firm Security Box (FSB) 37	
Firmware-Version AxProtector-Kommandozeile 294	
FSB (Firm Security Box) 37	
- H -	
HID cmu-Programmierung 504 Umstellen auf HID 511	

- IxProtector .NET**
- Quelldatei 233
 - Standard-Fehlermeldungen 234
 - Strong Name aus Container 236
 - Strong Name aus Datei 236
 - User Message DLL 234
 - Zieldatei 233
 - Zusammenfassung Ausführen 246
 - Zusammenfassung Fertigstellen 245
 - Zusammenfassung Zurück 245
- IxProtector Linux**
- Angepasste Fehlermeldungen 264
 - Dateiauswahl 262
 - Erweiterte Kommandozeilenoptionen 265
 - Erweiterte Optionen 265
 - Fehlermeldungen 263, 264
 - Inline-Meldungen (nur .NET) 264
 - IxProtector Funktion Beschreibung 271
 - IxProtector Funktion Id 271
 - IxProtector Funktion Länge 271
 - IxProtector Funktion Lizenzliste 271
 - IxProtector Funktion Name 271
 - IxProtector Funktion Trap 271
 - Lizenzliste Beschreibung 267
 - Lizenzliste Erstelldatum 268
 - Lizenzliste Feature Code 268
 - Lizenzliste Firm Code 267
 - Lizenzliste Id 266
 - Lizenzliste Ignoriere Linger Time 268
 - Lizenzliste Lizenzierungssysteme 267
 - Lizenzliste Minimale Firmware 268
 - Lizenzliste Minimale Treiberversion 268
 - Lizenzliste Product Code 267
 - Lizenzliste Subsystem 268
 - Lizenzliste WupiReadData 269
 - Lizenzliste WupiWriteData 269
 - Log-Datei erzeugen 265
 - Protokollierung 265
 - Quelldatei 262
 - Standard-Fehlermeldungen 263
 - Unterdrücke IxProtector-Fehlermeldungen 263
 - User Message DLL 263
 - Zieldatei 262
 - Zusammenfassung Ausführen 274
 - Zusammenfassung Fertigstellen 273
 - Zusammenfassung Zurück 273
- IxProtector Mac**
- Angepasste Fehlermeldungen 250
 - Dateiauswahl 248
 - Erweiterte Kommandozeilenoptionen 251
 - Erweiterte Optionen 251
 - Fehlermeldungen 249, 250
 - IxProtector Funktion Beschreibung 257
 - IxProtector Funktion Id 257
 - IxProtector Funktion Länge 257
 - IxProtector Funktion Lizenzliste 257
 - IxProtector Funktion Name 257
 - Lizenzliste Beschreibung 253
 - Lizenzliste Erstelldatum 254
 - Lizenzliste Feature Code 254
 - Lizenzliste Firm Code 253
 - Lizenzliste Id 252
 - Lizenzliste Ignoriere Linger Time 254
 - Lizenzliste Lizenzierungssysteme 253
 - Lizenzliste Minimale Firmware 254
 - Lizenzliste Minimale Treiberversion 254
 - Lizenzliste Product Code 253
 - Lizenzliste Subsystem 254
 - Lizenzliste WupiReadData 255
 - Lizenzliste WupiWriteData 255
 - Log-Datei erzeugen 251
 - Protokollierung 251
 - Quelldatei 248
 - Standard-Fehlermeldungen 249
 - User Message DLL 249
 - Zieldatei 248
 - Zusammenfassung Ausführen 260
 - Zusammenfassung Fertigstellen 259
 - Zusammenfassung Zurück 259
- IxProtector Mac OS**
- IxProtector Funktion Trap 257
- IxProtector Windows**
- Aktivieren WupiReadData 222
 - Angepasste Fehlermeldungen 221
 - Dateiauswahl 219
 - Dynamisches Laden der Wibu-Systems Bibliotheken 222
 - Erweiterte Kommandozeilenoptionen 222
 - Erweiterte Optionen 222, 223
 - Fehlermeldungen 220, 221
 - Inline-Meldungen (nur .NET) 221
 - IxProtector Funktion Beschreibung 228
 - IxProtector Funktion Id 228
 - IxProtector Funktion Länge 228
 - IxProtector Funktion Lizenzliste 228
 - IxProtector Funktion Name 228
 - IxProtector Funktion Trap 228
 - Lizenzliste Beschreibung 224
 - Lizenzliste Erstelldatum 225
 - Lizenzliste Feature Code 225
 - Lizenzliste Firm Code 224

