

CodeMeter コードメータ

ユーザズガイド Ver. 6.80

No. 5014

2019年6月1日



Windows, Office, Word, Excel, PowerPointは、米国Microsoft社の各国における商標もしくは登録商標です。Adobe, Adobe Acrobat, Adobe Readerは、Adobe Systems Incorporatedの登録商標です。また、本文中に登場する製品の名称は、すべて関係各社の登録商標または商標であることを明記して本文中の表記を省略させていただきます。

サンカーラ株式会社
www.suncarla.co.jp

目次

Chapter 1 はじめに	5
1-1. はじめに.....	6
1-2. コードメータの特長.....	6
1-3. コードメータ・アーキテクチャ.....	9
1-4. 暗号キー生成・データ暗号化のしくみ.....	9
1-5. 高度なセキュリティと信頼性を実現したハードウェア.....	10
Chapter 2 コードメータ開発ツールをインストールする	11
2-1. インストール作業について.....	12
2-2. CodeMeter SDK をインストールする.....	12
2-3. 貴社のライセンスファイル CmFirm.wbc をインポートする.....	15
Chapter 3 実行形式プログラムにプロテクトをかける	17
3-1. 実行形式プログラムにプロテクトをかける.....	18
3-2. sample.exe を暗号化する.....	19
3-3. CmStick (CmDongle) にコードを登録する.....	29
3-4. 動作を確認する.....	33
3-5. 使用回数 (ユニットカウンタ) を設定したプロテクトを行う.....	36
3-6. 使用有効期限 (Expiration Time) を設定したプロテクトを行う.....	40
3-7. アクティベーションタイム (使用開始日) を設定したプロテクトを行う.....	43
3-8. 使用期間 (Usage Period) を設定したプロテクトを行う.....	45
3-9. メンテナンス期間を設定したプロテクトを行う.....	49
3-10. プロテクトされたプログラムを起動する場合の注意点.....	53
Chapter 4 自動暗号化ツール AxProtector について	55
4-1. 自動暗号化ツール AxProtector について.....	56
4-2. 日本語モードにする.....	58
4-3. AxProtector のメニュー画面.....	59
4-4. Windows アプリケーション EXE または DLL (32bit/64bit).....	60
4-5. .NET アセンブリ.....	103
4-6. データファイルを暗号化する.....	136
4-7. コマンドラインでの使用方法.....	138
Chapter 5 IxProtector/WUPI について	143
5-1. IxProtector とは.....	144
5-2. WUPI ファンクションについて.....	145
5-3. WUPI ファンクション一覧.....	146
5-4. WUPI ファンクションの使い方.....	148
5-5. WUPI ファンクション詳細.....	152

Chapter 6	CodeMeter Core API について	159
6-1.	CodeMeter Core API.....	160
6-2.	サンプルプログラムについて	160
6-3.	CodeMeter API ガイドの使い方	161
6-4.	CodeMeter API ガイドの使用例.....	162
6-5.	CodeMeter Core API 一覧	165
Chapter 7	ネットワーク機能について	169
7-1.	ネットワークライセンス管理とは	170
7-2.	ネットワークカウンタの登録方法	171
7-3.	コードメータサーバーの起動方法	174
7-4.	ネットワーク対応型プロテクトの作成方法.....	176
Chapter 8	リモートアップデート機能について	179
8-1.	リモートアップデート機能とは.....	180
8-2.	リモートアップデート作業の流れ.....	181
8-3.	ライセンス要求ファイル (.WibuCmRaC) の作成 (ユーザー側)	182
8-4.	ライセンス更新ファイルの作成 (貴社側)	185
8-5.	CmDongle を更新する (ユーザー側)	192
Chapter 9	CodeMeter License Editor の使い方	195
9-1.	CodeMeter License Editor について	196
9-2.	ライセンスエディタの画面構成.....	196
9-3.	ライセンス要求ファイルからライセンス更新ファイルを作成する	201
9-4.	ライセンス修正ファイルからライセンス更新ファイルを作成する	204
9-5.	ファームアイテムの追加と修正.....	207
9-6.	プロダクトアイテムの追加と修正	209
Chapter 10	CmBoxPgm の使い方	217
10-1.	CmBoxPgm について.....	218
10-2.	コード登録の流れ.....	218
10-3.	ファームコード (Firm Code) を登録する	219
10-4.	プロダクトコード (Product Code) を登録する.....	220
10-5.	プロダクトコード (Product Code) を削除する.....	221
10-6.	使用有効期限 (Expiration Time) を登録する	222
10-7.	各パラメータの説明	223
Chapter 11	コードメータコントロールセンターの使い方	229
11-1.	コードメータコントロールセンターとは.....	230
11-2.	ストラクチャとナビゲーション	231
11-3.	メニューバー	232
11-4.	「ライセンス」.....	235
11-5.	「イベント」.....	238
11-6.	「ライセンス貸出」.....	239

11-7. ライセンス貸出・返却の方法.....	240
--------------------------	-----

Chapter 12 Web アドミン (WebAdmin) の使い方 247

12-1. コードメータ Web アドミン (WebAdmin) について.....	248
12-2. ダッシュボード.....	250
12-3. コンテナ.....	251
12-4. 構成.....	257
12-5. 情報.....	268

Chapter 13 Adobe PDF ファイルにプロテクトをかける 269

13-1. Adobe PDF ファイルにプロテクトをかける.....	270
13-2. 作業に必要なもの.....	271
13-3. コードメータ開発ツールをインストールする.....	271
13-4. SmartShelter PDF Author Ver18.04 をインストールする.....	272
13-5. sample.pdf を暗号化する.....	274
13-6. CmStick にコードを登録する.....	279
13-7. 動作を確認する.....	279
13-8. 複数のファームコード / プロダクトコードを入れて暗号化する.....	280
13-9. プレインテキストパスワードについて.....	282
13-10. 複数の PDF ファイルを一括して暗号化する.....	283
13-11. コマンドラインで暗号化する方法.....	291
13-12. 暗号化された PDF ファイルをユーザーに配布する場合.....	294
13-13. 起動時に保護モードを無効化する方法 (Adobe Reader DC/XI).....	298
13-13-1. Adobe Reader DC の「環境設定」を開く.....	299
13-13-2. 「起動時に保護モードを有効にする (M)」のチェックをはずす.....	300
13-13-3. Adobe Reader DC を一度終了させる。.....	302
13-13-4. Adobe Reader XI の「環境設定」を開く.....	303
13-13-5. 「起動時に保護モードを有効にする (M)」のチェックをはずす.....	304
13-13-6. Adobe Reader XI を一度終了させる。.....	306

Chapter 14 ユーザーに配布する場合 307

14-1. ユーザーに配布する場合.....	308
14-2. Windows アプリケーション / .NET アプリケーションを配布する.....	308
14-3. 暗号化された PDF ファイルを配布する (SmartShelterPDF).....	309
14-4. Mac OS X アプリケーションを配布する.....	310
14-5. Linux アプリケーションを配布する.....	310
14-6. Sun Solaris アプリケーションを配布する.....	310

Chapter 1

はじめに

- 1-1. はじめに
- 1-2. コードメータの特長
- 1-3. コードメータ・アーキテクチャ
- 1-4. 暗号キー生成・データ暗号化のしくみ
- 1-5. 高度なセキュリティと信頼性を実現したハードウェア

1-1. はじめに

このたびは、高機能セキュリティツール「コードメータ(CodeMeter)」をご購入いただき、誠に有難うございます。この「コードメータ(CodeMeter)」は、世界最高レベルのセキュリティ機能を持つ最も優れたセキュリティツールです。必ず、貴社のセキュリティニーズにお応えできるものと確信しております。

1-2. コードメータの特長

コードメータには、プログラムやコンテンツファイルの不正コピーやリバースエンジニアリング解析を防止するための機能が豊富に搭載されています。

1. 最強の自動暗号化ツール「AxProtector」を搭載

コードメータには、プログラムを強力に暗号化する「AxProtector」が搭載されています。Windows 32bit/64bit、.NETアセンブリ、Mac OS X、Javaアプリをプログラムのソースコードを変更せずに、自動的に暗号化できます。暗号化アルゴリズムAES 128bitと独自の暗号化技術を駆使し、ファイルを強力に暗号化します。

2. メモリー上の「オンデマンド復号」を実現

AxProtectorで暗号化されたプログラムはディスク上で暗号化されているだけでなく、メモリー上でも暗号化されています。必要な時に必要なモジュールだけを復号して実行するという「オンデマンド復号」を実現しました。コードがメモリー上でも暗号化されているため、メモリー解析によるハッキングに対して強力にブロックすることが可能になりました。

3. IxProtector/WUPI

コードメータとワイブキー、またコードメータActに共通使用できるユニバーサルなAPIファンクションWUPI(Wibu Universal Protection Interface)が登場。IxProtectorと組み合わせることにより、「オンデマンド復号」を実現しながらも、モジュール単独のプロテクトチェックやユニットカウンタによる"Pay per Use"を実現できます。

4. データファイルを強力に暗号化

Adobe PDFファイル(PDF)、Flash(SWF,FLV)などの動画ファイルを強力に暗号化する「SmartShelter」(スマートシェルター)機能やAxProtectorファイル暗号化機能を搭載。データファイル、コンテンツファイルの不正コピー防止、ライセンス管理に非常に効果的です。

5. 1個のキーに6,000種類の異なるコードを登録可能

1個のコードメータキー(CmDongle)の中に、異なるライセンスコード(ファームコード/プロダクトコード)を6,000個まで登録可能です。複数の異なるコンテンツのライセンス管理を1つのコードメータキー(CmDongle)で一元管理できます。

* 6,000個は最大値で、User DataやText Data等のセキュリティ情報を登録すると6,000個未満になります。

6. 貴社専用の CmFSB (CodeMeter Firm Security Box)

コードメータキー(CmDongle)にライセンスコードを登録するために必要なCmFSB(CodeMeter Firm Security Box)は貴社専用です。第三者が不正に貴社のライセンスコードを使ってコードメータキーを作成することができません。確実に、貴社のセキュリティを守ります。

7. 強力なアンチデバッグ機能

ハッカーによる解析を防ぐために、強力なアンチデバッグ機能が搭載されています。長年によるハッキング対策の経験から、考えられるアンチデバッグのノウハウをできるだけ多く搭載しております。

8. 豊富なセキュリティオプション

プログラム暗号時のオプション機能として、使用回数制限、使用有効期限、使用開始期日、ランタイムチェック機能、アンチデバッグ機能、ウイルスによる改ざんチェック機能、拡張メモリ、セキュリティデータ領域など、セキュリティニーズに応じた多数のセキュリティオプションを用意しています。

9. 高機能なセキュリティ API を多数用意

WUPI (Wibu Universal Protection Interface)とは別に、ソースコードの中に直接組み込んでプロテクトチェックを行うコードメータ専用のコアAPIファンクションを多数用意しています。これらのコアAPIは、Windows/Linux/Mac OSに共通なクロスプラットフォームAPIで、きめの細かいプロテクトチェックを行うことができます。また、コアAPIは、WUPI (Wibu Universal Interface)と連携して使用することが可能です。

10. 充実したネットワーク機能

ネットワーク上のサーバーにコードメータを1個装着することで、1 - 65,536台までの範囲でネットワークライセンス数(フローティングライセンス数)を制限することができます。ネットワークライセンス数は、コードメータCmDongleの中に登録します。ローカルPCに1つ1つ装着する必要がないので、プロテクトコストを大幅に削減することが可能になります。

11. ライセンス貸出機能

さらに、ライセンス貸出機能も追加されました。ネットワークから外してPCを持ち出す際、オフライン用CmDongleにライセンスを貸出すことで、オフラインの状態でのアプリケーションを使用することが可能になります。コードメータサーバー側CmDongleのライセンス数はその分減りますので、全体としてのライセンス数は変わりません。また、貸出されたライセンスをサーバー側CmDongleに返却することでライセンス数は復元します。

12. リモートアップデート機能

メール添付による更新ファイル操作により、遠隔地にいるユーザー先のコードメータキーの内容を更新することができます。更新のために、コードメータキー(CmDongle)を送ったり、送り返す必要がないため、商品コード(ファームコード/プロダクトコード)の追加更新、使用期限の更新、使用回数の更新などがスピーディに行えます。

13. CmBOX タイマー機能

コードメータキー(CmDongle)の中に、時刻を確実に刻むCmBOXタイマー機能を内蔵しています。このタイマー機能を使って、アプリケーションの使用有効期限(Expiration Time)や使用開始期日(Activation Time)を確実にコントロールできます。ソフトウェアやコンテンツのレンタル販売や評価用、SaaS/ASP事業に効果的です。また、使用有効期限の更新は、オフライン/オンラインによるアップデート機能を使ってタイムリーに対応できます。

14. RoHS 指令対応済み

コードメータは、RoHS指令で規制されている鉛、水銀、カドミウム、六価クロム、ポリ臭化ビフェニール、ポリ臭化ジフェニルエーテルの使用基準を満たしているWEEE/RoHS指令適合製品です。

15. ISO9001:2008 認証取得

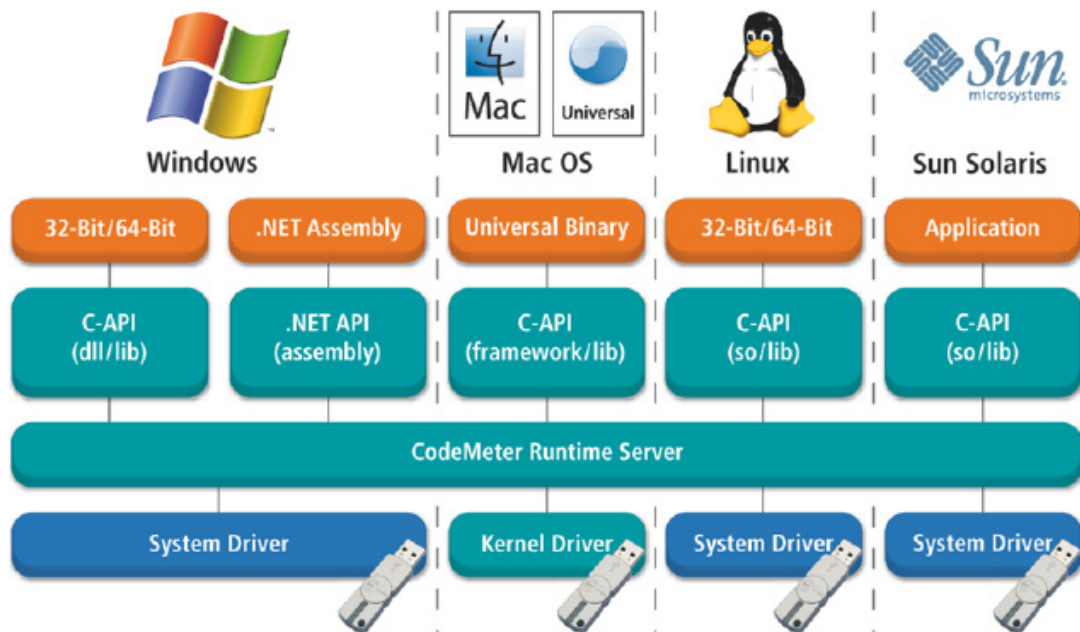
コードメータは、ISO9001:2008認証されたWIBU-SYSTEMS社(ドイツ)で開発・製造されており、高い品質と信頼性を実現した商品です。

16. その他、多くの高度なセキュリティを実現

モバイルアプリケーション機能を使うことで、PCにランタイムキットをインストールせずにコードメータで暗号化されたプログラムを使用できるモバイルアプリケーション機能を実現。ソフトウェアのインストールが禁止されているユーザー権限のPC上でも、コードメータを使用することが可能になりました。また、その他、多くのセキュリティ機能が用意されています。

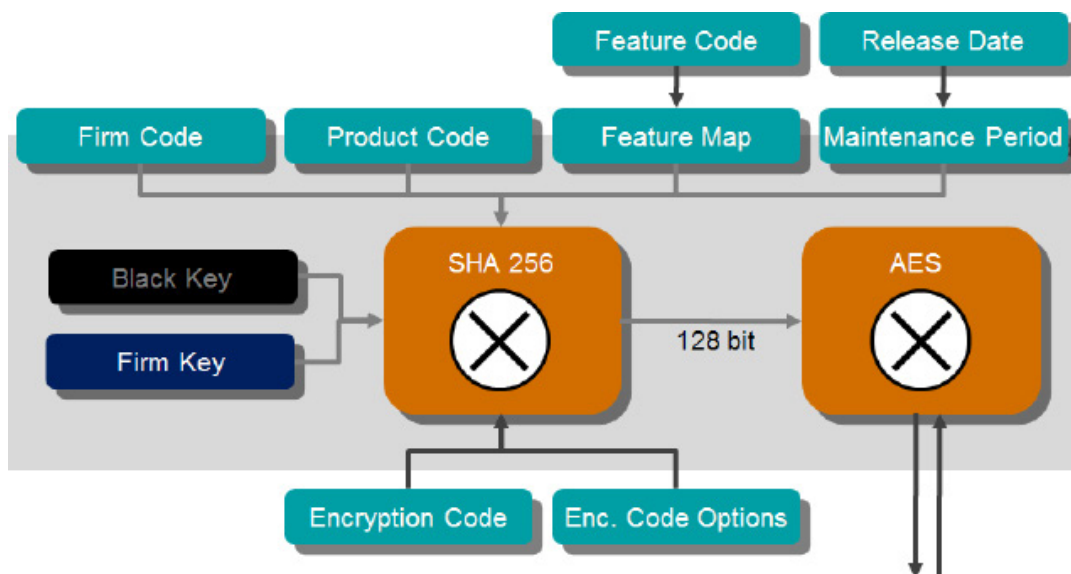


1-3. コードメータ・アーキテクチャ



1-4. 暗号キー生成・データ暗号化のしくみ

暗号化に使用する暗号キーは、CmDongle内部にあらかじめ組み込まれたBlack Key、Firm keyを始め、Firm Code、Product Code、Feature Code/Feature Map、Release Date/Maintenance Period、Encryption Code、Encryption Code Optionsを要素にして、ハッシュ関数SHA 256で生成され、その暗号キーをもとにデータ（プログラム）を暗号アルゴリズムAES 128bitで暗号化します。



1-5. 高度なセキュリティと信頼性を実現したハードウェア

高いセキュリティ性能と安定した品質のスマートカードチップを搭載し、従来にない高度なセキュリティを提供します。





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

Wibu-Systems is member of:

	PCMCIA since 1993
	USB Implementers Forum since 1997
	SD Card Association since 2007
	SIIA Software & Information Industry Association since 1998
	Bitkom, German Association of Information Technology, Telecommunications, and New Media since 2003
	VDMA, German Engineering Federation since 2008

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

	OEM Hardware Solutions	Microsoft Gold Certified Partner
		Microsoft Embedded Partner

Chapter 2

コードメータ開発ツールをインストールする

- 2-1. インストール作業について
- 2-2. CodeMeter SDK をインストールする
- 2-3. 貴社のライセンスファイル CmFirm.wbc をインポートする

2-1. インストール作業について

コードメータ開発ツールのインストール作業は以下の2ステップを行ないます。

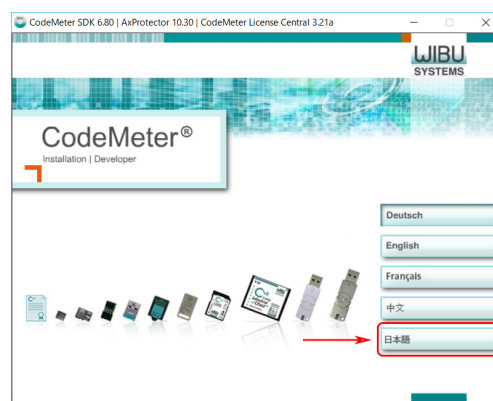
1. CodeMeter SDKをインストールする
2. 貴社のライセンスファイルCmFirm.wbcをインポートする

2-2. CodeMeter SDK をインストールする

①コードメータ DVD を起動する

コードメータDVDをDVDドライブに挿入すると右の画面が表示されます。表示されない場合はコードメータDVDのルートにある"Start.exe"をダブルクリックして起動します。

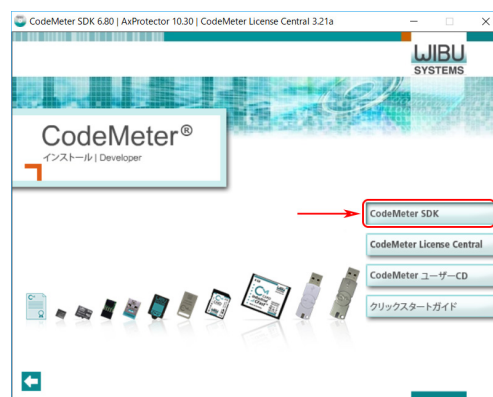
選択する言語が表示されますので、"English"をクリックします。



② CodeMeter SDK をクリックする

"CodeMeter SDK"をクリックします。

あとは、メッセージに従い、コードメータ開発キットをインストールします。



③ CodeMeter Development Kit v6.80 をインストールする

「CodeMeter Development Kit v6.80 のセットアップ」画面が表示されますので、「次へ」をクリックします。



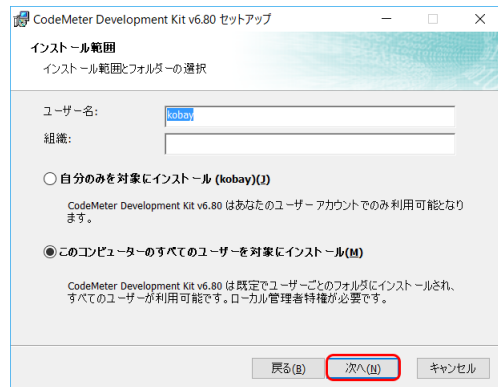
④ライセンス契約書

ライセンス契約書が表示されます。ライセンス契約書に同意のうえ、先に進みます。



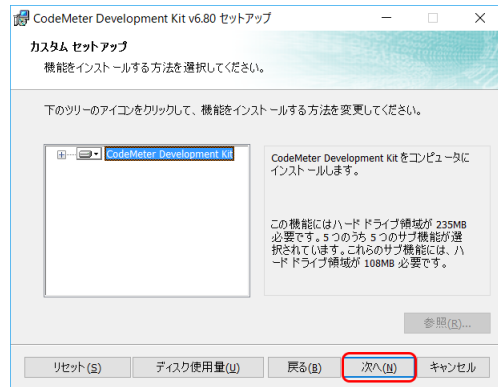
⑤ユーザー情報を入力する

ユーザー名などを入力します。



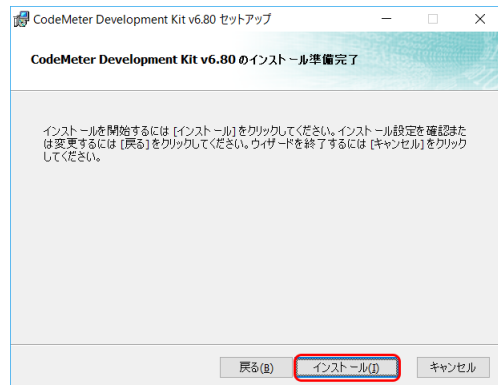
⑥インストールする内容を選択する

インストールする内容を選択します。ここでは、デフォルトのまま進めてください。



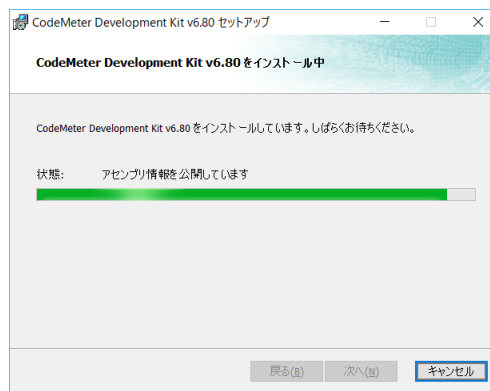
⑦インストールを開始する

インストールを開始する準備ができましたら、「インストール(I)」ボタンをクリックし、インストールを開始します。



⑧インストール作業が行われる

インストール作業が開始されます。インストール作業が開始するまで数秒待たされることがありますが、そのままお待ちください。



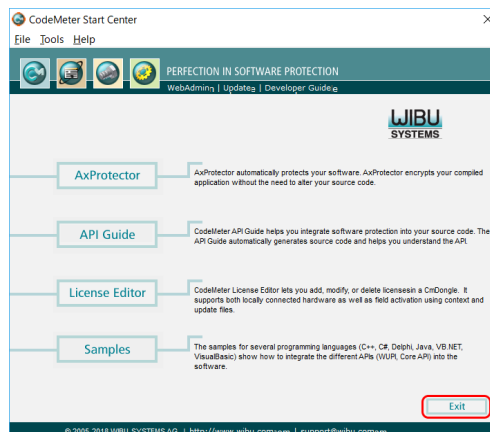
⑨正常にインストールされた

インストールが正常に終了すると、右の画面が表示されます。「完了」ボタンをクリックして画面を閉じます。



⑩コードメータ スタートセンター画面を閉じる

インストール作業の途中で、右のコードメータ スタートセンター画面が開きますが、ここでは使用しませんので、右下の"Exit"をクリックして画面を閉じます。



⑪最後に「閉じるボタン」をクリックする

閉じるボタン×をクリックして、コードメータ開発キットのインストール画面を終了します。

次に、貴社のライセンスファイルCmFirm.wbcをインポートします。次章の「2-3. 貴社のライセンスファイルCmFirm.wbcをインポートする」にお進みください。

評価版の場合は、CodeMeter SDKをインストールした時点で評価版CmFirm.wbcがインポートされるため、次章の「2-3. 貴社のライセンスファイルCmFirm.wbcをインポートする」作業は不要です。評価版の場合は、これでインストール作業が終了です。プログラム等の暗号化作業が可能です。

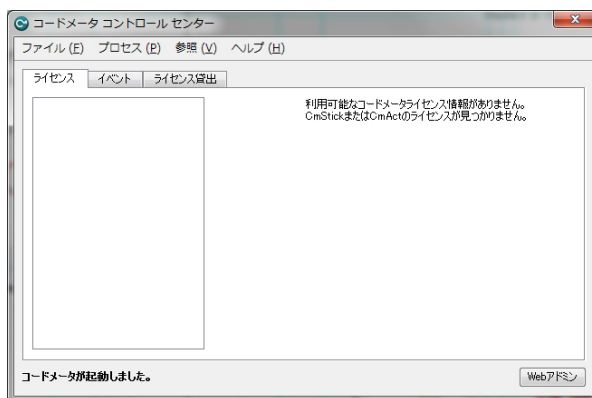
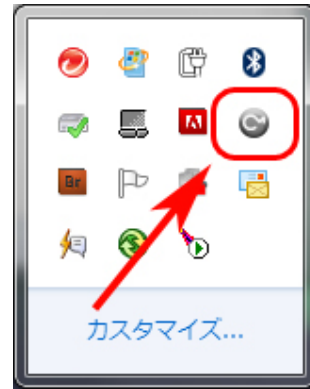


2-3. 貴社のライセンスファイル CmFirm.wbc をインポートする

コードメータFSB(CmFSB)をPCのUSBポートに装着し、CmFSBフラッシュ領域のルートディレクトリに存在する貴社のライセンスファイル"CmFirm.wbc"をインポートします。評価版の場合は、CodeMeter SDKをインストールした時点で、評価版用CmFirm.wbcがすでにインポートされていますのでこの作業は不要です。

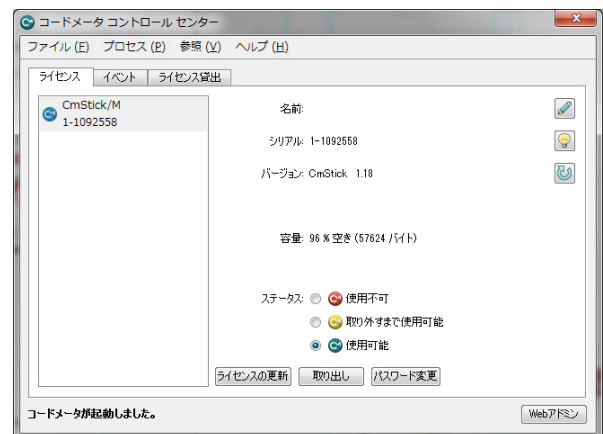
① コードメータコントロールセンターを開く

コードメータコントロールセンターを開くには、PC画面右下のタスクバーにあるコードメータアイコンをクリックします。タスクバーにアイコンが表示されていない場合は、[すべてのプログラム]-[CodeMeter]-[CodeMeter Control Center]をクリックします。



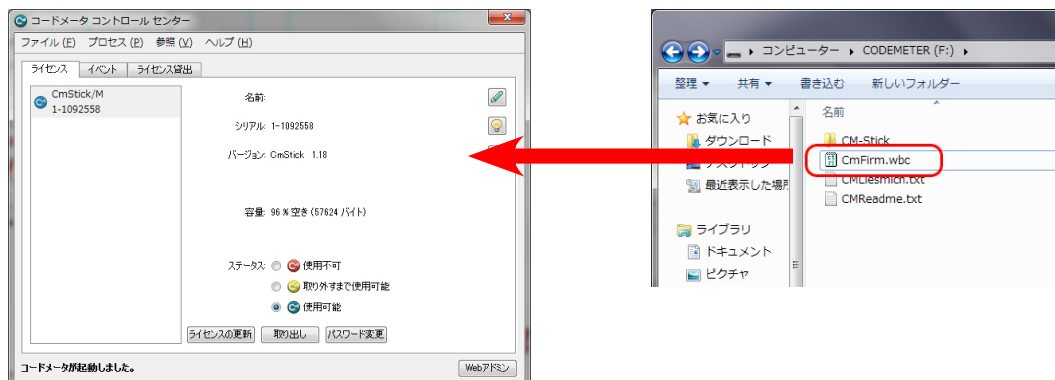
② コードメータ FSB を PC に装着する

コードメータFSB(CmFSB)をPCに装着します。しばらくすると、コードメータFSB(CmFSB)の情報が表示されます。



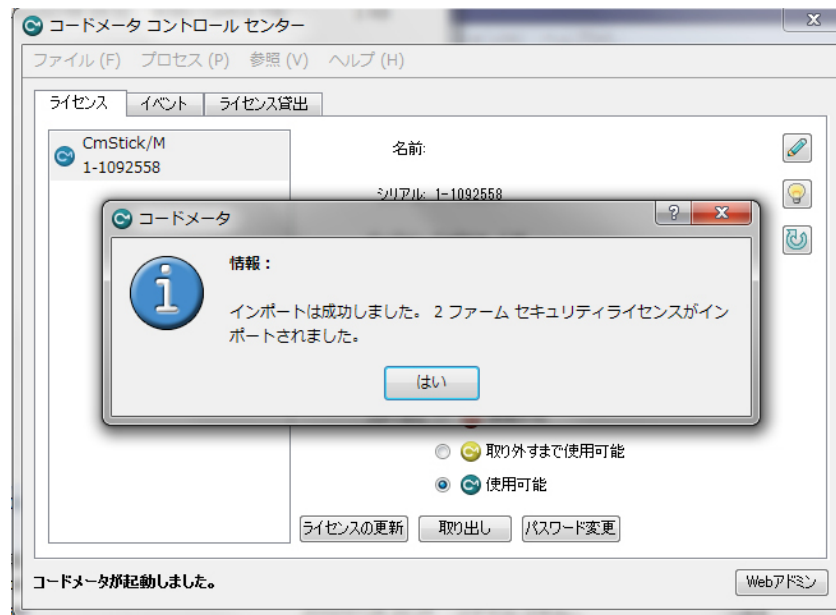
③ CmFirm.wbc をドラッグ&ドロップする

コードメータFSB(CmFSB)のルートディレクトリにあるCmFirm.wbcをコードメータコントロールセンターにドラッグ&ドロップします。



④ インポートに成功

インポートに成功すると、以下のメッセージが表示されます。



これで、コードメータ開発ツールの設定は完了です。プログラム等の暗号化作業が可能です。

Chapter 3

実行形式プログラムにプロテクトをかける

- 3-1. 実行形式プログラムにプロテクトをかける
- 3-2. sample.exe を暗号化する
- 3-3. CmStick にコードを登録する
- 3-4. 動作を確認する
- 3-5. 使用回数 (ユニットカウンタ) を設定したプロテクトを行う
- 3-6. 使用有効期限 (Expiration Time) を設定したプロテクトを行う
- 3-7. アクティベーションタイム (使用開始日) を設定したプロテクトを行う
- 3-8. 使用期間 (Usage Period) を設定したプロテクトを行う
- 3-9. メンテナンス期間を設定したプロテクトを行う
- 3-10. プロテクトされたプログラムを起動する場合の注意点

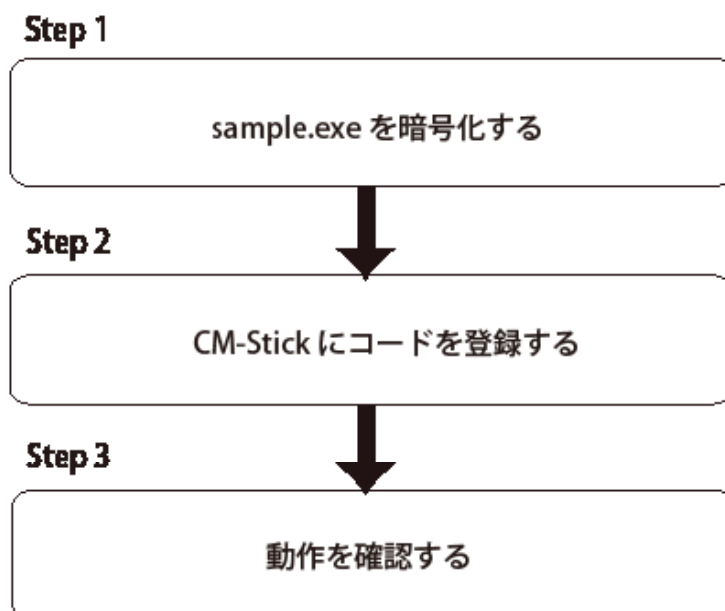
3-1. 実行形式プログラムにプロテクトをかける

コードメータの自動暗号化ツール「AxProtector」を使うと、実行形式プログラムに対して、ソースコードを変更することなく強力に暗号化を行うことができます。また、メモリー上で展開されるコードを常に暗号化し、必要な時に必要なモジュールを実行する「メモリー上のオンデマンド復号機能」を使用することができます。

この章では、コードメータDVDの中にあるsample.exeを使って、自動暗号化ツール「AxProtector」の基本的な使用方法をご説明いたします。sample.exeは、コードメータDVDのTools¥AxProtectorフォルダの中に格納されています。PCのハードディスク等にコピーしてお使いください。sample.exeは弊社サイトからもダウンロードできます。

<http://www.suncarla.co.jp/codemeter/manual/v680/tool.zip>

作業の流れとして、以下のようになります。



3-2. sample.exe を暗号化する

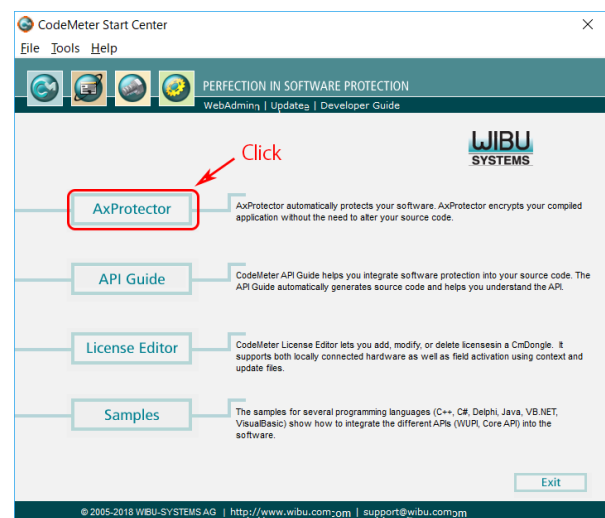
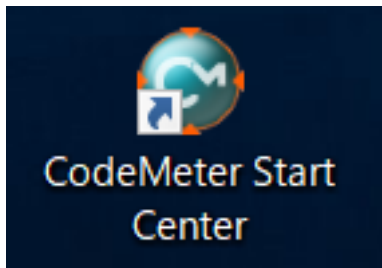
sample.exeを暗号化(プロテクト処理)します。プログラムを暗号化するには、自動暗号化ツール「AxProtector」を使用します。この「AxProtector」は、プログラムファイル全体を強力に暗号化し、デバッガからの解析を非常に困難にします。ソースコードを変更する必要がないのでとても便利です。

コードメータDVDのToolsフォルダからsample.exeを任意のフォルダにコピーします。ここでは、testフォルダを作成し、そこにコピーします。

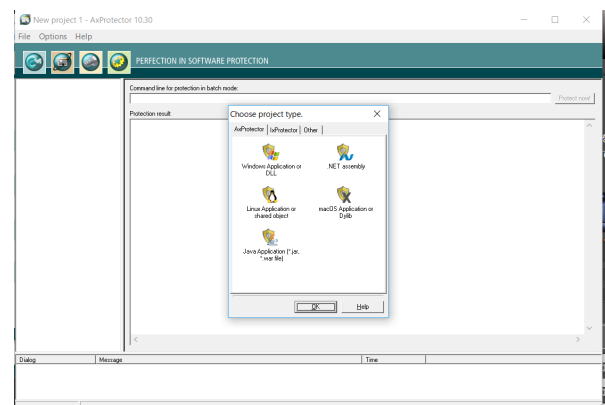
まず、コードメータ FSB(CmFSB) を装着する

AxProtectorを使ってファイルを暗号化するには、貴社のコードメータFSB (CmFSB)が必要です。評価版の場合は、評価用コードメータFSB(CmFSB10)が必要です。

まず、コードメータFSB(CmFSB)をPCのUSBポートに装着し、デスクトップ上の"CodeMeter Start Center"アイコンをダブルクリックすると、"CodeMeter Start Center"(右図)が起動します。



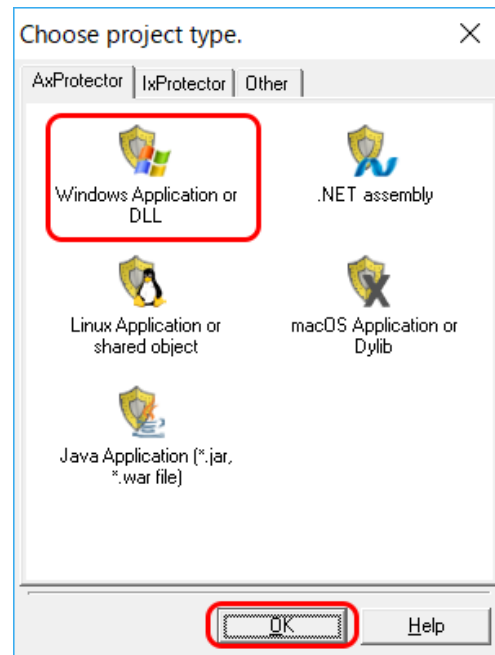
"AxProtector"ボタンをクリックするとAxProtectorが起動します。



[NOTE]

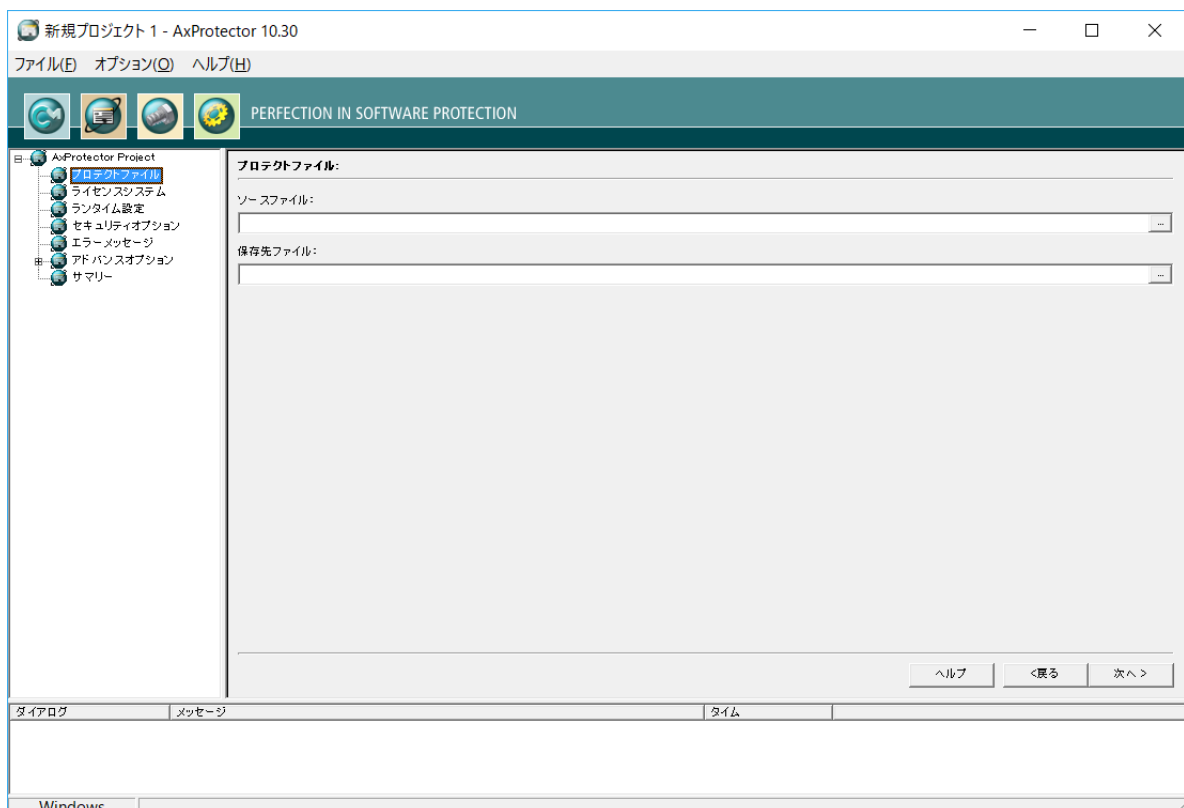
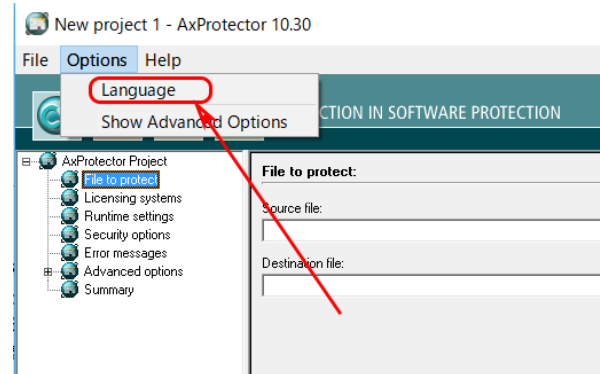
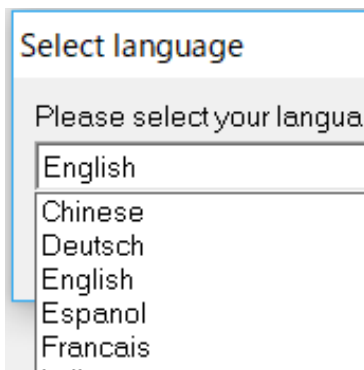
AxProtectorは、【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】からも起動できます。"AxProtectorGui.exe"は、インストール先の¥Program Files(x86)¥WJIBU-SYSTEMS¥AxProtector¥DevKit¥binフォルダの中にあります。

"Choose project type"画面で"Windows Application or DLL"を選択し、OKボタンをクリックします。



画面を日本語モードにする

[Options]-[Language]を選択し、Select language (言語選択)画面で"Japanese"を選択し、OKをクリックします。AxProtectorが日本語モードに変換されます。

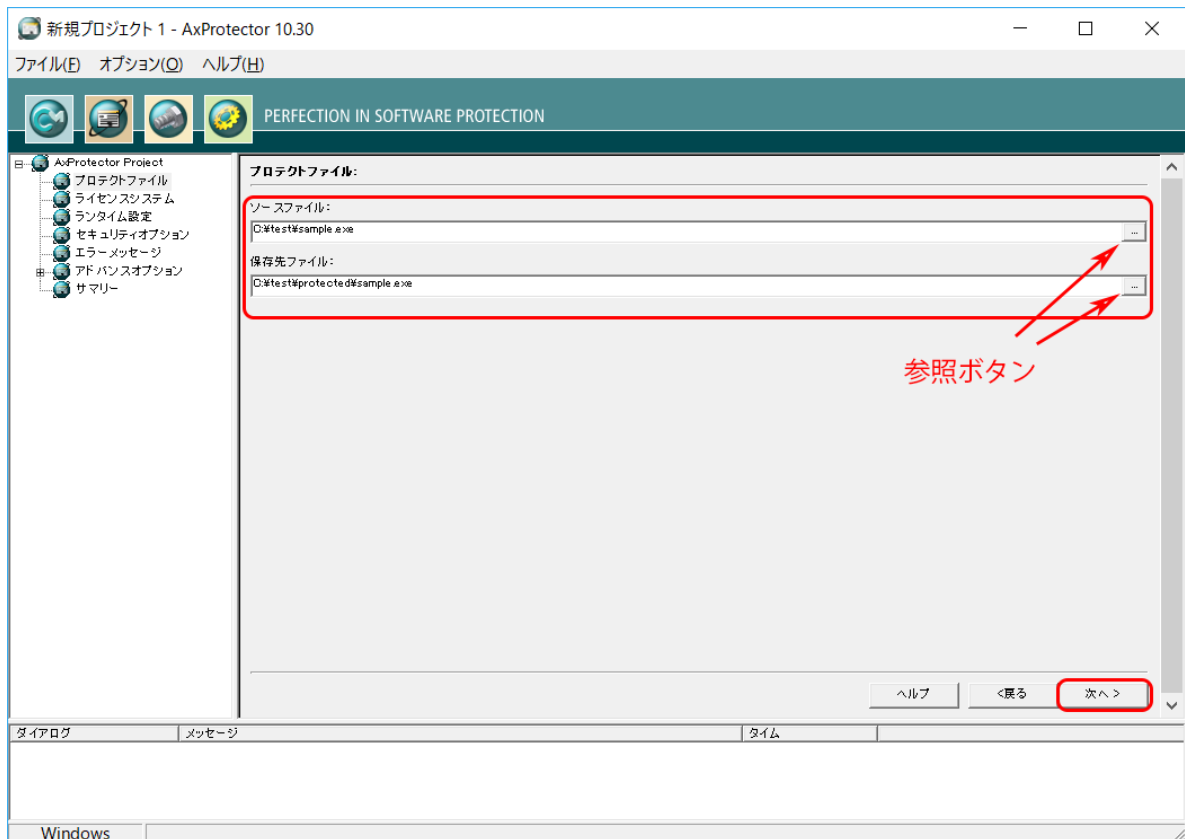


1. プロテクトファイル：

「プロテクトファイル」画面で、「ソースファイル」に、オリジナルファイル名を入力(参照ボタンより)し、「保存先ファイル」にプロテクト生成されるファイル名を入力(参照ボタンより)します。ここでは、「ソースファイル」に「C:¥test¥sample.exe」を設定し、「保存先ファイル」に「C:¥test¥protected¥sample.exe」を設定します。ファイル名の設定が終了したら[次へ]ボタンをクリックして先に進めます。

[NOTE]