IxProtector Windows
Lizenzliste Id 224
Lizenzliste Ignoriere Linger Time 225
Lizenzliste Lizenzierungssysteme 224
Lizenzliste Minimale Firmware 225
Lizenzliste Minimale Treiberversion 225
Lizenzliste Product Code 224
Lizenzliste Subsystem 225
Lizenzliste WupiReadData 226
Lizenzliste WupiWriteData 226
Log-Datei erzeugen 222
Protokollierung 223
Quelldatei 219
Standard-Fehlermeldungen 220
Unterdrücke IxProtector-Fehlermeldungen 220
User Message DLL 220
Zieldatei 219
Zusammenfassung Ausführen 231
Zusammenfassung Fertigstellen 230
Zusammenfassung Zurück 230

- J -

jQuery
CodeMeter WebAdmin 13
Lizenzbedingungen 13

- K -

Kommunikationsmodus
IPv4, IPv5 466
Plattform-spezifische Standards 466
Profiling 466
Shared Memory 466

- L -

License information file (*.lif) 30
License Quantity
Product Item Option 40
Linger Time 82, 117, 146, 175, 196, 281
Product Item Option 48
Lizenzaktualisierungsdatei
einspielen 461
Lizenzanforderungsdatei
Bestehende Lizenz erweitern 457
erzeugen 457
Lizenz eines neuen herstellers hinzufügen 459
Lizenzzahl
siehe: License Quantity 40
Lizenzausleihe (Beispiel) 377
Lizenzausleihe (CmBoxPgm) 376, 377
Lizenzen
*.WibuRac 455

*.WibuRaU 455
aktualisieren 455
CmFAS 455
einspielen 455
Lizenzaktualisierungsdatei 455
Lizenzanforderungsdatei 455

Lizenzmodel

Concurrent Lizenzen 54
Demo-Versionen 55
Downgrade Management 56
Einzelplatz-Lizenzen 54
Floating Lizenzen 54
Hot / Cold Standby 57
Lizenzausleihe 58
Miete, Leasing 56
Modulare Lizenzen 55
Named User Lizenzen 58
Netzwerk-Lizenzen 54
nutzungsabhängig 52
Overflow Lizenzen 57
Pay-per ... 56
Rechnergebundene Lizenzen 58
Standard 52
Versionsmanagement 56

Lizenzverfolgung 491

- M -

Maintenance Period
Product Item Option 47
Modularer Softwareschutz 320

- N -

Nachlaufzeit
siehe: Linger Time 48
Netzwerkzugriff-Modus
AxProtector-Kommandozeile 295, 296
None-Bind 29
Nutzungsdauer
siehe: Usage Period 43

- O -

Obfuscierung .NET 305, 306
On-Demand Decryption 58

- P -

PIO (Product Item Options) 38
Product Code 35
AxProtector-Kommandozeile 293
Product Item Option 39
Product Item 35
Product Item Option
Activation Time 41