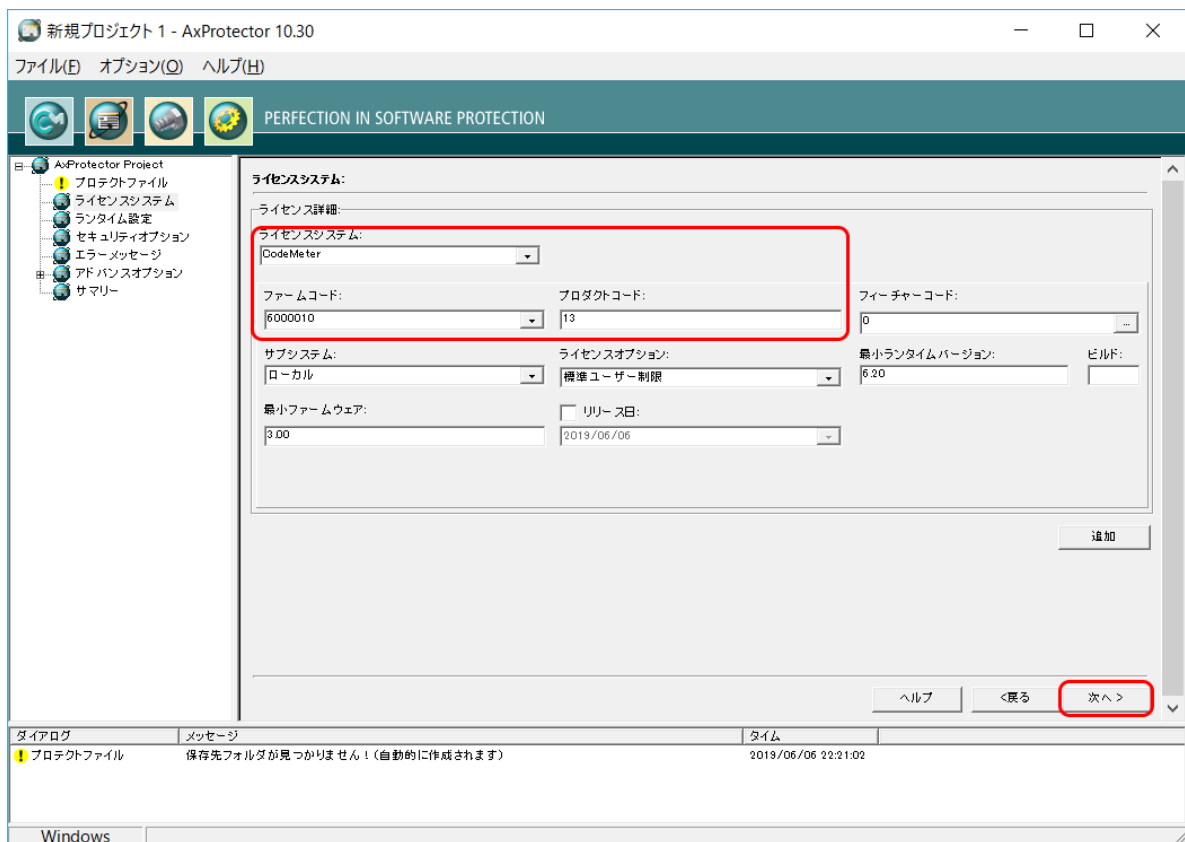
オリジナルファイルを上書きするのを防ぐために、保存先ファイル名はソースファイル名と別名にするか、または異なるパス名(異なるフォルダ)にしてください。ソースファイルを指定すると、同一フォルダにprotectedフォルダが自動的に作成されます。



2. ライセンスシステム：

「ライセンスシステム」画面でライセンスの指定を行います。ここで入力したコード(ファームコード/プロダクトコード)と同じコードを持つCmDongle(ハードウェアキー)またはCmActLicense(ソフトウェアキー)がPCに装着されていないと、暗号化されたプログラムは起動しません。プログラムは、入力されたファームコード、プロダクトコード、その他の要素をベースに生成される独自の暗号キー(ランダム値)によって暗号化されます。

自動暗号化ツールAxProtectorは、コードメータ(CmDongle)、コードメータAct(CmActLicense)、ワイブキー(WibuKey)で共通に使用することができます。ここではコードメータ(CodeMeter)を使用し、ファームコード=6000010、プロダクトコード=13 を設定します。フィーチャーコード、サブシステム、ライセンスオプション、最小ランタイムバージョン、ビルド、最小ファームウェア、リリース日はデフォルトのままにし、「次へ」ボタンをクリックします。



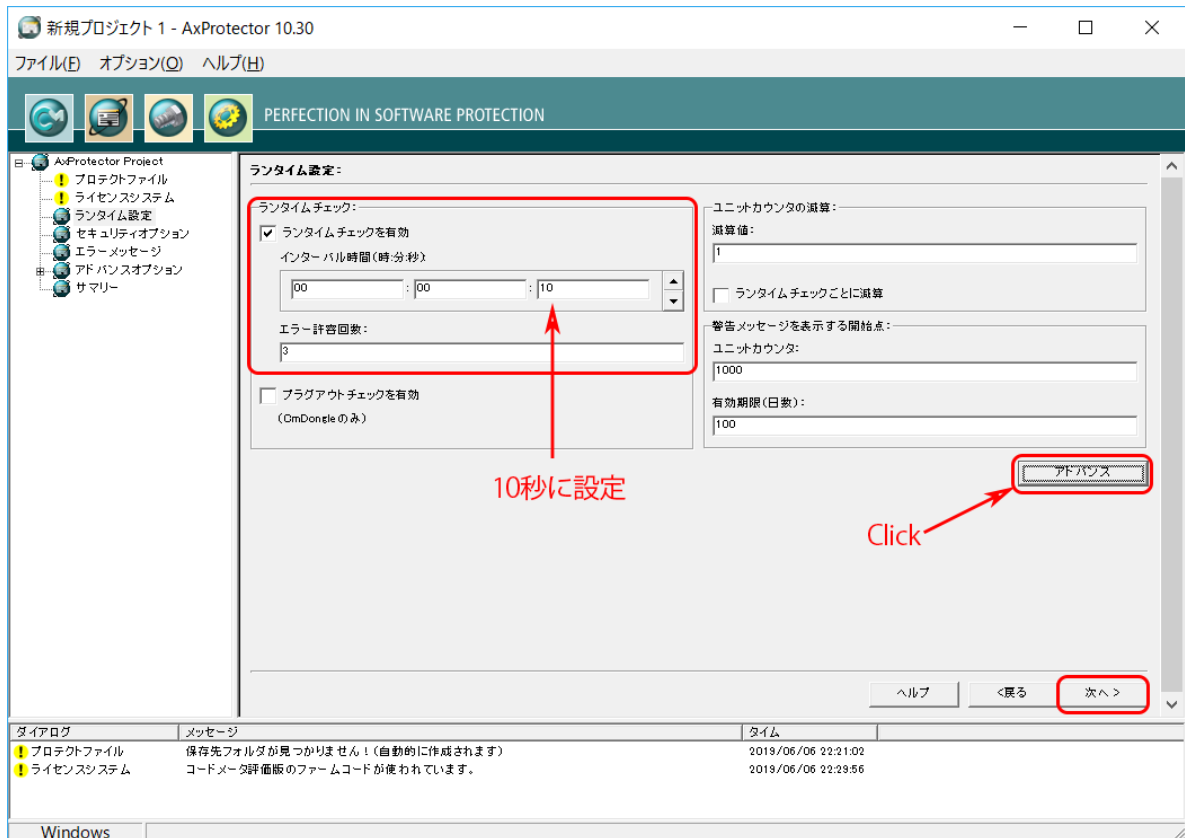
[NOTE]

ファームコードは貴社専用の6桁または7桁のコード(整数値)になり、コードメータ開発キット(CmFSB)を購入した時点で自動的に割り当てられます。世界で唯一、貴社専用のファームコードになります。プロダクトコードは、1~4294967295(32bits)の範囲の整数値が有効で、貴社にて自由に割り当てることが可能です。

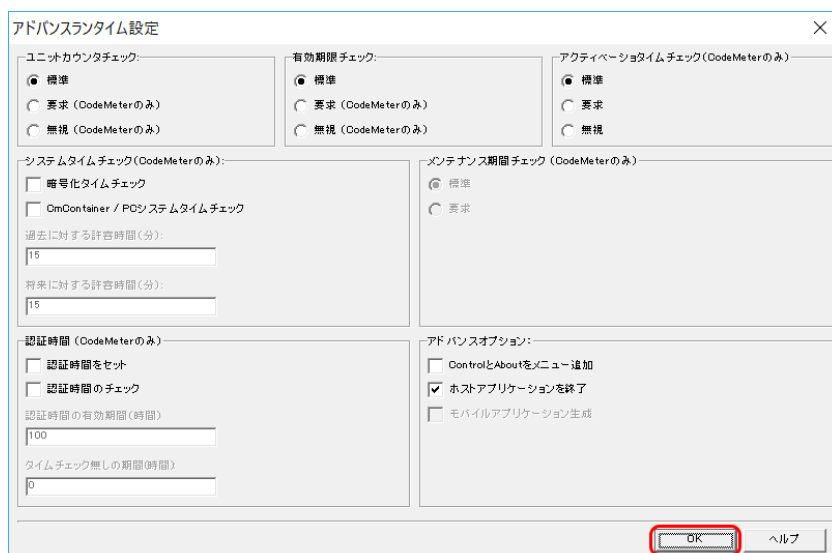
3. ランタイム設定 :

「ランタイム設定」画面で、一定時間ごとにコードメータキーをチェックする「ランタイムチェック」を設定します。このランタイムチェック機能を使うことで、プロテクトされたプログラムの動作中は必ずコードメータキーをPCに装着し続ける必要があり、1個のコードメータキーを使ってプログラムを同時に複数PC上で使用するライセンス違反を防ぐことができます。

設定できるインターバル時間は、時、分、秒単位で設定可能です。デフォルトでは30秒ですが、貴社のセキュリティポリシーに応じてインターバル時間を調整してください。ここでは、テスト的に10秒を設定します。「ランタイムチェックを有効」にチェックを入れ、秒欄(最右部)に10を入力します。

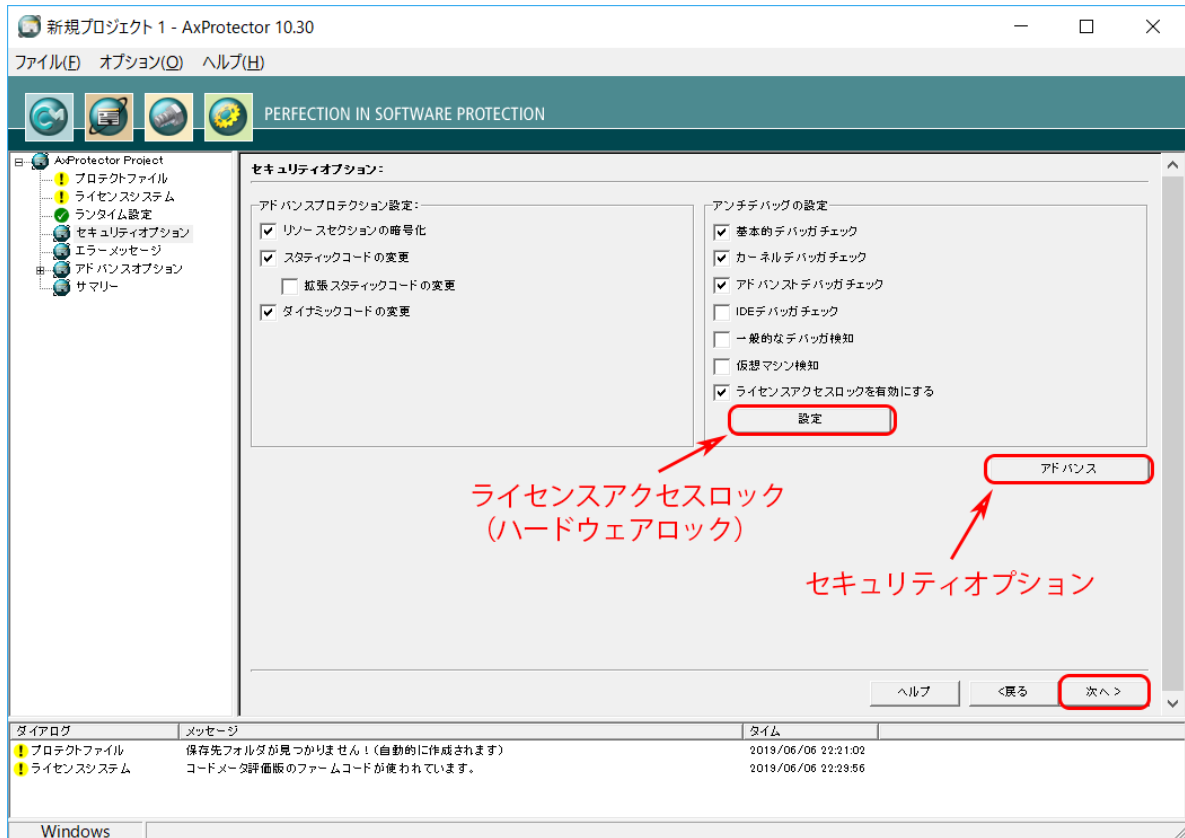


また、「アドバンス」ボタンをクリックすると、さらにきめの細かいランタイム設定を行うことができます。ここでは、特に設定しませんので、「OK」ボタンをクリックして閉じ、ランタイム設定画面の「次へ」ボタンをクリックして次に進みます。



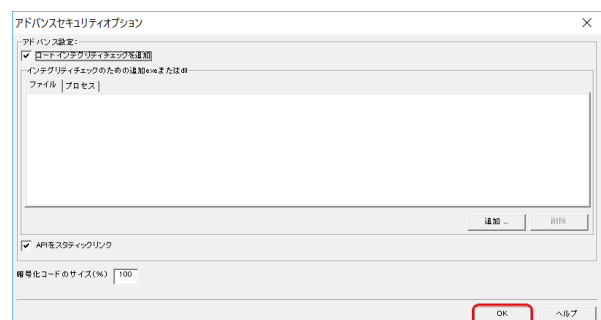
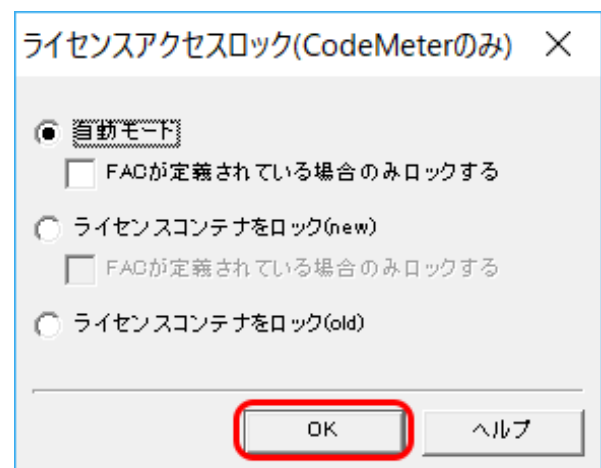
4. セキュリティオプション：

「セキュリティオプション」画面で、セキュリティ強度の設定を行います。暗号化の方法や、デバッグ解析に対するセキュリティポリシーを決めます。AxProtectorは、デフォルトの設定で十分強力な暗号化を実現できるため、ここではデフォルトの状態にします。



アンチデバッグ設定の「設定」ボタンをクリックすると、ライセンスアクセスロック (CodeMeterのみ) の選択画面が表示されます。これは、デバッガ等による不正解析行為を検知した場合、どのようにCmContainerアクセスをロックするかを選択です。ここでは「自動モード」(デフォルト)を選択します。

また、セキュリティオプションの「アドバンス」ボタンをクリックすると、コードインテグリティチェックや暗号化コードのサイズ(%指定)などを設定することができます。ここでは、デフォルトのままOKボタンをクリックします。「4. セキュリティオプション」画面に戻り、「次へ」をクリックして次に進みます。

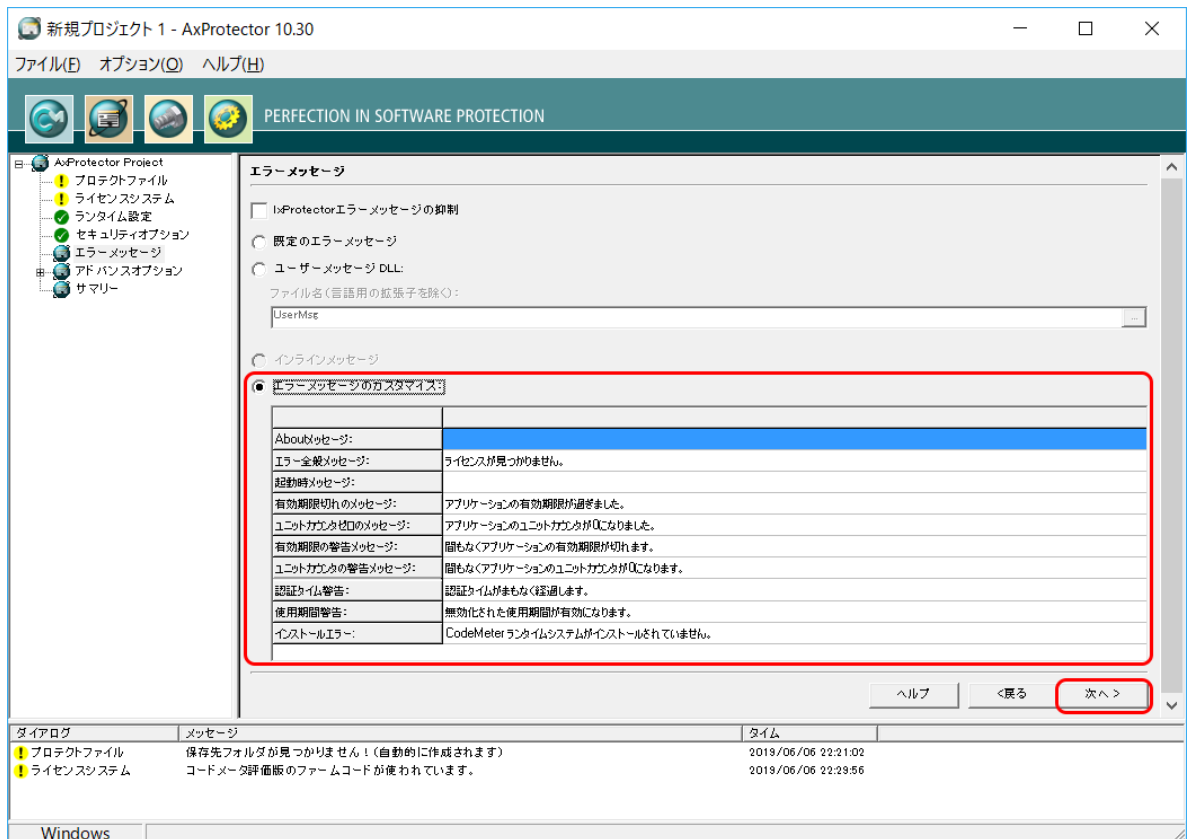


5. エラーメッセージ :

「エラーメッセージ」画面で、エラーメッセージを作成します。エラーメッセージの作成には4通りの方法があります。

- 既定のエラーメッセージ (英語デフォルト)
- ユーザーメッセージ DLL (INIファイルと画像を使ってメッセージを作成する方法)
- インラインメッセージ (.NETアセンブリ用)
- エラーメッセージのカスタマイズ (フォームからメッセージを直接入力する方法)

ここでは、「エラーメッセージのカスタマイズ」にチェックを入れます。



「次へ」をクリックして次に進みます。

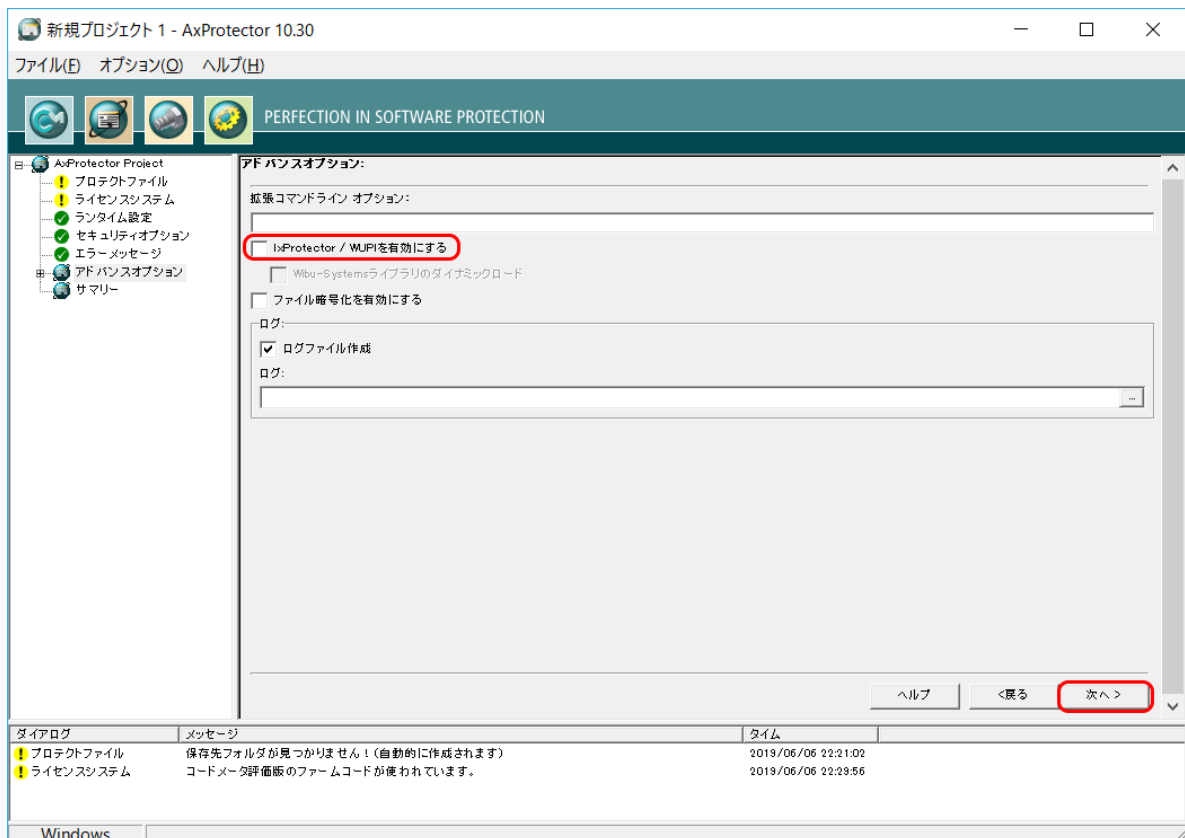
[NOTE]

エラーメッセージの詳細につきましては、「Chapter 4 自動暗号化ツール AxProtectorについて」をご参照ください。

6. アドバンスオプション：

IxProtectorは、メモリー上で展開されるコードを常に暗号化しておき、必要な時に必要なモジュールを復号するメモリー上での「オンデマンド復号」を実現する機能です。AxProtectorで暗号化されたコードが、メモリー上でも暗号化されているため、クラッキングに対して非常に強力なセキュリティを実現できます。

IxProtectorを使用するには、ソースコードにWUPI (Wibu Universal Protection Interface)ファンクションを組み込む必要があります。ここでは、IxProtectorは使用しませんのでチェックをはずしておきます。

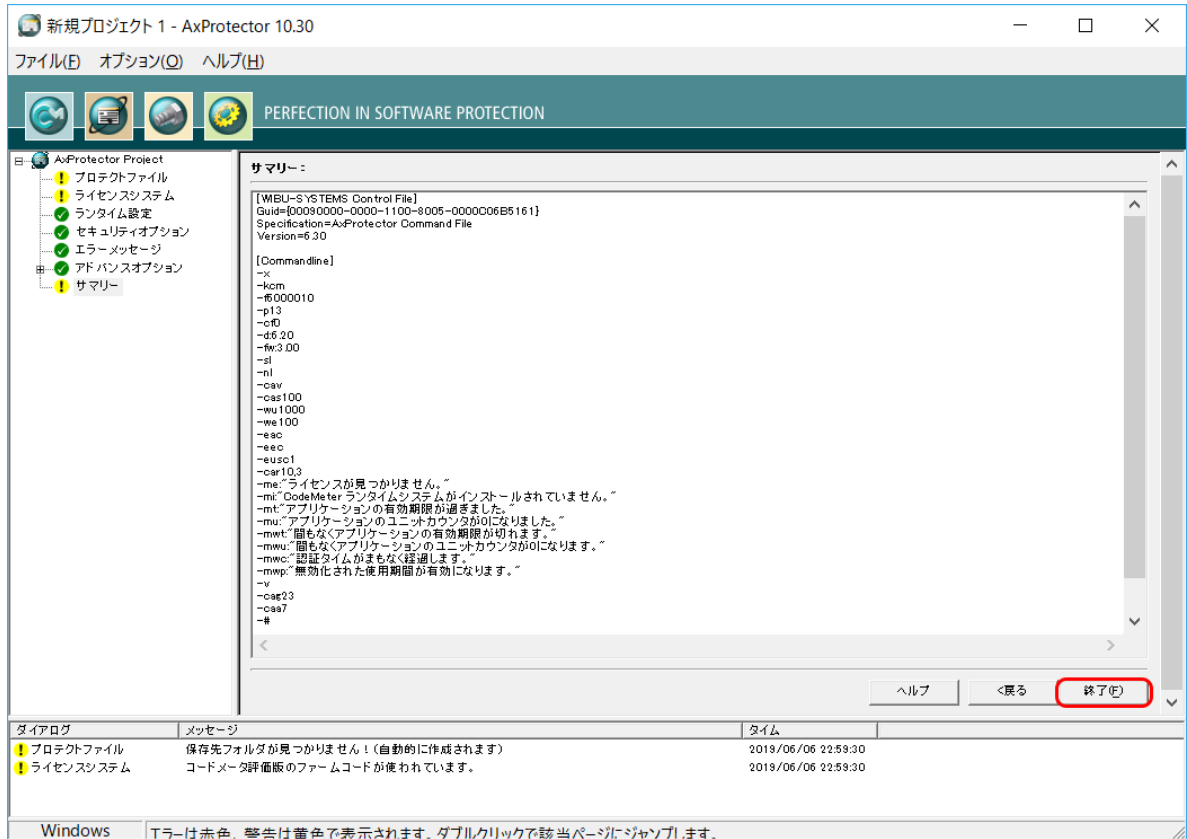


[NOTE]

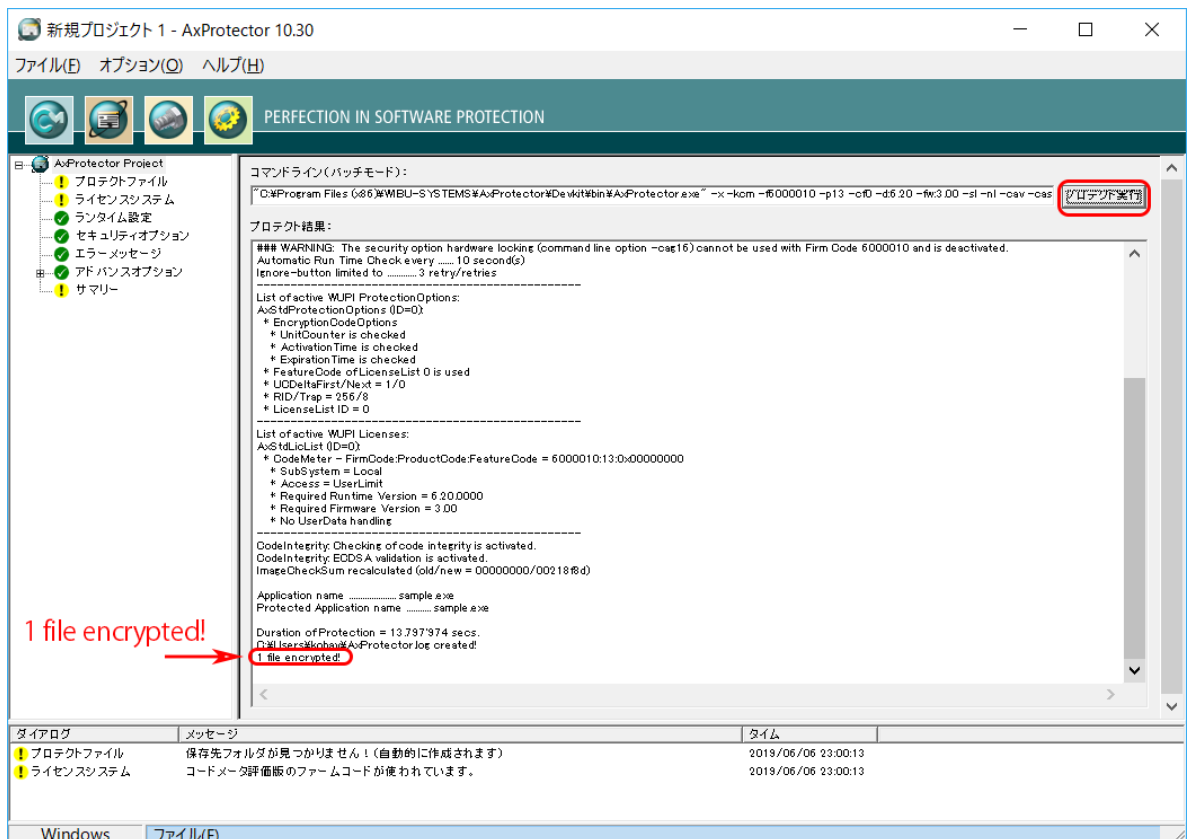
C#やVB.NETなどで作成したMicrosoft .NET Framework対応プログラム(マネージコード)やJavaの場合は、IxProtector/WUPIを使用しなくても、AxProtector(.NETアセンブリ)で暗号化することで自動的に「オンデマンド復号」機能が組み込まれます。「オンデマンド復号」を実現するために、あえてソースコードを修正する必要がありません。IxProtector/WUPIをソースコードに組み込む必要があるのは、C/C++などのネイティブコード(アンマネージコード)の場合です。


7. サマリー (プロテクト内容を確認する) :

「サマリー」画面に、いままでに設定したプロテクト内容が表示されます。



"終了"ボタンをクリックし、暗号化処理を開始します。コードメータFSB(CmFSB)が装着されていないと暗号化処理ができませんのでご注意ください。暗号化処理が正常に行われると、以下の画面に"1 file encrypted!"が表示されます。(右のスクロールバーで最終行までスクロールしてください。)

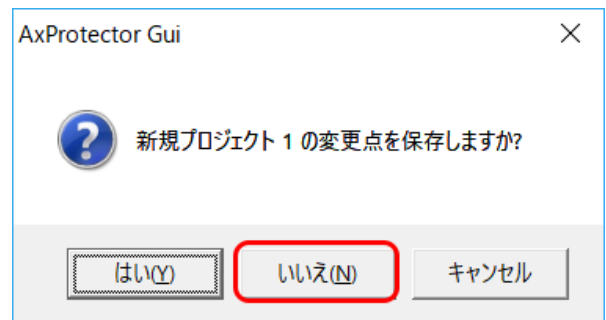


これで、sample.exeの暗号化処理は終了です。メニューバーの「ファイル(F)」-「終了(X)」、または画面右上の  ボタンでAxProtectorを終了します。



AxProtector を終了する時に右のダイアログBoxが表示されます。いままでのプロテクト内容をファイルに保存するかどうかの選択です。ここでは、「いいえ(N)」をクリックして終了します。

指定したフォルダに、暗号化されたsample.exeが作成されていることを確認してください。



3-3. CmStick (CmDongle) にコードを登録する

AxProtectorで暗号化したsample.exeと同じファームコード(Firm Code)とプロダクトコード(Product Code)をCmStick (CmDongle)に登録します。CmStick (CmDongle)にコードを登録するには、4通りの方法があります。

1. コードメータライセンスエディタ(CodeMeter License Editor)を使用する方法
2. CmBoxPgmプログラム(コマンドライン環境)を使用する方法
3. コードメータFAS(リモートプログラミング)を使用する方法
4. CodeMeter License Centralを使用する方法

ここでは、コードメータライセンスエディタ(CodeMeter License Editor)を使って、CmStickにコードを登録します。

1. 用意するもの

CmStickにコードを登録するには下記が必要です。

- コードメータFSB (CmFSB)
- CmStick (CmDongle)

[NOTE]

ここでは、USBタイプのレギュラーサイズCmDongle(CmStick)を例にすすめますが、他のCmDongle (CmStick/C Basic, CmStick/C with Cap, CmCard/SD, CmCard/CFなど) の場合も操作は同じです。

[NOTE]

FSB = Firm Security Boxの略。

2. コードメータライセンスエディタを起動する

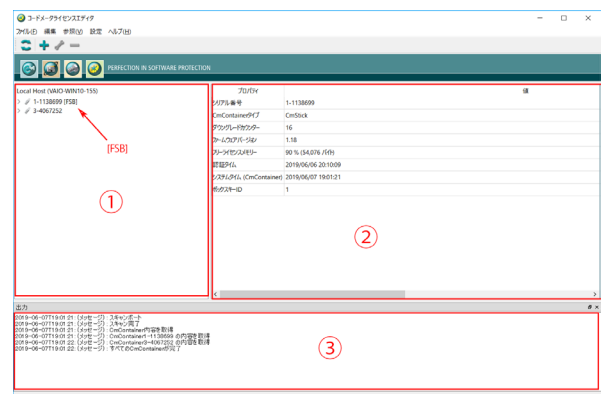
①コードメータ FSB (CmFSB) と CmStick を装着する

まず、コードメータFSB(CmFSB)とコードを登録するCmStickをPCのUSBポートにそれぞれ装着します。必ず、2つを同時に装着します。

②コードメータライセンスエディタを起動する

スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

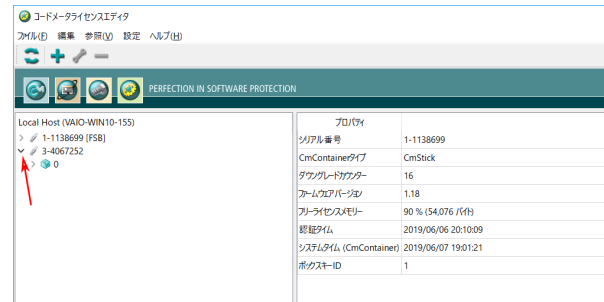
左ペイン①には、CmStickのシリアル番号がツリー状で表示されます。コードメータFSBの場合は、シリアル番号の右側に"[FSB]"と表示されます。右ペイン②には、左ペイン①で選択されたCmStickのプロパティが表示されます。また、下部の出力ペイン③には、「実行」ボタンで実行された操作結果のステータスが表示されます。



③ CmStick の>マークをクリックする

CmStick(ここでは3-4067252)の左部にある>マークをクリックすると、現在登録されているファームコードが表示されます。Vマークをクリックすると元に戻ります。

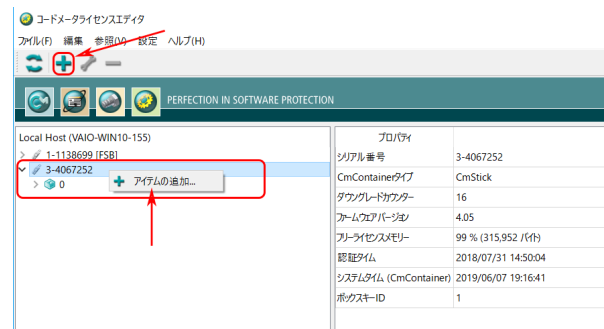
CmStickには、すでにファームコード=0が登録されています。ファームコード=0はCodeMeter System側が使用するため、そのままにします。ここでは、ファームコード=6000010を新規に登録します。次の④「アイテムの追加」ボタンを表示するに進んでください。



④ 「アイテムの追加」ボタンを表示する

CmStick(ここでは3-4067252)を選択し、右マウスをクリックすると「アイテムの追加」ボタンが表示されます。その「アイテムの追加」ボタンをクリックすると、ファームアイテム設定画面が表示されます。

ファームアイテム設定画面は、ツールメニューの+ボタンをクリックして表示することもできます。



⑤ ファームコードを入力する

「ファームアイテムの追加」画面で、ファームコードの入力欄にファームコード=6000010を選択し、「実行」ボタンをクリックします。



[NOTE]

正式なファームコードは6桁または7桁の整数値になります。弊社から出荷されたCmDongleには、すでに貴社のファームコードが登録されています。実際には、貴社のファームコードの下にプロダクトコードを追加して使用します。ここでは、サンプル例として、ファームコード=6000010を追加しています。

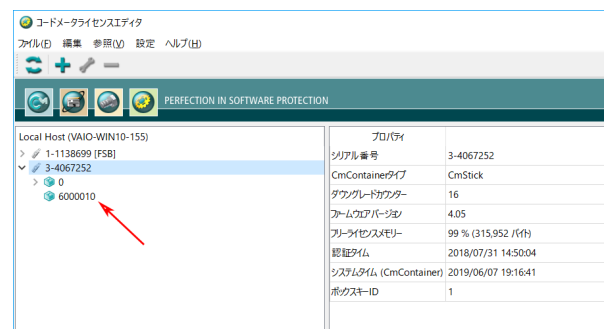
「ファームアクセスカウンタ」は、暗号化・復号をコントロールするカウンタです。通常はデフォルトのまま使用します。

「ファームアイテムテキスト」は、ファームコードに対するコメント文です。半角英数文字で80文字まで使用可能です。

「ファームアイテムの追加」画面で「実行」ボタンをクリックすると、CmStick(ここでは3-4067252)にファームコード=6000010が登録されます。

[NOTE]

ファームコード=6000010は、あとから削除可能です。ただし、貴社の正式ファームコードは一度登録すると削除することができません。



⑥ プロダクトコードを入力する

ファームコード"6000010"を選択し、右マウスをクリックすると「追加・修正・削除」メニューが表示されます。「アイテムの追加」をクリックして、プロダクトアイテム設定画面を表示させます。

「プロダクトアイテム追加」画面で、プロダクトコードを指定します。

この「プロダクトアイテム追加」画面では、プロダクトコードだけでなく、プロダクトコードに対してユニットカウンタ、ライセンス数、使用期間(日数)、フィーチャーマップ、アクティベーションタイム(使用開始日)、有効期限などのオプション項目も設定することができます。

ここでは、プロダクトコードに"13"を入力して「OK」ボタンをクリックします。

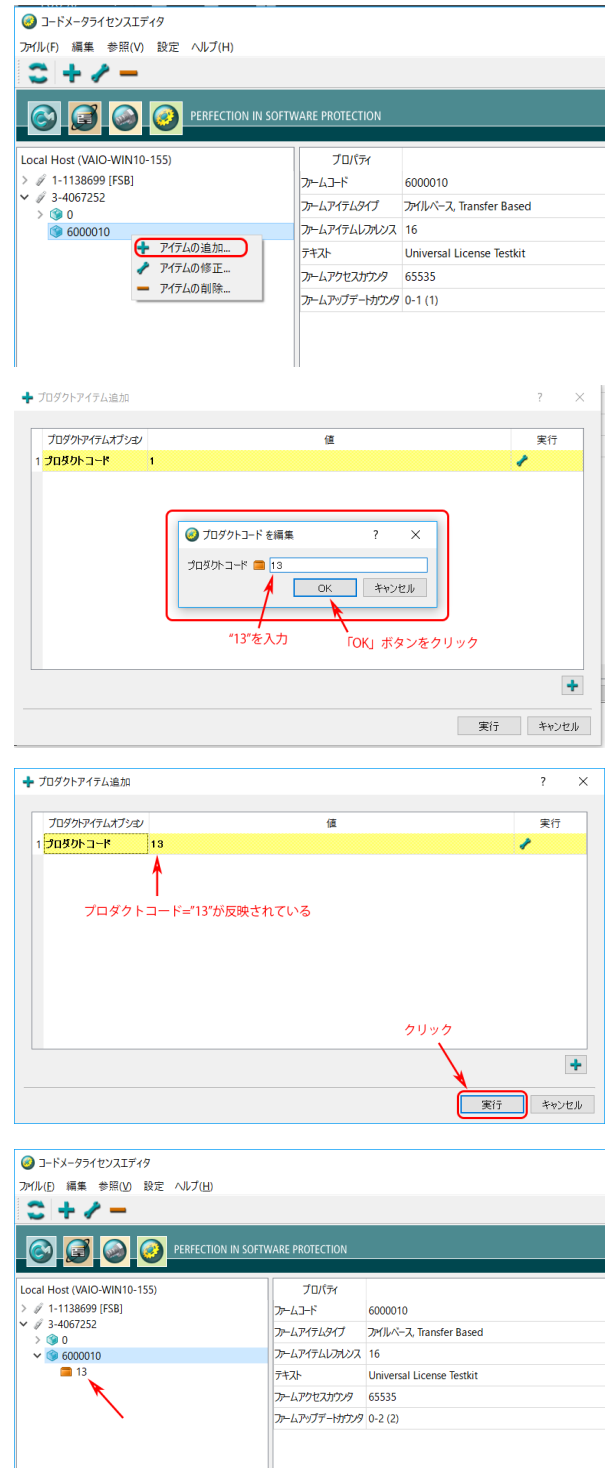
⑦ CmStick に登録する

プロダクトコードにプロダクトコード=13が反映されていることを確認したら、右下の「実行」ボタンをクリックします。

右下の「実行(E)」ボタンをクリックすると、ファームコード=6000010、プロダクトコード=13がCmStickに書き込まれます。

⑧ 登録されたことを確認する

CmStick 3-4067252の>マークをクリックし、ファームコード、プロダクトコードが正しく登録されたことを確認してください。ファームコードの>マークをクリックするとプロダクトコードが展開されます。

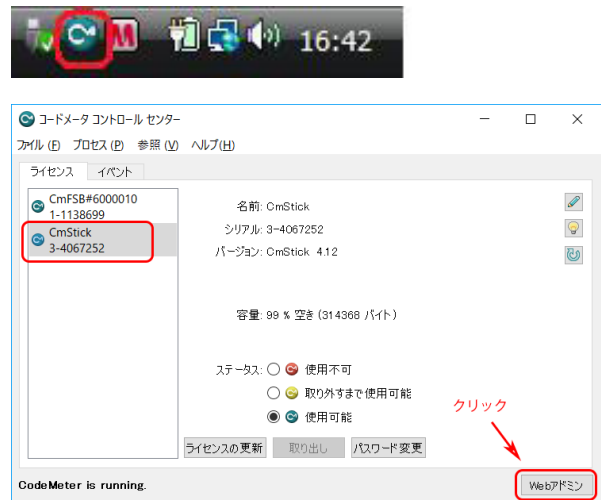


3. Web アドミン (WebAdmin) 上からも確認する

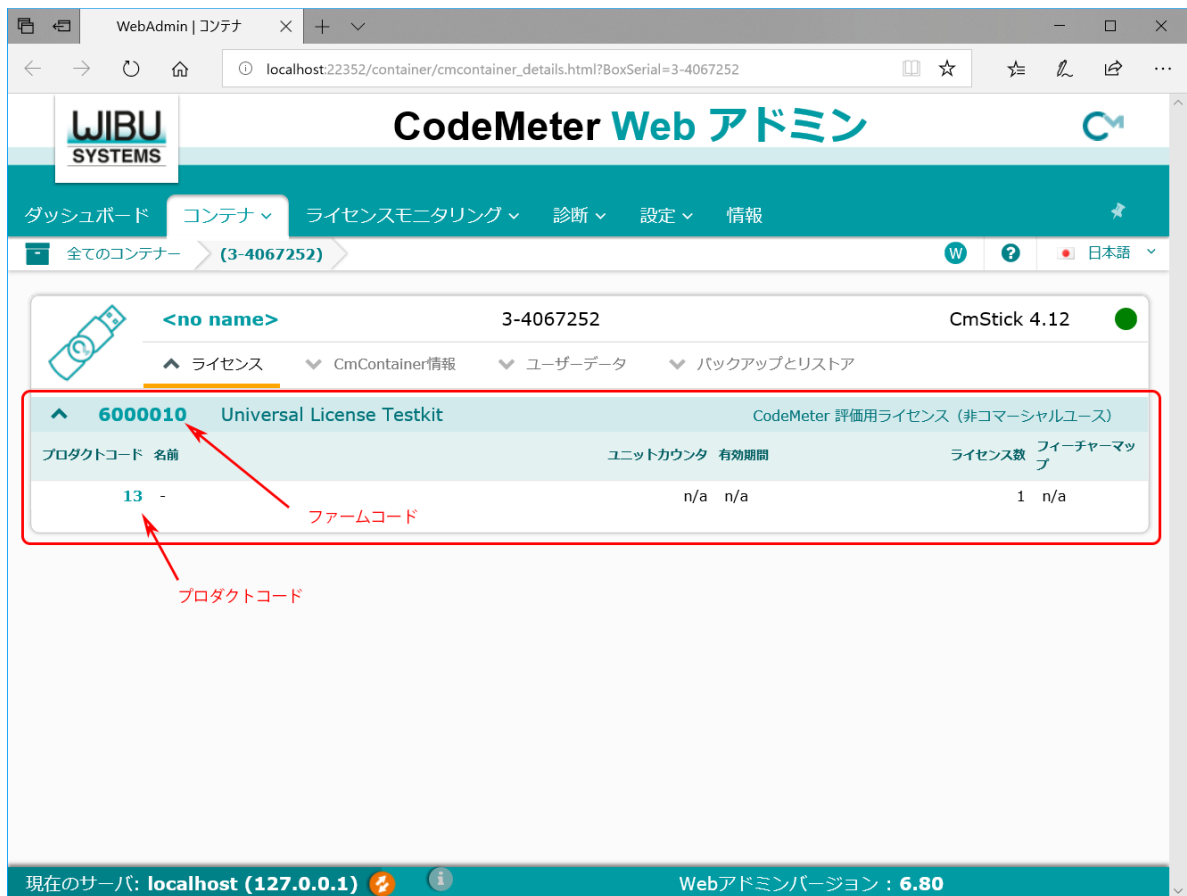
CmStickに書き込んだ内容を、コードメータWebアドミン(WebAdmin)を使って確認することもできます。このWebアドミン(WebAdmin)は、CmDongleを管理する便利なツールです。

Windows起動時にコードメータコントロールセンターが起動し、タスクトレイに常駐します。手動で起動させる場合は、スタートボタンから[すべてのプログラム]-[CodeMeter]-[CodeMeter Control Center]をクリックします。コードメータコントロールセンターが起動するとタスクトレイにアイコンが表示されます。

アイコンをクリックして、コードメータコントロールセンターを開き、CmStick 3-4067252を選択し、右下の「Webアドミン」ボタンをクリックします。



Webブラウザが起動し、「CodeMeter Webアドミン」が表示され、[コンテナ]タブページに、CmStick 3-4067252の情報が表示されます。ファームコード=6000010 とプロダクトコード=13が登録されていることが確認できます。



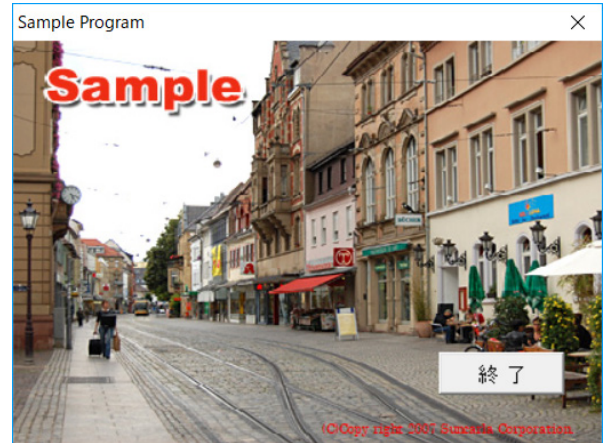
[NOTE]

大量のCmStickにコードを書く場合は、CmBoxPgm.exeを使って、コマンドラインからバッチ処理で書く方法が便利です。CmBoxPgm.exeの使い方は、「Chapter 10 CmBoxPgmの使い方」をご参照ください。