- Product Item Option
 - Customer Owned License Information (COLI) 43
 - Expiration Time 42
 - Extended Protected Data 49
 - Feature Map 45
 - Hidden Data 50
 - License Quantity 40
 - Linger Time 48
 - Maintenance Period 47
 - Product Code 39
 - Protected Data 49
 - Secret Data 51
 - Text 40
 - Unit Counter 44
 - Usage Period 43
 - User Data 48
- Product Item Options (PIO) 35, 38
- Profiling 466
 - Ablageort für verschiedene Betriebssysteme 505
- Programmierbeispiele
 - Samples 18
- Programmierung von CmContainern
 - *.wbb-Datei 392
 - CmBoxPgm 355
 - CmLicense Editor 346
 - CodeMeter License Central 384
 - Kontext-Datei 392
 - LIF, License Information File 392
 - Modifizierte Kontext-Datei 392
 - Programmieren per Dateitransfer 392
 - Update-Datei 392
 - WibuCmRaC 392
 - WibuCmRaM 392
 - WibuCmRaU 392
- Protected Data
 - Product Item Option 49
- Protection Only-Lizenz
 - CmActLicense binding 29
 - Programmieren 375
- S -**
 - Schlüsselableitung
 - Black Key 59
 - Erstelldatum 59
 - Feature Map 59
 - Firm Code 59
 - Product Code 59
 - Secret Data
 - Product Item Option 51
 - Server Suchliste 471
- *.ini-Konfigurationsdatei 472
- Serversuchliste (CodeMeter WebAdmin) 471
- Shared Memory 466
- Sicherung des CmDongle 422
- SmartBind
 - CmActLicense 369
 - CmBoxPgm 369
- Sperren des CmContainers 421
- Suchreihenfolge für Lizenzen
 - AxProtector-Kommandozeile 294, 295
- Suchreihenfolge für Lizenzen (WAN)
 - AxProtector Kommandozeile 294, 295
- Support
 - Wibu-Systems 19
- Systemstart
 - CodeMeter Dienst 433
- T -**
 - Temporary Enabling 409
 - Text
 - Product Item Option 40
 - Token
 - CodeMeter 32
 - Trailing Validation Block
 - TVB 361
 - Treiber-Version (minim.)
 - AxProtector-Kommandozeile 294
 - Trial-Lizenz
 - CmActLicense Bindung 29
 - Programmieren 374
 - TVB
 - Trailing Validation Block 361
- U -**
 - Unit Counter
 - Product Item Option 44
 - Usage Period
 - Product Item Option 43
 - User Data
 - Product Item Option 48
- V -**
 - Verfallsdatum
 - siehe: Expiration Time 42
 - Verschlüsselung
 - asymmetrisch 64
 - symmetrisch 62
 - Verschlüsselung 58
 - direkte 62
 - indirekte 62

Verschlüsselung	58	WupiWriteDataInteger	325
Schlüsselableitung	59		
Verwendung eigener Schlüssel		- X -	
Hidden Data	415	X.509 Zertifikate	
Secret Data	415	CodeMeter	32
Vorkonfigurierte Installationspakete (Windows)		- Z -	
Merge-Module	397	Zertifizierte Zeit	417
Reduziertes Paket	397	aktualisieren	470
Volumfähiges Paket	397		
- W -			
WAN			
Infrastruktur	424		
WAN, Wide Area Network	423		
Wartungszeitraum	47		
Erstelldatum	47		
siehe: Maintenance Period	47		
wbb-Datei (CmActLicense)	433		
Wide Area Network, WAN	423		
WUPI			
Beispiel: WupiCalculator	326		
WupiAllocateLicense	322		
WupiCheckDebugger	322		
WupiCheckLicense	323		
WupiDecryptCode	322		
WupiEncryptCode	322		
WupiFreeLicense	322		
WUPI-Funktion			
WupiAllocateLicense	322		
WupiCheckDebugger	322		
WupiCheckLicense	323		
WupiDecreaseUnitCounter	323		
WupiDecryptCode	322		
WupiEncryptCode	322		
WupiFreeLicense	322		
WupiGetHandle	322		
WupiGetLastError	325, 326		
WupiQueryInfo	323		
WupiReadData	324		
WupiReadDataInteger	324, 325		
WupiWriteData	325		
WupiWriteDataInteger	325		
WupiGetHandle	322		
WupiGetLastError	325, 326		
WupiQueryInfo	323		
WupiReadData	324		
WupiReadDataInteger	324, 325		
WupiWriteData	325		