3-4. 動作を確認する

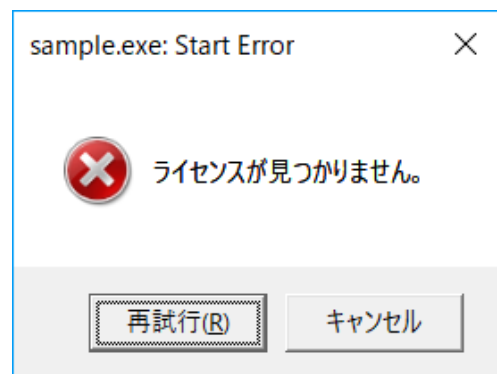
コードを登録したCmStickをPCに装着し、先ほど暗号化したsample.exeを起動します。正常に起動することを確認してください。次に、CmStickを取り外した状態で、sample.exeを起動します。エラーメッセージが表示されsample.exeが起動しません。

sample.exeが正しく起動すると右の画面が表示されます。（「終了」ボタンをクリックするとプログラムが終了します。）



エラーの場合は、右のメッセージが表示されます。正しいCmStickが無いと暗号化されたsample.exeが起動しません。

これで、プロテクト処理は完了です。



[NOTE]

エラーメッセージは、独自でカスタマイズすることができます。
詳しくは、「Chapter 4 自動暗号化ツール AxProtector!について」をご参照ください。

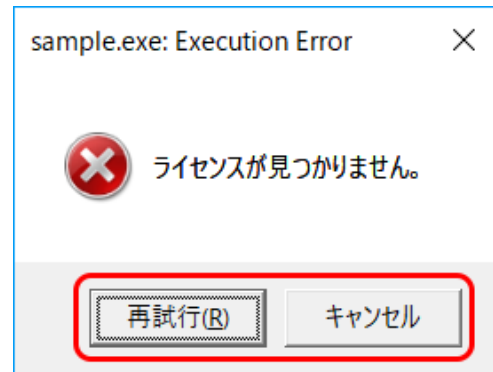
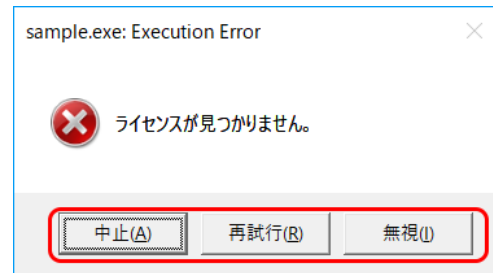
ランタイムチェックを確認する

sample.exeが起動された状態で、コードメータCmStickをPCから取り外します。ランタイムチェックのインターバル時間を10秒に設定してあるので、10秒後に右のメッセージが表示されます。

コードメータCmStickをPCに装着して「再試行(R)」ボタンをクリックすると、sample.exeに制御が移ります。「中止(A)」ボタンをクリックすると、sample.exeは正常に終了します。「無視(I)」ボタンをクリックすると、「エラー許容回数」で設定した回数だけsample.exeを続行することができます。

今回は、「エラー許容回数」を"3" (デフォルト)に設定したため、「無視(I)」が3回表示されました。「エラー許容回数」の設定は、AxProtectorの「ランタイム設定」で行います。

「エラー許容回数」で設定した回数を超えると、「無視(I)」ボタンが表示されなくなり、コードメータCmStickを装着して再試行するか、「キャンセル」ボタンをクリックしてsample.exeを終了させるかの2つの選択だけになります。



[NOTE]

このランタイムチェック機能を利用することで、1個のコードメータCmStick/Cを使って複数のPC上で同時にプログラムを使用するライセンス違反(1個のキーによる使い回し行為)を防止できます。このランタイムチェックを使用しない場合は、AxProtectorの「ランタイム設定」画面で、「ランタイムチェックを有効」オプションのチェックをはずして暗号化処理を行います。この場合、コードメータチェックはプログラムの起動時だけ行われ、プログラム起動後はチェックが行われなくなります。

また、最初のエラーから「無視(I)」ボタンを表示させたくない場合は、「エラー許容回数」に"0"を設定します。

3-5. 使用回数（ユニットカウンタ）を設定したプロテクトを行う

sample.exeに対し、使用回数(ユニットカウンタ=Unit Counter)を設定したプロテクトを行ってみます。プログラム起動回数を設定するには、コードメータCmStickにユニットカウンタを設定します。

現在のCmStickのファームコード = 6000010、プロダクトコード=13には、ユニットカウンタが設定されていません。("n/a"の状態)この状態では、プログラムの使用回数(起動回数)は無制限になります。

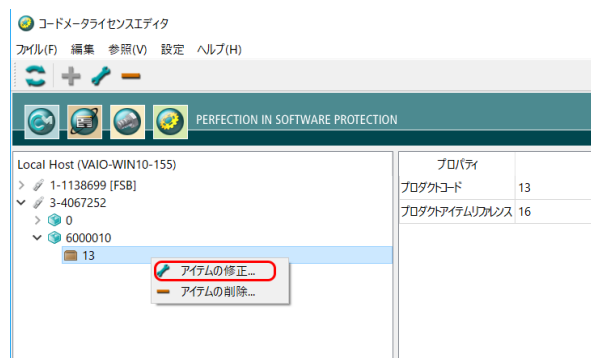


ユニットカウンタを"5"に設定し、プログラム起動回数を制限します。ユニットカウンタを編集するには、コードメータライセンスエディタを使って行います。

①コードメータライセンスエディタを起動する

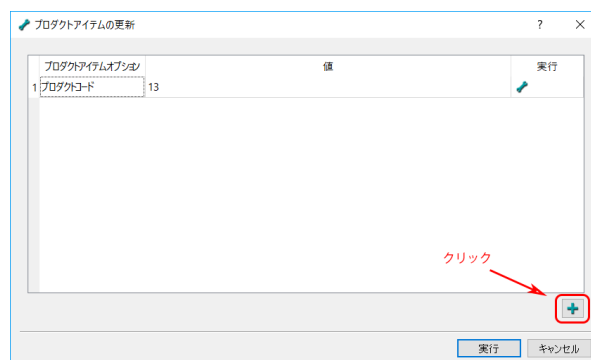
コードメータFSB(CmFSB)とCmStickをPCに装着して、スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

CmStickのファームコード=6000010のプロダクトコード=13を選択し、右マウスをクリックして「アイテムの修正...」をクリックします。

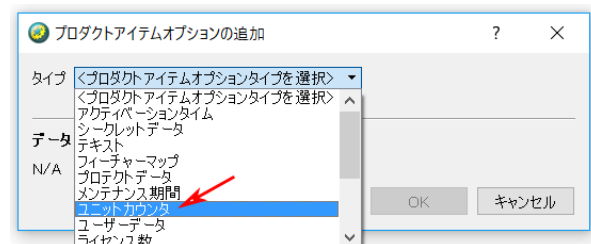


②ユニットカウンタを設定する

「プロダクトアイテムの更新」画面右下の+ボタンをクリックします。

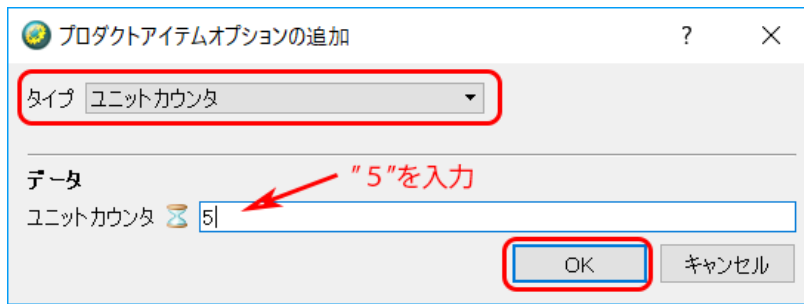


「プロダクトアイテムオプションの追加」画面の「タイプ」で「ユニットカウンタ」を選択します。

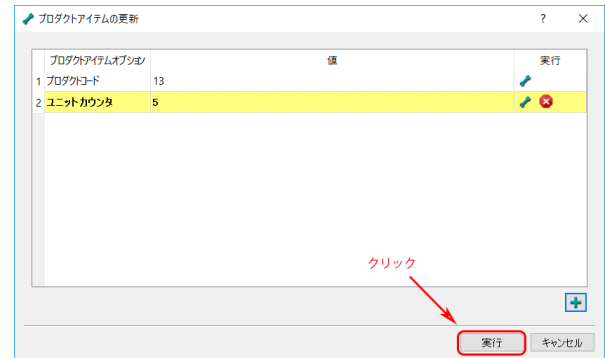


③ 「5」を入力する

「データ」の「ユニットカウンタ」に"5"を入力し、「OK」ボタンをクリックします。

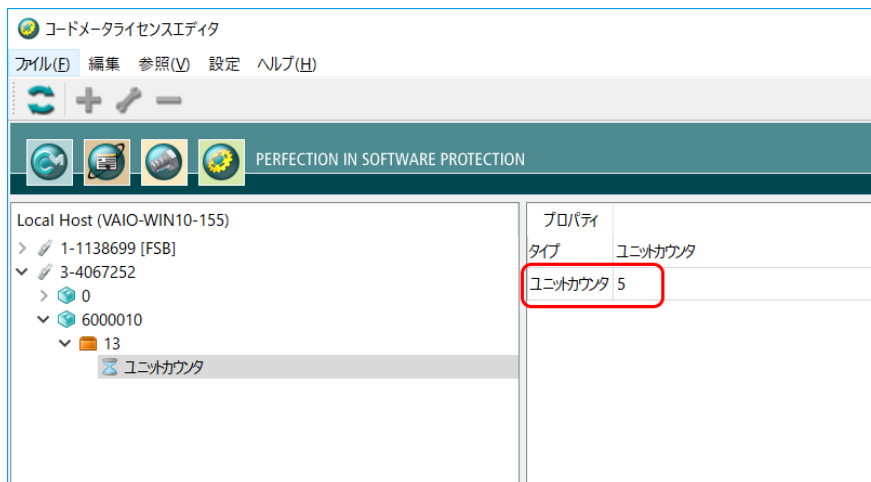


「実行」ボタンをクリックすると、CmStickのFC:PC=6000010:13のユニットカウンタに"5"が登録されます。



④ ユニットカウンタを確認する

ライセンスエディタ上でユニットカウンタが"5"になっていることを確認します。



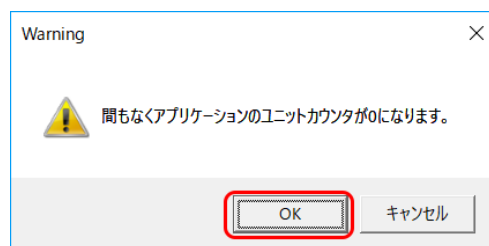
Webアドミン上からも確認できます。



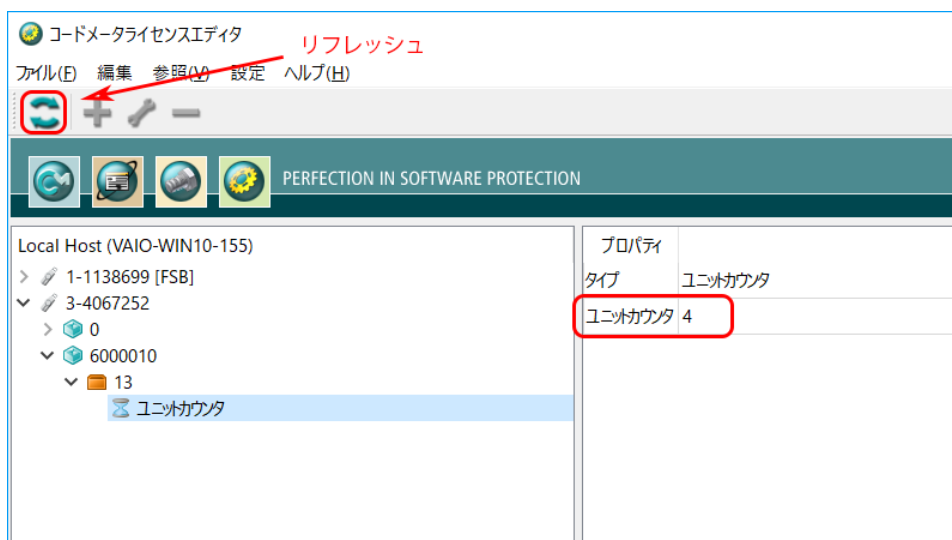
⑤ sample.exe を起動しユニットカウンタを確認

sample.exeを一度起動すると、ユニットカウンタが1つ減ったことが確認できます。

sample.exeを実行すると右の警告メッセージが表示されますが、これはユニットカウンタが残り1000回に達した際に毎回出るメッセージです。今回は、AxProtectorで暗号化した時にデフォルト(残り1000回)で設定したために出ます。貴社にてこの警告メッセージを出すタイミング(残り回数)を自由に設定できます。「OK」ボタンをクリックして続行します。



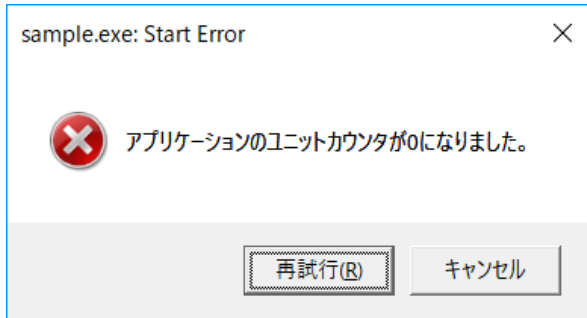
コードメータライセンスエディタ上で、「リフレッシュ」ボタンをクリックすると、新しいユニットカウンタが反映されます。



Webアドミン上からも確認できます。



sample.exeを5回起動すると、6回目は起動できないことを確認してください。



Webアドミン上で、ユニットカウンタが"0"になったことを確認してください。



ユニットカウンタは、コードメータFSB(CmFSB)があれば、コードメータライセンスエディタを使って、何回でも更新することができます。

このユニットカウンタ機能は、貴社ソフトウェアの評価版・デモ版、または使用料課金(使用回数により課金する)販売に利用すると非常に効果的です。

3-6. 使用有効期限 (Expiration Time) を設定したプロテクトを行う


sample.exeに対し、使用有効期限(Expiration Time)を設定したプロテクトを行ってみます。プログラムの使用有効期限を設定するには、コードメータCmStickの「有効期限」に日時を設定します。

まず、コードメータライセンスエディタを起動し、前章「3-5.使用回数(ユニットカウンタ)を設定したプロテクトを行う」で設定したユニットカウンタを削除します。

①ユニットカウンタを削除する

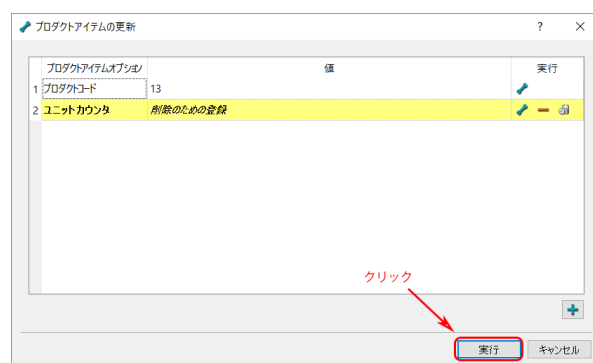
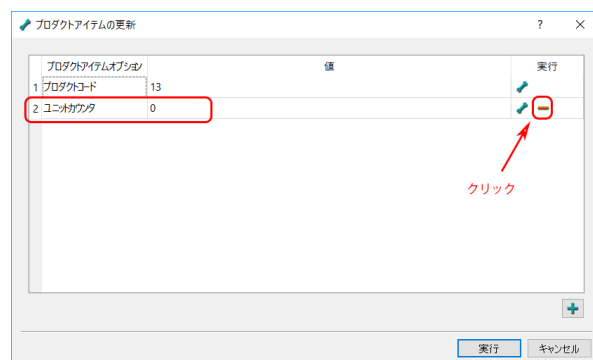
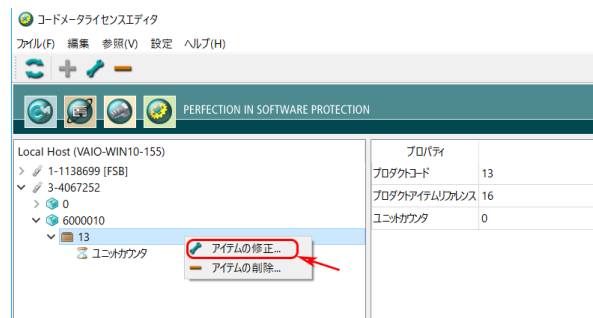
コードメータFSB(CmFSB)とCmStickをPCに装着し、スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

プロダクトコード(ここでは13)を選択し、右マウスをクリックし、「アイテムの修正...」メニューをクリックします。

「プロテクトアイテムの更新」画面の「ユニットカウンタ」項目の右端  ボタンをクリックします。

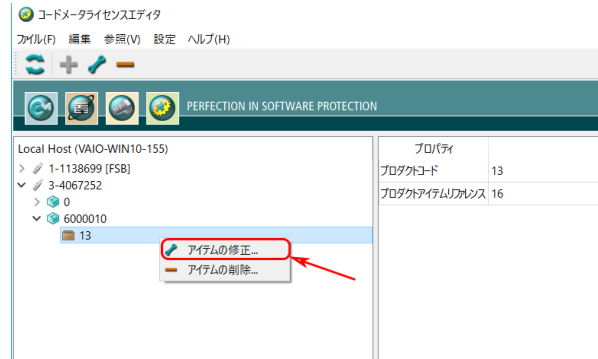
さらに、右下の「実行」ボタンをクリックすると、ユニットカウンタが削除されます。

ライセンスエディタで確認してください。

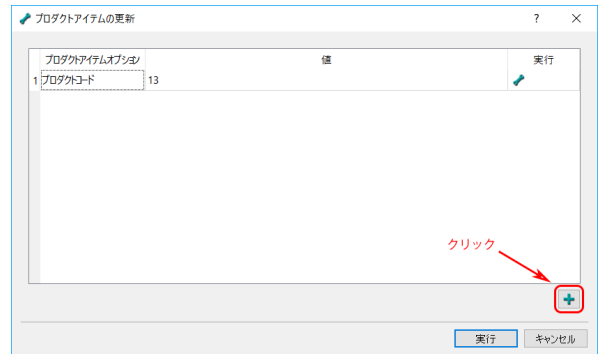


② 有効期限を設定する

ライセンスエディタ上でプロダクトコード"13"を選択し、右クリックで「アイテムの修正...」メニューを選択します。



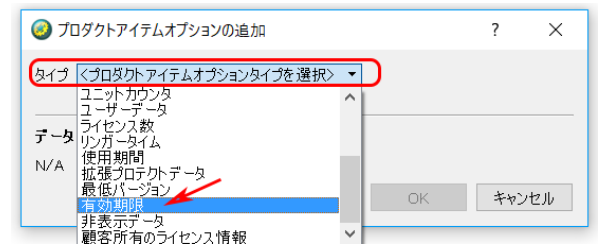
「プロダクトアイテムの更新」画面の右下の+ボタンをクリックします。



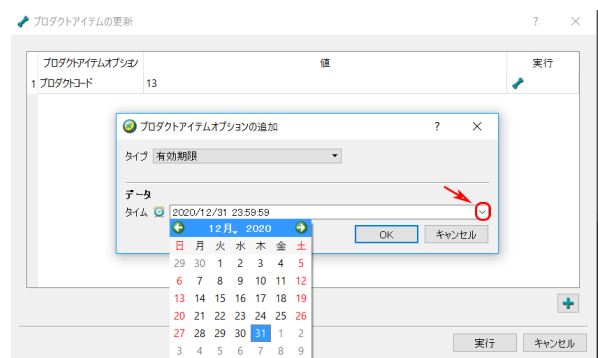
「プロダクトアイテムオプションの追加」画面の「タイプ」項目で「有効期限」を選択します。

[NOTE]

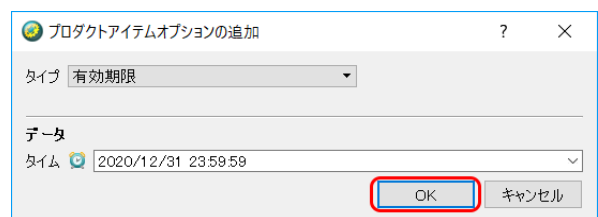
「有効期限」とは別に「使用期間」がありますのでご注意ください。「使用期間」はプログラムの使用可能日数を設定するオプションです。(30日、60日等)



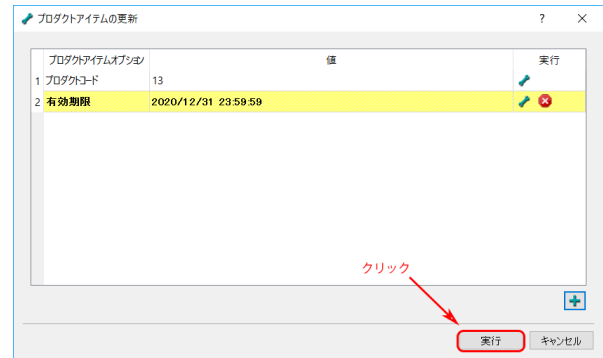
右端のボタンをクリックするとカレンダーが表示されます。有効期限を設定します。



有効期限は直接入力も可能です。入力後、「OK」ボタンをクリックします。



「実行」ボタンをクリックすると、CmStickに有効期限が登録されます。



ライセンスエディタ上で確認してください。



Webアドミン上でも確認できます。



これで、使用有効期限2020年12月31日のCmStickが作成できました。sample.exeは、有効期限までは使用できますが、有効期限を過ぎると起動できなくなります。この使用有効期限は、CmStick内に刻まれたシステムタイム(CmContainer)を参照しているため、PCのシステム時計を過去に戻してもプログラムは動作しません。CmStick内のシステムタイムは過去に戻せないため、確実にセキュリティを維持できます。

3-7. アクティベーションタイム（使用開始日）を設定したプロテクトを行う

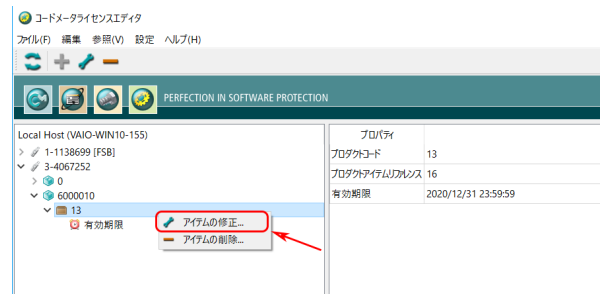
使用有効期限 (Expiration Time)と同じように、プログラムの使用開始日(アクティベーションタイム = Activation Time)を設定することができます。指定した期日にならないとプログラムの起動ができません。使用有効期限と組み合わせて使用すると、確実に使用可能日数を設定することができます。

ファームコード=6000010/プロダクトコード=13に対して、アクティベーションタイム(使用開始日)=2020年1月1日、使用有効期限=2020年12月31日を設定します。

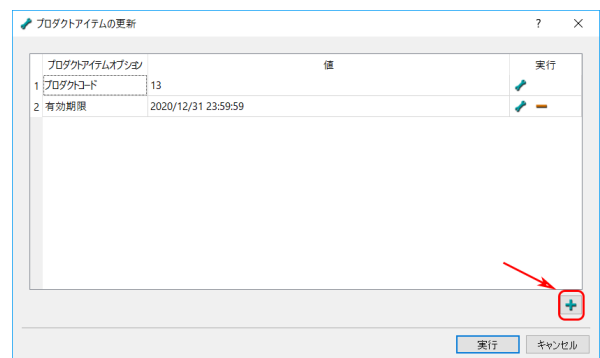
①アクティベーションタイムを設定する

コードメータFSB(CmFSB)とCmStickをPCに装着してから、[スタート]/[すべてのプログラム]/[CodeMeter]/[Tools]/[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

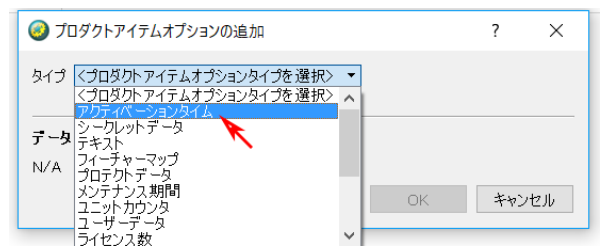
プロダクトコード"13"を選択し、右マウスをクリックし、「編集」メニューをクリックします。



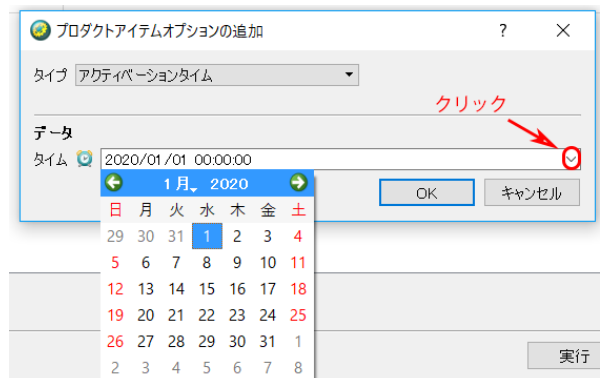
「プロダクトアイテムの更新」画面の「+」ボタンをクリックします。



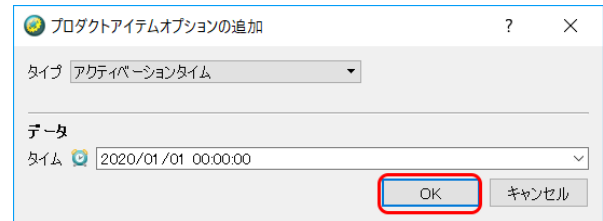
「プロダクトアイテムオプションの追加」画面で「アクティベーション」を選択します。



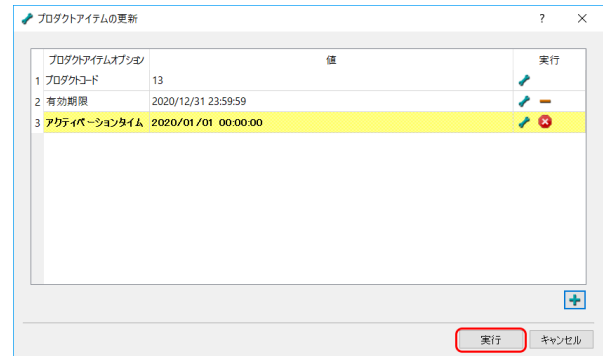
「タイム」欄に"2020/01/01 00:00:00"を設定します。



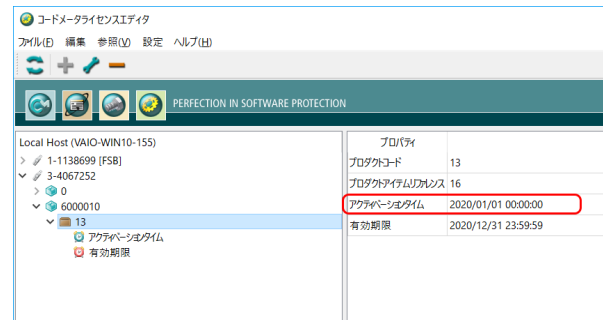
直接、入力・編集することも可能です。
入力後、「OK」ボタンをクリックします。



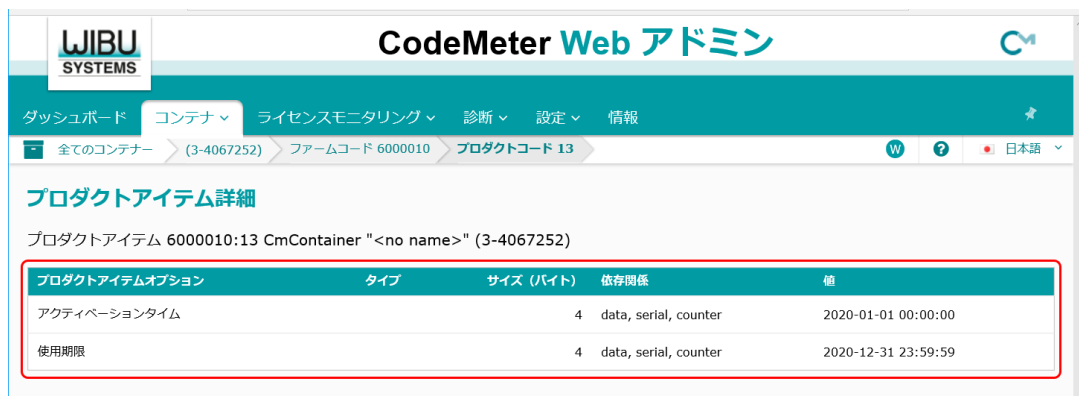
②「実行 (E)」 ボタンをクリックして登録する
「実行」ボタンをクリックすると、CmStickにアクティベーションタイムが登録されます。



③ 登録内容を確認する
ライセンスエディタ上で確認してください。



Webアドミン上でも確認できます。



これで、使用開始日2020年1月1日、使用有効期限2020年12月31日のCmStickが作成できました。
sample.exeは、使用開始日から使用有効期限までの期間で使用できますが、それ以外の期間では起動しません。

3-8. 使用期間 (Usage Period) を設定したプロテクトを行う

使用有効期限 (Expiration Time) や使用開始日 (アクティベーションタイム/Activation Time) とは別に、コードメータには、使用期間 (Usage Period) を設定する機能があります。これは、ソフトウェアの使用可能な期間 (日数) を限定する機能で、例えば、「30日間使用可能」という期間 (日数) を設定することができます。

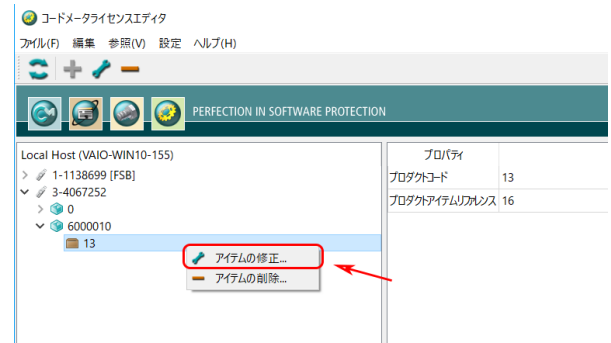
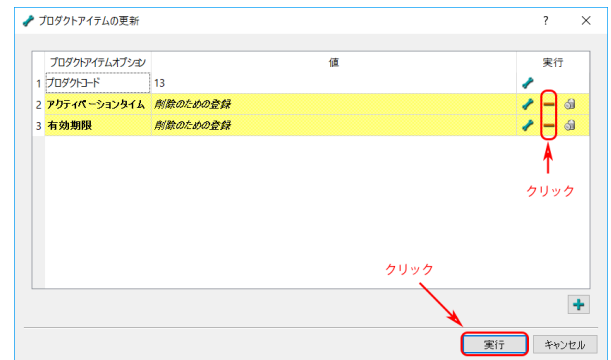
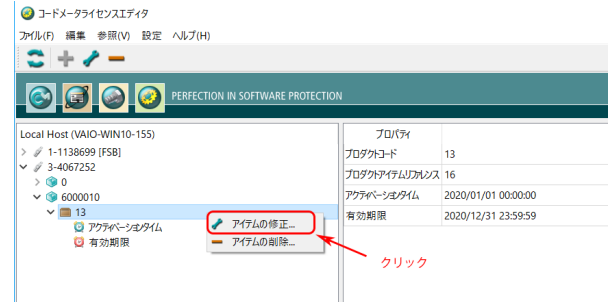
① 使用期間 (Usage Period) を設定する

コードメータFSB(CmFSB)とCmStickをPCに装着してから、[スタート]/[すべてのプログラム]/[CodeMeter]/[Tools]/[CodeMeter License Editor] を選択し、コードメータライセンスエディタを起動します。

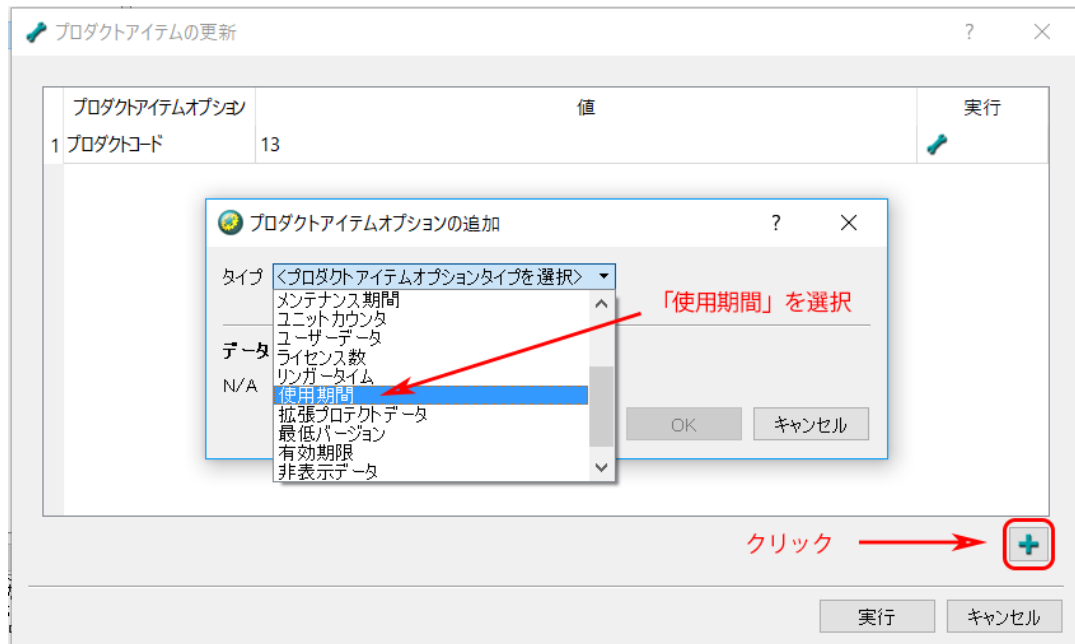
プロダクトコード"13"を選択し、右マウスをクリックし、「アイテムの修正...」メニューをクリックします。すでに、有効期限やアクティベーションタイムが登録されている場合は、あらかじめ削除しておいてください。

「プロダクトアイテムの更新」画面で、「アクティベーションタイム」と「有効期限」の右端「-」ボタンをクリックし、「実行」ボタンをクリックすると削除できます。

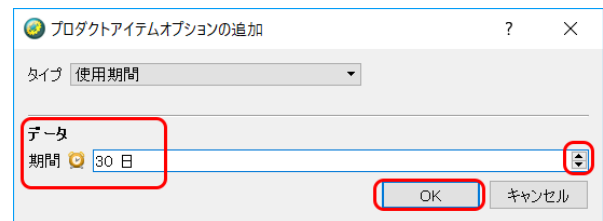
プロダクトコード"13"を選択し、右マウスをクリックし、「アイテムの修正...」メニューをクリックします。



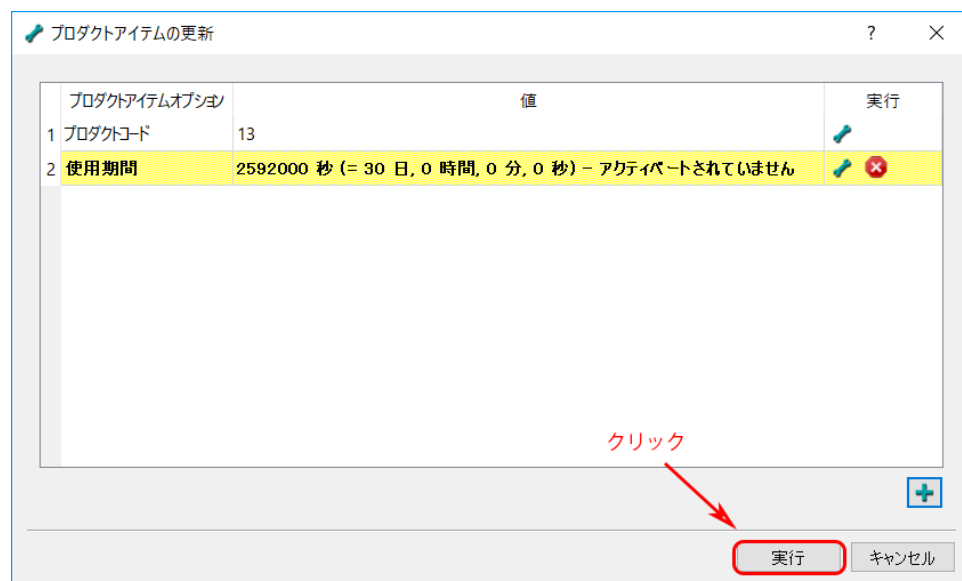
「プロダクトアイテムの更新」画面の「+」ボタンをクリックして「プロダクトアイテムオプションの追加」画面を表示し、「使用期間」を選択します。



「期間」項目に「30日」を設定し、「OK」ボタンをクリックします。



② 「実行」 ボタンをクリックして登録する



③ 登録内容を確認する

左ペインのプロダクトコード"13"には、新たに使用期間が追加されているのが確認できます。右ペインのプロパティには、"使用期間"に使用可能日数が設定されているのが確認できます。



これで、「使用期間=30日間」というCmStickが作成されます。Webアドミンから、プロダクトコード=13をクリックして確認してください。



「使用期間」の「値」の欄に

30 日 - 開始時間: n/a

と登録されています。

"開始時間: n/a"とは、まだ一度もプログラムを起動していないために開始時刻が記録されていないことを表します。実際に、暗号化されたプログラムを初めて起動すると、その時の時刻が使用開始時刻としてCmStickに登録されます。そして、その時刻から数えて30日間が使用有効期間になります。

WIBU SYSTEMS CodeMeter Web アドミン

ダッシュボード コンテナ ライセンスモニタリング 診断 設定 情報

全てのコンテナ (3-4067252) ファームコード 6000010 プロダクトコード 13

プロダクトアイテム詳細

プロダクトアイテム 6000010:13 CmContainer "<no name>" (3-4067252)

プロダクトアイテムオプション	タイプ	サイズ (バイト)	依存関係	値
使用期間		8	data, serial, counter	30日 - 開始時間: 2019-06-08 22:03:14

初めてプログラムを起動した時刻が開始時間になる

当然のことながら、使用開始時刻から30日を経過するとプログラムの起動ができなくなります。期間限定の評価版または使用期間の課金販売に利用すると非常に効果的です。また、使用期間を経過したCmStickに対し、後日コードメータのリモートアップデート機能を使って、使用期間の更新または解除(削除)を行うことが可能です。

3-9. メンテナンス期間を設定したプロテクトを行う

メンテナンス期間 (Maintenance Period)とは、ランセンサーがユーザーに対してサポートする期間を設定する機能です。メンテナンス期間のデータはCmStickの中にも書かれており、アプリケーションのリリース日と常に照合します。もし、リリース日がメンテナンス期間を過ぎていると、そのアプリケーションは起動しません。

この機能を利用することで、ライセンスは古いアプリケーションのサポートを自動的に終了させることができます。商品のサポート期間終了の設定や、ユーザーとのサポート契約、メンテナンス契約に役立つ機能です。

アプリケーションのリリース日は、AxProtectorで暗号化する際に設定します。また、このメンテナンス期間機能を利用するには、CmStick(CmDongle)のファームウェア Ver1.18以上が必要になります。

[NOTE]

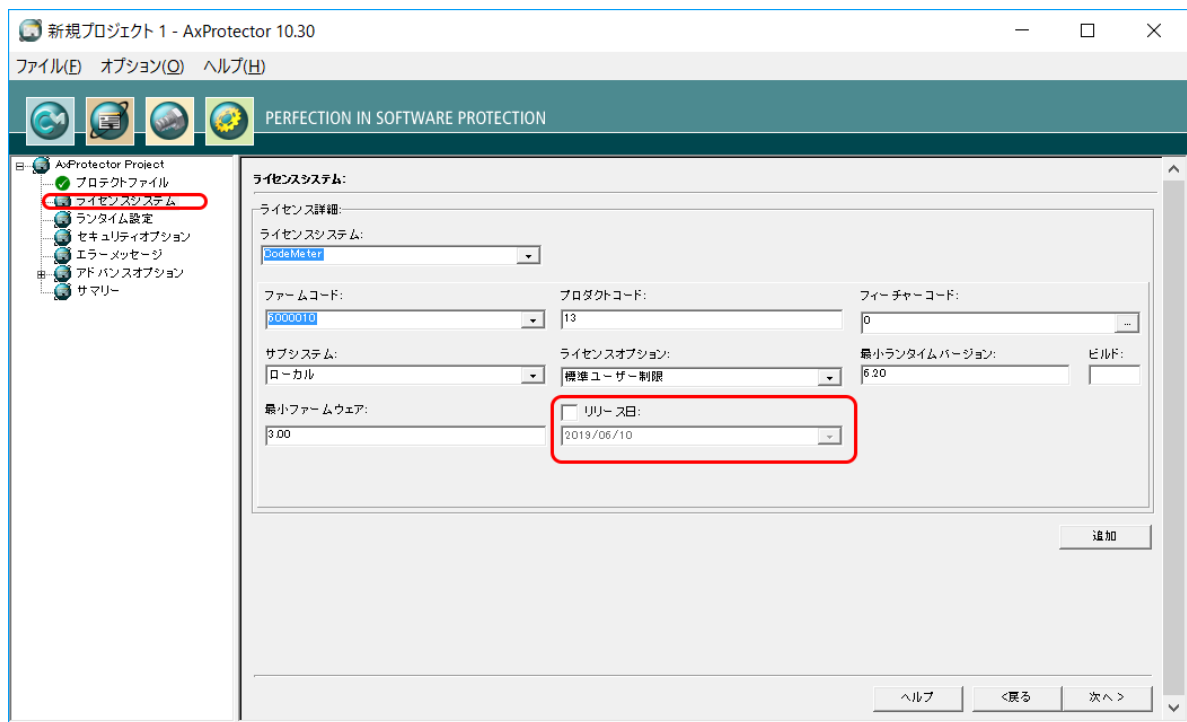
CmStick(CmDongle)のファームウェアは、コードメータコントロールセンターから更新することができます。

[NOTE]

アプリケーションのリリース日を設定したり、CmStick(CmDongle)にメンテナンス期間を登録するには、コードメータFSB(CmFSB)のファームウェアがVer1.18以上である必要があります。

① アプリケーションのリリース日を設定する

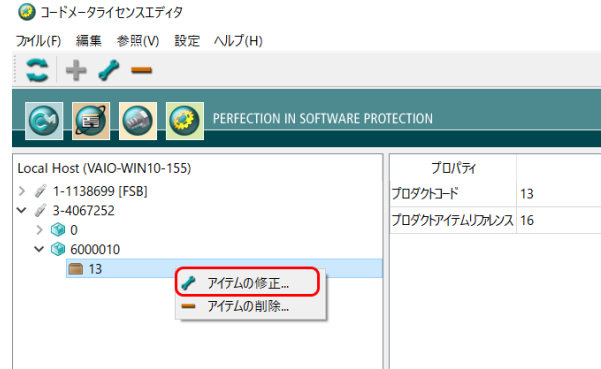
アプリケーションのリリース日は、自動暗号化ツールAxProtectorで暗号化する時に設定します。
(「ライセンスシステム」画面で設定する)



AxProtectorで暗号化されたプログラムは、内部にリリース日データを取り込んでおり、起動時にCmStick(CmDongle)内に保存されているメンテナンス期間と照合します。リリース日がメンテナンス期間内であればプログラムは起動し、メンテナンス期間外であれば起動しません。

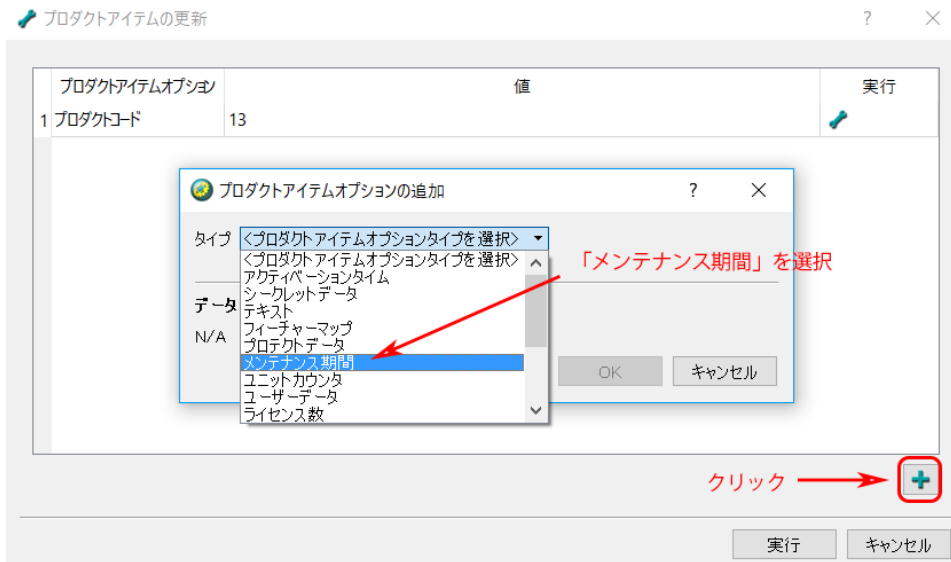
② CmStick にメンテナンス期間を登録する

CmStickにメンテナンス期間を登録するには、貴社のコードメータFSB(CmFSB)が必要になります。メンテナンス期間を登録するCmStickとコードメータFSBをPCに同時に装着し、コードメータライセンスエディタを起動します。ファームコード=6000010 / プロダクトコード=13にカーソルを合わせ右マウスをクリックして「アイテムの修正...」を選択します。

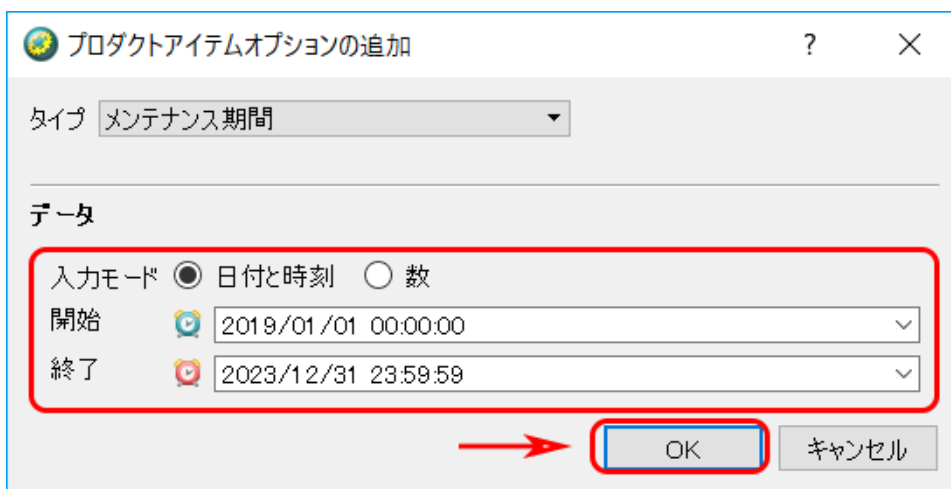


③ メンテナンス期間を設定する

「プロダクトアイテムの追加」画面で「メンテナンス期間」を選択します。

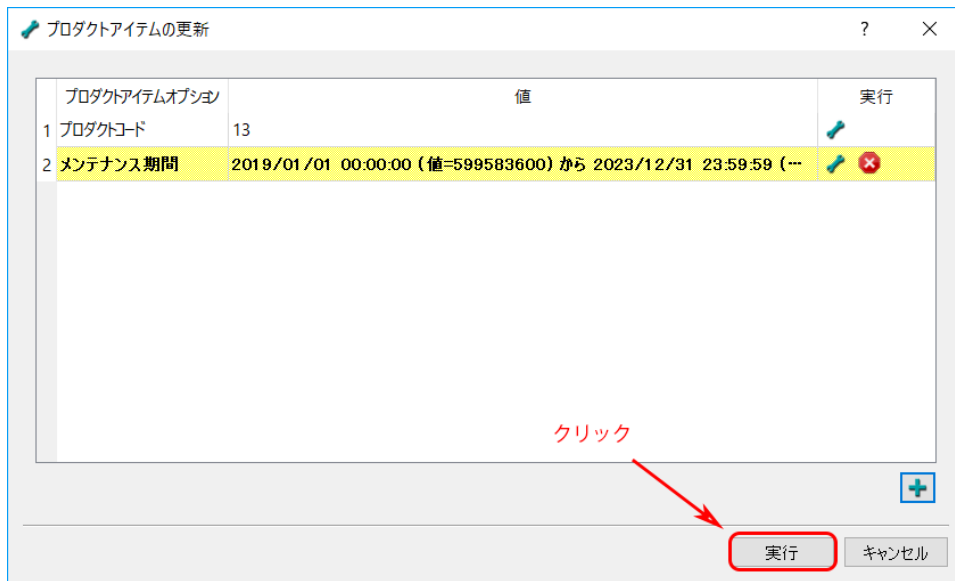


メンテナンス期間の「開始」と「終了」を入力し、「OK」ボタンをクリックします。



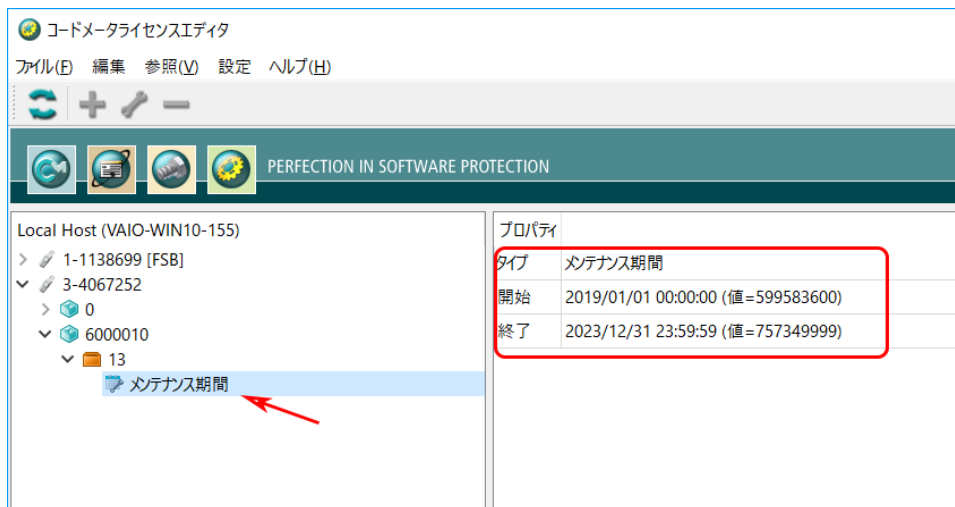
④ 「実行」 ボタンをクリックして登録する

「実行」ボタンをクリックしてCmStick にメンテナンス期間を登録します。



⑤ 登録内容を確認する

左ペインのプロダクトコード"13"には、新たにメンテナンス期間が追加されているのが確認できます。右ペインのプロパティには、「メンテナンス期間」に開始日と終了日が設定されているのが確認できます。



これで、メンテナンス期間：開始2019年1月1日、終了2023年12月31日というCmStickが作成されました。

⑥ Web アドミンからも確認する

Webアドミン上から、ファームコード=600010 / プロダクトコード=13 をクリックします。

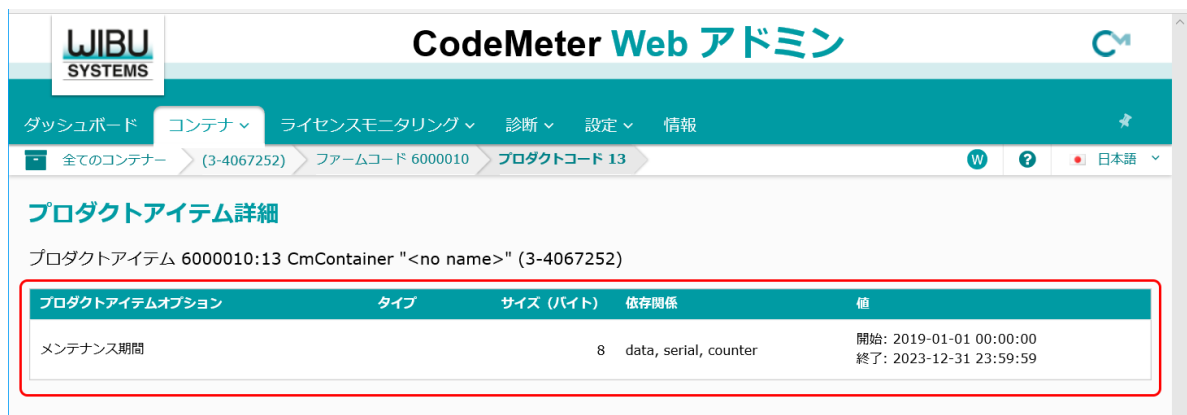


The screenshot shows the CodeMeter Web Admin interface. The breadcrumb navigation is: ホーム > 全てのコンテナ (3-4067252) > コンテナ. The main content area shows a table of containers. The first row is expanded to show details for container '6000010 Universal License Testkit'. A table below shows product items:

プロダクトコード	名前	ユニットカウンタ	有効期間	ライセンス数	フィーチャーマップ
13		n/a	n/a	1	n/a

The value '13' in the first column is highlighted with a red box, and a red arrow points to it with the text 'クリック'.

メンテナンス期間 (Maintenance Period) が登録されていることが確認できます。



The screenshot shows the CodeMeter Web Admin interface with the breadcrumb navigation: ホーム > 全てのコンテナ (3-4067252) > ファームコード 6000010 > プロダクトコード 13. The page title is 'プロダクトアイテム詳細'. The main content area shows the product item details for '6000010:13 CmContainer "<no name>" (3-4067252)'. A table below shows the maintenance period:

プロダクトアイテムオプション	タイプ	サイズ (バイト)	依存関係	値
メンテナンス期間		8	data, serial, counter	開始: 2019-01-01 00:00:00 終了: 2023-12-31 23:59:59

The table is highlighted with a red border.

3-10. プロテクトされたプログラムを起動する場合の注意点

コードメータでプロテクトされたプログラムが動作するためには、コードメータ・ランタイムキットがPCにインストールされている必要があります。今回プロテクト作業を行ったPCには、コードメータ開発キットをインストールした時点でコードメータ・ランタイムキットも同時にインストールされたため、特に単独でコードメータ・ランタイムキットをインストールする必要がありませんでした。プロテクトされたプログラムを別のPC上で起動するには、あらかじめコードメータ・ランタイムキットをインストールする必要があります。詳しくは、「Chapter 14 ユーザーに配布する場合」をご参照ください。

Chapter 4

自動暗号化ツール AxProtector について

- 4-1. 自動暗号化ツール AxProtector について
- 4-2. 日本語モードにする
- 4-3. AxProtector のメニュー画面
- 4-4. Windows アプリケーション EXE または DLL (32bit/64bit)
- 4-5. .NET アセンブリ
- 4-6. データファイルを暗号化する
- 4-7. コマンドラインでの使用方法

4-1. 自動暗号化ツール AxProtector について

コードメータには、ソースコードを変更せずに、貴社のEXEやDLLなどの実行形式プログラムを強力に暗号化する自動暗号化ツール「AxProtector」が用意されています。暗号化アルゴリズム128ビットAESを使い、プログラムを貴社のファームコード (Firm Code) / プロダクトコード (Product Code) を取り込みながら強力に自動暗号化します。

また、PCメモリー上で展開されるプログラムコードを常に暗号化し、必要な時に必要なモジュールを実行する「メモリー上のオンデマンド復号機能」を使用することができます。PCのハードディスク上や外部メディア上だけでなくPCのメモリー上でもプログラムコードが暗号化されているため、ハッキングに対する強力なセキュリティを実現することが可能になります。

AxProtectorの対象になるファイルは以下のとおりです。

1. Windowsアプリケーション(32/64) (EXE, DLL)
2. .NET アセンブリ
3. Linux Application/Shared Object
4. Mac OS X Application/Dylib
5. Javaアプリケーション (jar-file)



また、IxProtectorとして、

6. Windowsアプリケーション(32/64) (EXE, DLL)
7. .NETアセンブリ
8. Linux Application/Shared Object
9. Mac OS X Application/Dylib



その他、

10. ファイル暗号化

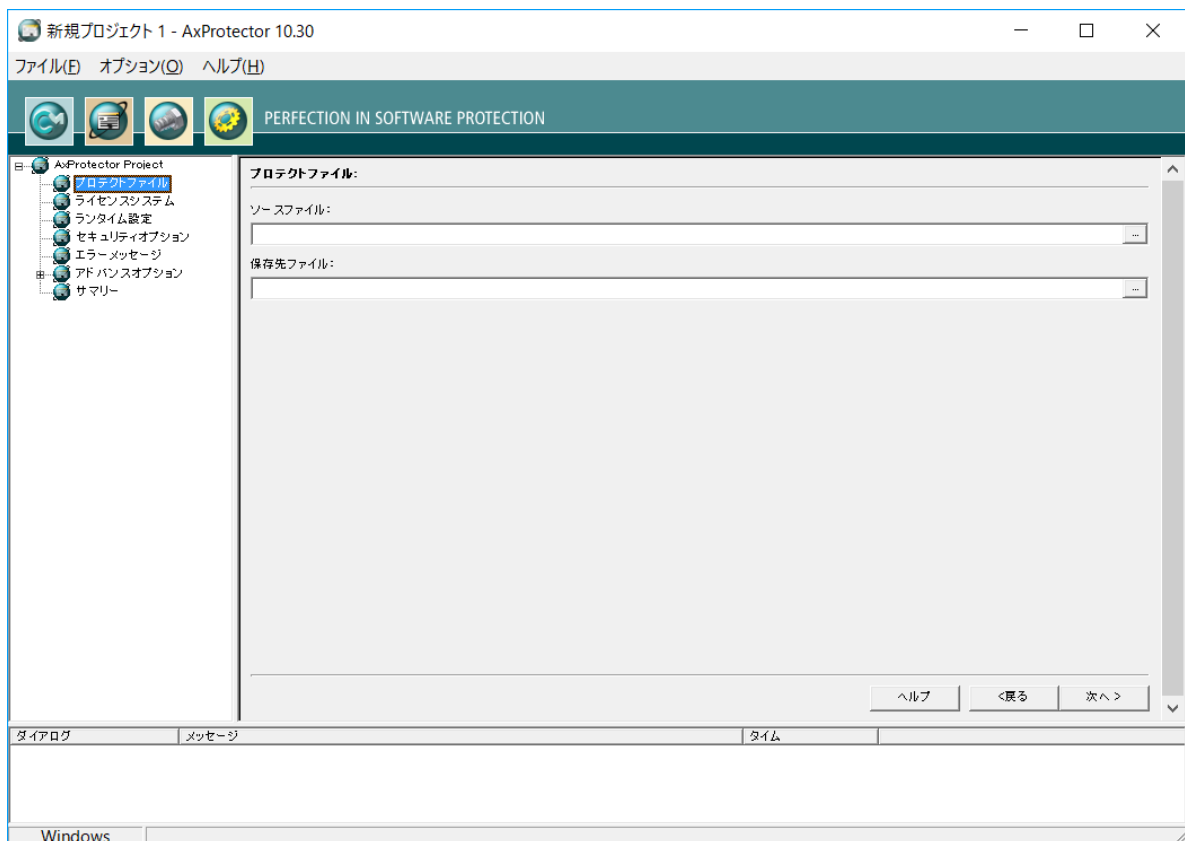
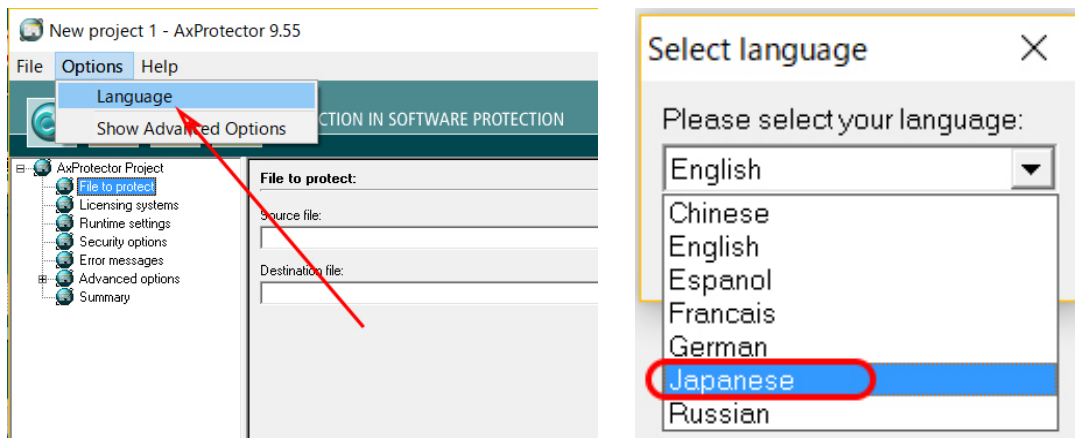
があります。

AxProtectorGui.exeは、¥Program Files (x86)
¥WIBU-SYSTEMS¥AxProtector¥DevKit¥binフォル
ダに存在します。



4-2. 日本語モードにする

はじめてAxProtectorを起動すると、英語モードで立ち上がります。[Options]-[Language]を選択し、「言語選択」画面で"Japanese"を選択し、OKをクリックすると日本語モードに変換されます。



4-3. AxProtector のメニュー画面

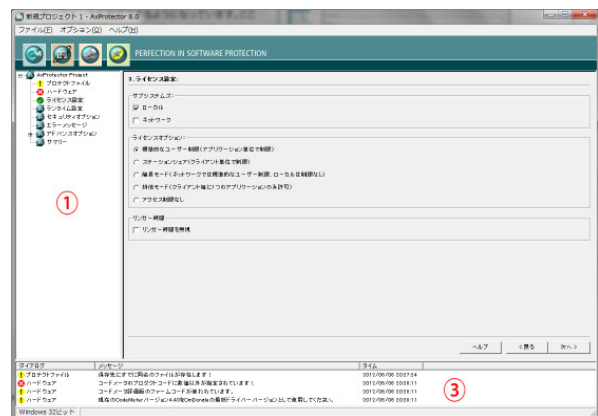
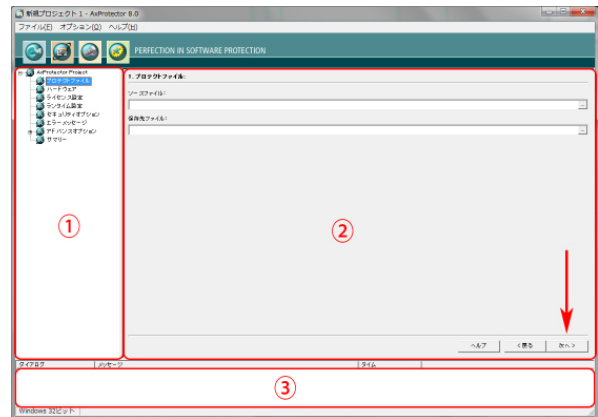
①のペインにはAxProtectorの操作項目が表示されます。各項目を上から順に実行すれば自動的に全てのパラメータを設定できるようになっています。ここで選択した項目に対するパラメータの値は②のペインで入力します。

②のペインで実際のパラメータを入力、設定します。入力が終了したら [次へ] ボタンをクリックすれば、次のプロテクト項目の設定画面に切り替わります。この操作は、①のペインで1つ下のプロテクト項目を選択したのと同じ動作になります。

③のペインには操作結果がリアルタイムで表示されます。

AxProtectorを操作した結果は、右図の様に①と③のペインにアイコンをともなってリアルタイムに表示されます。

なお、項目を選択した時の状況によっては、①と③のペインに、エラーや警告のアイコンが表示されることがありますが、そのまま先に進めて構いません。その後の操作で適切な値を設定することによってこれらのアイコンは表示されなくなります。最終的に、ファイルの暗号化を実行する時点でエラーや警告のアイコンが表示されなければ問題ありません。



アイコンの説明



… エラーが無く操作が行なわれたことをあらわします。

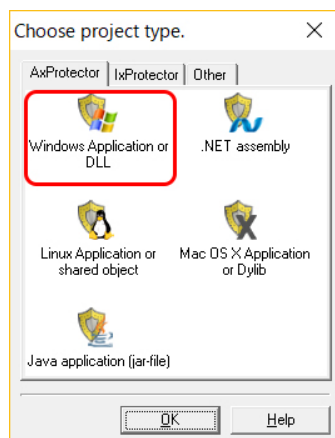


… 警告を表します。警告の内容によっては無視しても構わない場合がありますが、できるだけ警告を解消することをお勧め致します。



… エラーを表します。このアイコンが表示された項目に対して正しい設定をし直す必要があります。このアイコンが表示されていると暗号化処理時にエラーになります。

4-4. Windows アプリケーション EXE または DLL (32bit/64bit)



1. プロテクトファイル

暗号化する前のオリジナルファイル名と、暗号化生成されるファイル名を指定します。AxProtectorを起動した直後はこの画面が表示されます。

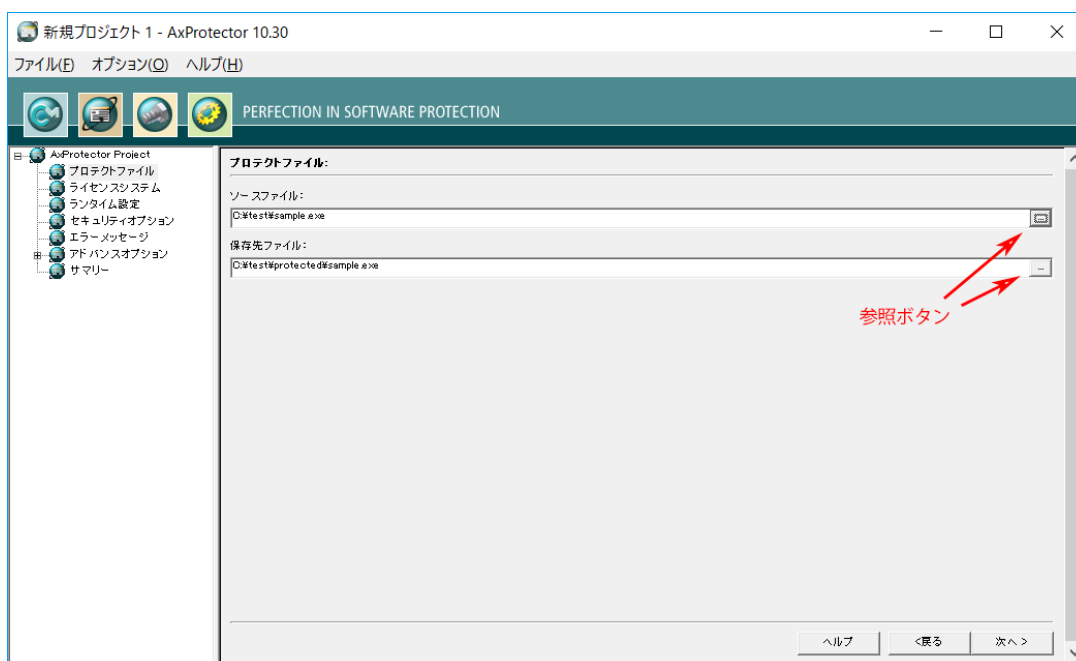
ソースファイル:

暗号化する前のオリジナルファイル名を指定します。右部の参照ボタンからファイルを指定することもできます。

保存先ファイル:

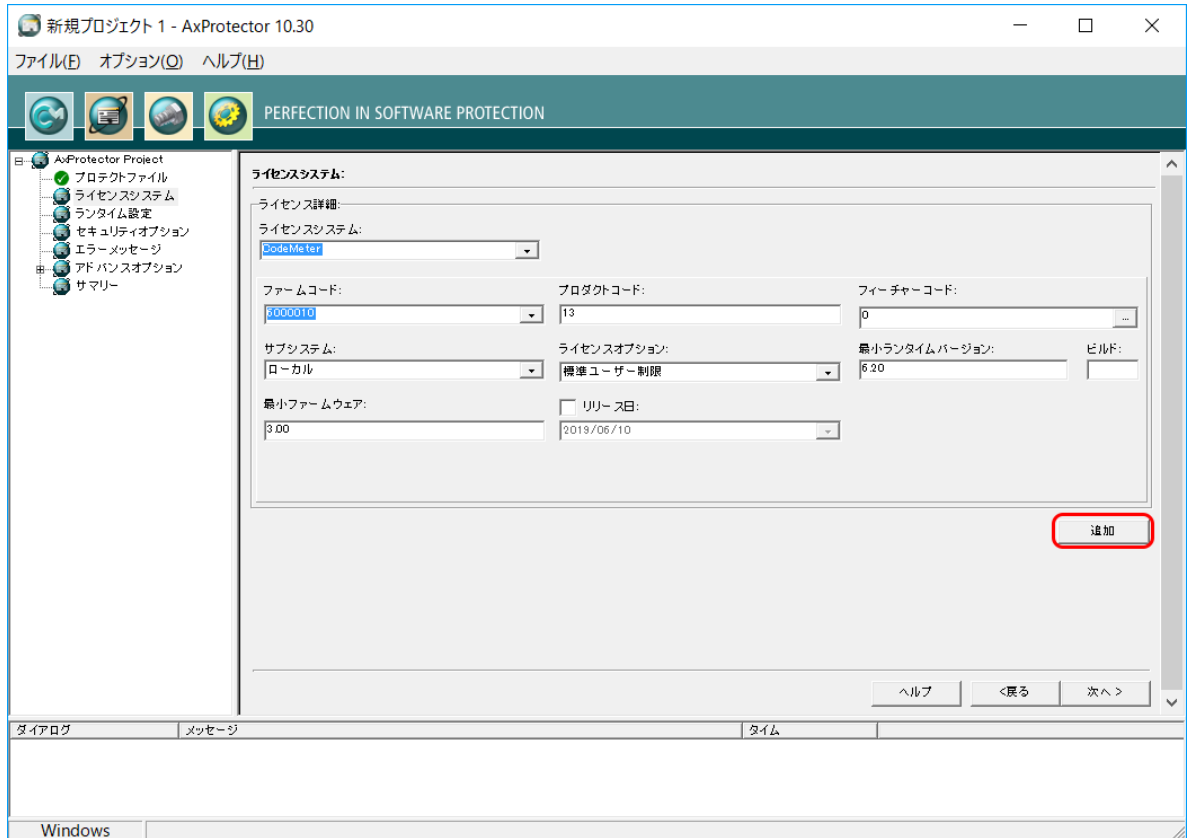
暗号化生成されるファイル名とその保存先のフォルダ名を指定します。右部の参照ボタンからファイルを指定することもできます。

ソースファイルと保存先ファイルを同一にすると、ソースファイルが暗号化生成されるファイルに上書きされますのでご注意ください。ファイル名が同じ場合は別フォルダに保存するか、同じフォルダに保存する場合は、ファイル名を変更するようにしてください。なお、ソースファイルを指定すると、同一フォルダの中にprotectedフォルダが自動的に作成されます。



2. ライセンスシステム

プロテクトに使用するライセンスシステムを設定します。AxProtectorはライセンスシステムとして、コードメータ (CmDongle)、コードメータAct (CmActLicense)、ワイブキー (WibuKey) の3種類を使用することができます。

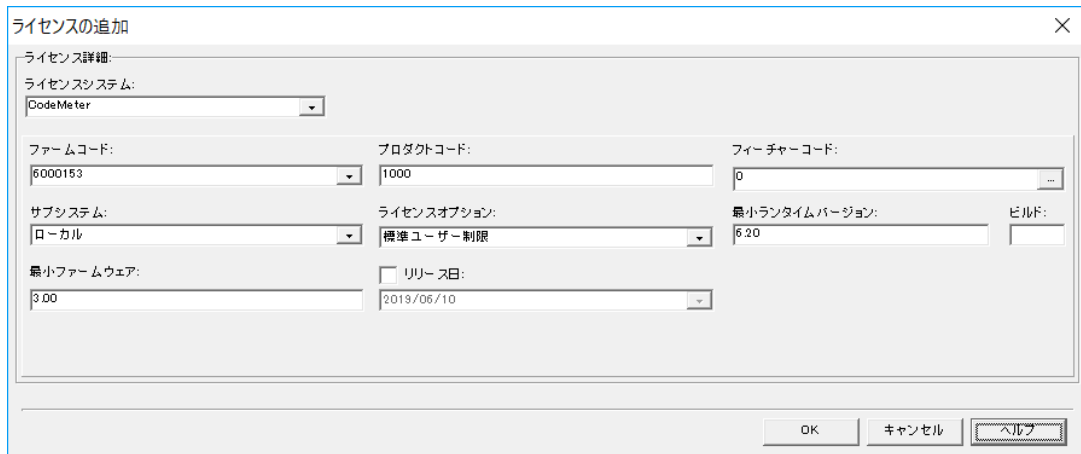


□ ライセンスシステム

リストボックスからCodeMeterとWibuKeyの選択ができます。ライセンスシステムにCmDongleおよびCmActLicenseを使用する場合はCodeMeterを選択します。

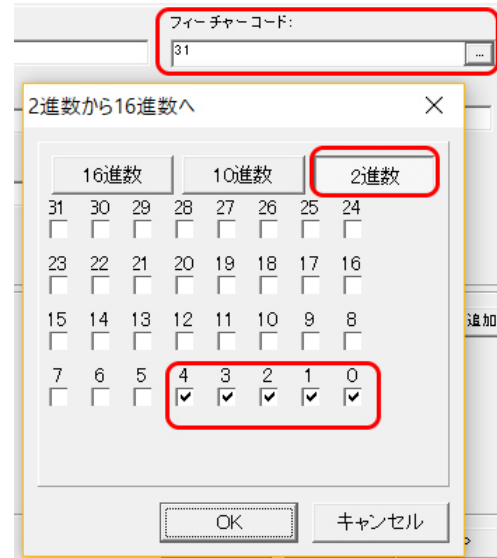
[NOTE]

「追加」ボタンをクリックして、異なるファームコードやプロダクトコードを持つCmDongleやCmActLicense、またはWibuKeyのライセンスシステムを追加することができます。生成されるEXEまたはDLLファイルは、ここで指定したライセンスが1つでも存在すると起動することができます。



また、後日、バージョン4(ファームコード=10、プロダクトコード=13、フィーチャーコード=16)をリリースした場合、CmDongle側のフィーチャーコードを"15"から"31"に変更することで、5つのバージョンを起動させることが可能になります。

0x00011111 = 31 (10進数)



CmDongle側のフィーチャーコードの変更(更新)は、コードメータのリモートアップデート機能を利用すると、メール添付のやりとりだけで実現できます。現物(ハードウェア)を送ったり、送り返したりする必要がありません。

□サブシステム:

暗号化されたプログラムが検索する範囲(サブシステム)を割り当てます。

ライセンスシステム:

ライセンスアイテム詳細:

ライセンスシステム:
CodeMeter

ファームコード: 10 プロダクトコード: 13 フィーチャーコード: 0

サブシステム:
ローカル
ローカル
ネットワーク
ローカル - ネットワーク
ネットワーク - ローカル

ライセンスオプション:
ユーザー制限なし

最小ドライババージョン: 6.10 Build:

リリース日:
2010/11/23

アイテムの追加

「ローカル」

ローカルのみアクセスする

「ネットワーク」

ネットワークのみアクセスする

「ローカル-ネットワーク」

最初にローカルアクセスを行い、次にネットワークアクセスを行う

「ネットワーク-ローカル」

最初にネットワークアクセスを行い、次にローカルアクセスを行う

□ライセンスオプション:

○ 標準ユーザー制限 (アプリケーション単位で制限) (Normal User Limit)

実行するアプリケーションごとに 1 つのライセンスを割り当てます。例えば、同じアプリケーションを同時に 2 回起動する場合は 2 つのライセンスが必要になります。この原則はコードメータキー (CmDongle) がローカル上にある場合もネットワーク (LAN) 上にある場合も同じように適用されます。(1 アプリケーション = 1 ライセンスの考え方)

○ ステーションシェア (クライアント単位で制限) (Station Share)

1 台の PC で同一のアプリケーションを同時に複数回起動した場合でも 1 ライセンスとして扱われます。(1 PC = 1 ライセンスの考え方)

○ WK 互換モード (ネットワーク上では標準ユーザー制限、ローカル上は制限なし)

ネットワーク (LAN) 上のコードメータキー (CmDongle) に対しては「標準ユーザー制限 (1 アプリケーション = 1 ライセンス)」として動作しますが、ローカル PC 上のコードメータキー (CmDongle) に対しては制限がありません。

* CmDongle をローカル PC に装着すると、その PC 上では同一のアプリケーションが無限に起動できるが、その CmDongle をコードメータサーバーに装着すると、「標準ユーザー制限 (1 アプリケーション = 1 ライセンス)」として機能する。

○ 排他モード (クライアントごとに 1 つのアプリケーションのみ許可) (Exclusive Mode)

1 台の PC で同一アプリケーションの重複起動を禁止します。

○ ユーザー制限なし (ユーザー数無制限) (No user Limit)

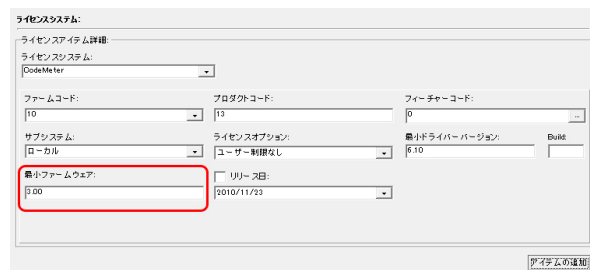
起動に必要なコードメータキー (CmDongle) がネットワーク (LAN) 上で見つければ、ライセンス数の制限にかかわらずアプリケーションが起動します。ライセンス数の制約を受けません。

□最小ドライバー (ランタイム) バージョン:

コードメータランタイムキットの最小バージョンを指定します。ここで指定したバージョンより古いバージョンがインストールされていると、暗号化されたプログラムは起動しません。コードメータランタイムキットは、OS 環境に応じて定期的にバージョンアップしており、貴社プログラムが古いバージョンで動作し、誤動作するのを防ぐことができます。

□最小ファームウェア:

CmDongleの要求ファームウェアの最小バージョンを指定します。CmDongle側のファームウェアが、ここで設定した最小ファームウェアのバージョンより低い場合は、暗号化されたプログラムは起動しません。



[注意]

デフォルトでは最小ファームウェアが"3.00"に設定されますが、従来からのCmDongleと互換性をもたせるためには最小ファームウェアを"1.14"または"1.18" (「リリース日」機能を使用する場合) にすることをお勧め致します。

[NOTE]

Universal FirmCode=60xxxxxを使用する場合は、ファームウェアバージョンが3.00以上必要になります。FirmCode=10xxxxの場合、ファームウェアバージョン1.14以降で動作します。

[NOTE]

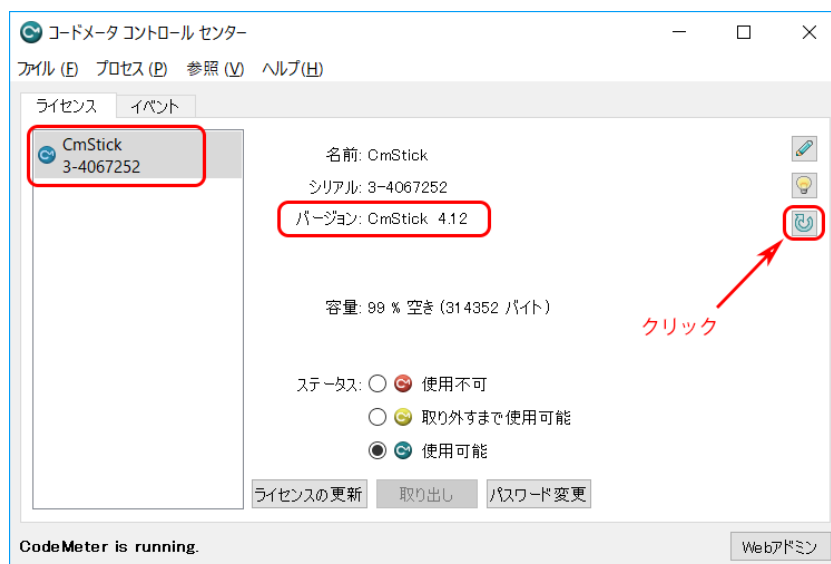
CmDongleファームウェアのバージョンアップは、コードメータコントロールセンター上でユーザーが更新することができます。ただし、CmDongleのシリアル番号によって、最大可能なバージョンが異なります。

シリアル番号 1-xxxxxxxのCmDongleはファームウェア1.18まで可能。

シリアル番号 2-xxxxxxxのCmDongleはファームウェア2.04まで可能。

シリアル番号 3-xxxxxxxのCmDongleはファームウェア3.00以降可能。

CmDongleのファームウェアの更新は、コードメータコントロールセンター画面の「ファームウェアの更新」ボタンをクリックすると自動的に最新バージョンに更新されます。更新作業は、インターネットに接続する必要があります。



□リリース日:

プログラムのリリース日を設定します。リリース日は、CmContainer内に登録されたメンテナンス期間と連動し、リリース日がメンテナンス期間内であればプログラムが起動し、メンテナンス期間以外の場合はプログラムが起動しません。

この機能は、ユーザーとのプログラムのメンテナンス契約をコントロールするために役立つ機能です。

[注意]

リリース日機能を利用するには、最小ファームウェアは1.18以上が必要になります。

□「追加」ボタンについて

複数のライセンスシステムを追加する場合は、「追加」ボタンをクリックしてライセンスシステムを追加します。

3. ランタイム設定

使用制限機能に関する動作を設定します。



ランタイムチェック：

ランタイムチェック(コードメータキーの定期的チェック)のインターバル時間を設定します。

ランタイムチェックを有効

ランタイムチェック機能を有効にします。

インターバル時間(時:分:秒)：

ランタイムチェックにおいてコードメータキーのチェックが行なわれてから次のチェックが行なわれるまでのインターバルを時分秒で設定します。デフォルトは30秒(00:00:30)が設定されています。

エラー許容回数：

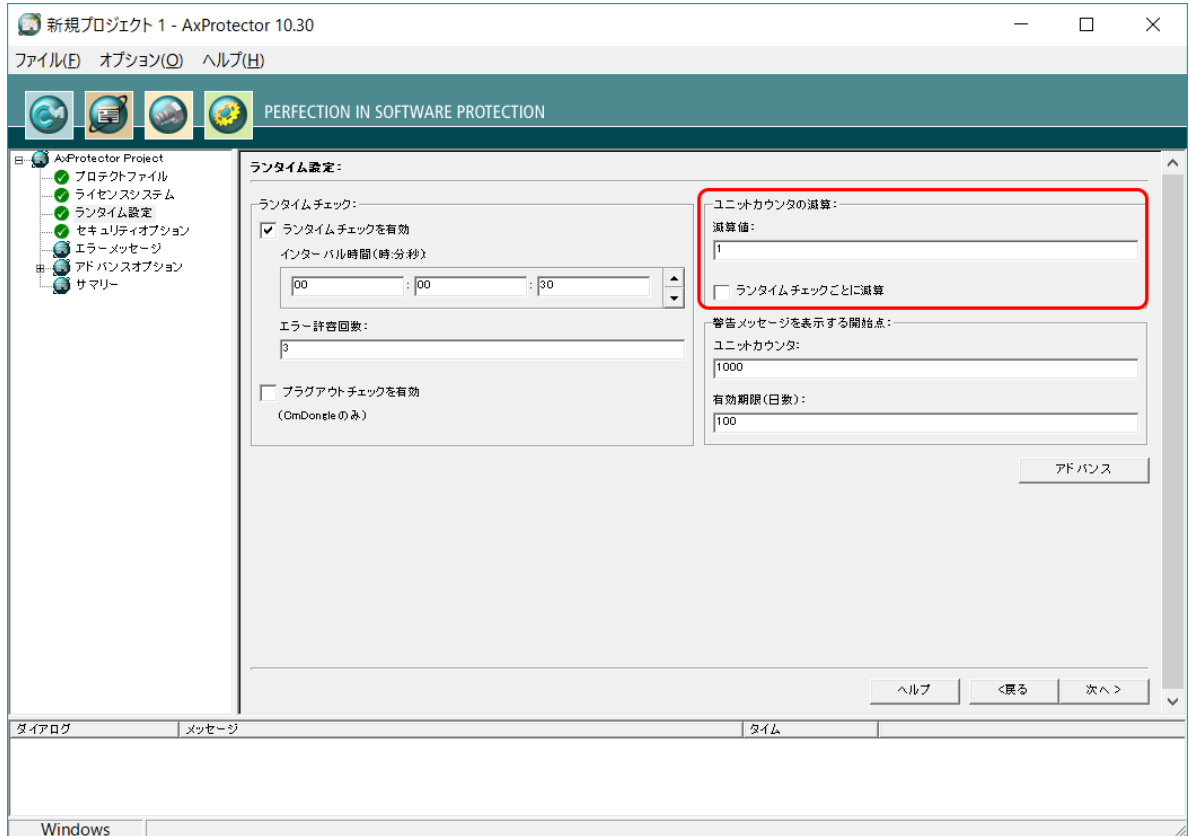
この機能は、何らかの理由でコードメータキーのランタイムチェックが失敗した場合(チェックエラーの場合)、ただちにアプリケーションを終了させず、許可した回数だけアプリケーションを続行させる機能です。例えば、ここで3(デフォルト)を設定すると、ランタイムチェックエラーになっても3回まではアプリケーションを続行させることが可能です。しかし、4回目でエラーになるとアプリケーションを終了させます。その際、強制的にアプリケーションを終了させるのではなく、「中止」か「再試行」の選択メッセージを出します。コードメータキーを装着して「再試行」をクリックすると、アプリケーションは続行します。コードメータキーを装着せずに「中止」をクリックするとアプリケーションは終了します。この機能を使わず、1回目のランタイムチェックエラーでアプリケーションを終了させたい場合は0を設定します。

プラグアウトチェックを有効(CmDongleのみ)

コードメータキーをPCから取り外すとすぐにエラーを表示させます。「プラグアウト」を監視する。

ユニットカウンタの減算:

コードメタキーのユニットカウンタの減算値を設定します。ユニットカウンタを使用すると、コードメタキーのチェックが行なわれるたびにユニットカウンタチェックが行なわれ、「減算値」で設定した数値が減算されます。ユニットカウンタが0(ゼロ)になるとアプリケーションの起動ができなくなります。



減算値:

1回のチェックで減算する値を設定します。デフォルト値は"1"です。

ランタイムチェックごとに減算

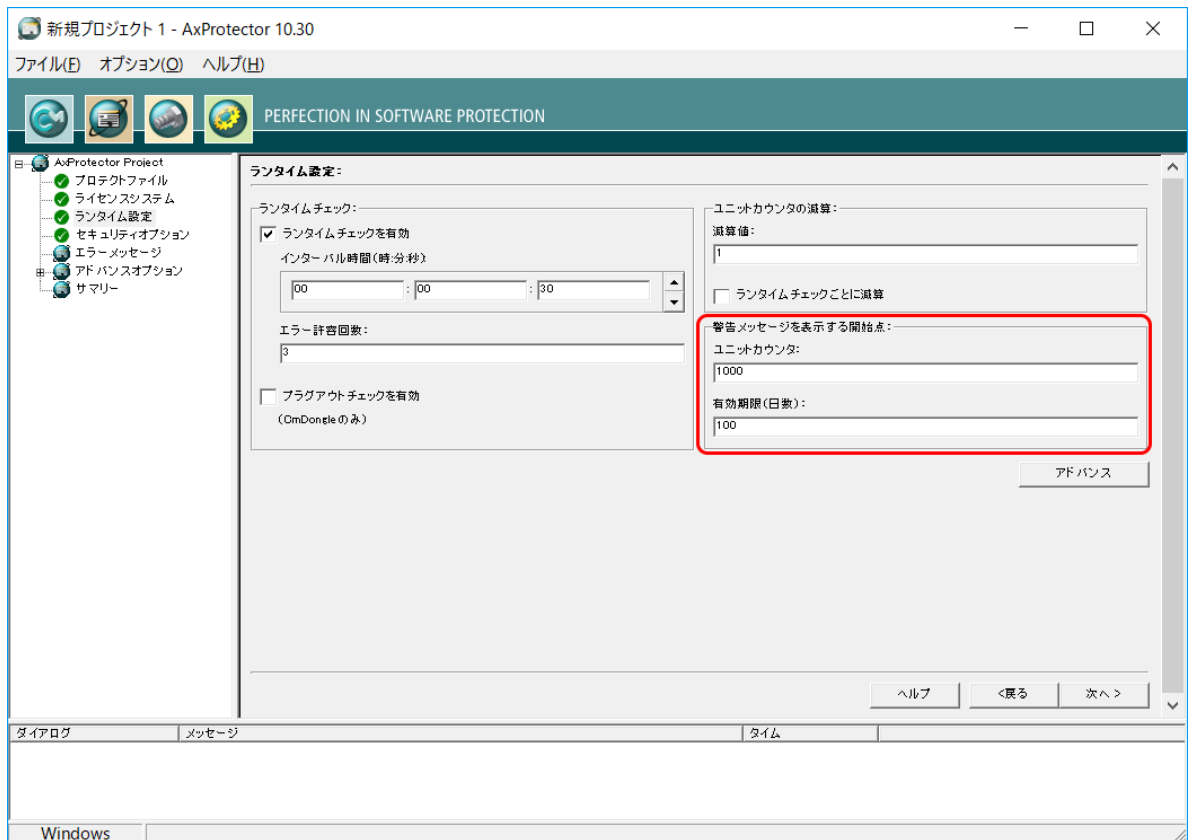
ランタイムチェックが行なわれるごとにユニットカウンタの減算を行いません。ランタイムチェック時に減算を行わない場合はチェックをはずします。この機能はランタイムチェックのインターバル時間と組み合わせることで、プログラムの使用可能時間を設定することができます。

[NOTE]

プログラムの使用可能時間は、使用期間 (Usage Period)機能でも設定できます。

警告メッセージを表示する開始点：

ユニットカウンタや有効期限(日数)の残りが一定の値を下回った場合に警告メッセージを表示させるオプションです。



□ ユニットカウンタ：

ユニットカウンタがあと残りいくつになると警告メッセージを出すかの開始点を設定します。警告メッセージは開始点から0になるまでアプリケーションを起動するたびに表示されます。出力される警告メッセージは、ユニットカウンタ警告用メッセージです。

(デフォルト値は1000回：ユニットカウンタが1000になると警告メッセージを表示し始める)

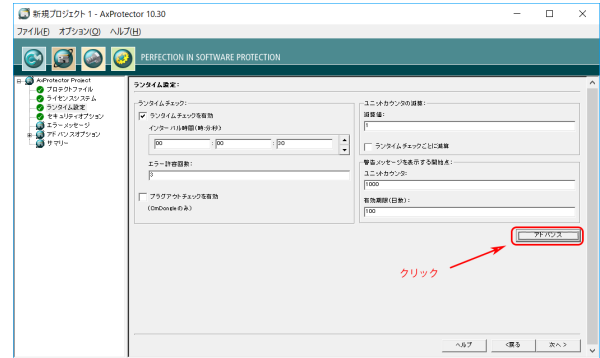
□ 有効期限(日数)：

有効期限まであと何日になると警告メッセージを出すかの開始点を設定します。警告メッセージは有効期限に達するまで、アプリケーションを起動するたびに表示されます。出力される警告メッセージは有効期限警告用メッセージです。

(デフォルト値は100日：有効期限(日数)が100日になると警告メッセージを表示し始める)

アドバンス:

[アドバンス]ボタンをクリックすると「ランタイムの設定」(アドバンス設定)画面が開きます。



アドバンスランタイム設定 ✕

<p>ユニットカウンタチェック:</p> <p><input checked="" type="radio"/> 標準</p> <p><input type="radio"/> 要求 (CodeMeterのみ)</p> <p><input type="radio"/> 無視 (CodeMeterのみ)</p>	<p>有効期限チェック:</p> <p><input checked="" type="radio"/> 標準</p> <p><input type="radio"/> 要求 (CodeMeterのみ)</p> <p><input type="radio"/> 無視 (CodeMeterのみ)</p>	<p>アクティベーションタイムチェック (CodeMeterのみ)</p> <p><input checked="" type="radio"/> 標準</p> <p><input type="radio"/> 要求</p> <p><input type="radio"/> 無視</p>
<p>システムタイムチェック (CodeMeterのみ):</p> <p><input type="checkbox"/> 暗号化タイムチェック</p> <p><input type="checkbox"/> CmContainer / PCシステムタイムチェック</p> <p>過去に対する許容時間(分): <input type="text" value="15"/></p> <p>将来に対する許容時間(分): <input type="text" value="15"/></p>	<p>メンテナンス期間 チェック (CodeMeterのみ)</p> <p><input checked="" type="radio"/> 標準</p> <p><input type="radio"/> 要求</p>	
<p>認証時間 (CodeMeterのみ)</p> <p><input type="checkbox"/> 認証時間をセット</p> <p><input type="checkbox"/> 認証時間のチェック</p> <p>認証時間の有効期間(時間) <input type="text" value="100"/></p> <p>タイムチェック無しの期間(時間) <input type="text" value="0"/></p>	<p>アドバンスオプション:</p> <p><input type="checkbox"/> ControlとAboutをメニュー追加</p> <p><input checked="" type="checkbox"/> ホストアプリケーションを終了</p> <p><input type="checkbox"/> モバイルアプリケーション生成</p>	

ユニットカウンタチェック:**○標準**

コードメータキー (CmContainer) にユニットカウンタが設定されている場合は、ユニットカウンタ値をチェックし、設定されていない場合は無視する。(推奨)

○要求 (CodeMeterのみ)

ユニットカウンタを必ずチェックする。もし、コードメータキー (CmContainer) にユニットカウンタが設定されていない場合は、プログラムを起動しない。

○無視 (CodeMeterのみ)

ユニットカウンタを無視する。コードメータキー (CmContainer) にユニットカウンタが設定されていても、プログラムは無視する。実質上、ユニットカウンタ機能なしの状態になる。

有効期限チェック:

○標準

コードメータキー (CmContainer) に有効期限が設定されている場合は、有効期限をチェックし、設定されていない場合は無視する。(推奨)

○要求(CodeMeterのみ)

有効期限を必ずチェックする。もし、コードメータキー (CmContainer) に有効期限が設定されていない場合は、プログラムを起動しない。

○無視 (CodeMeterのみ)

有効期限を無視する。コードメータキー (CmContainer) に有効期限が設定されていても、プログラムは無視する。実質上、有効期限なしの状態になる。

アクティベーションチェック (CodeMeterのみ):

○標準

コードメータキー (CmContainer) にアクティベーションタイム (使用開始日) が設定されている場合は、アクティベーションタイムをチェックし、設定されていない場合は無視する。(推奨)

○要求

アクティベーションタイムを必ずチェックする。もし、コードメータキー (CmContainer) にアクティベーションタイムが設定されていない場合は、プログラムを起動しない。

○無視

アクティベーションタイムを無視する。コードメータキー (CmContainer) にアクティベーションタイムが設定されていても、プログラムは無視する。実質上、アクティベーションタイムなしの状態。

システムタイムチェック (CodeMeterのみ):

暗号化タイムチェック

プログラム暗号化時のPC時刻 (PCシステムタイム) を記憶し、プログラム起動時に、CmDongle (CmContainer) 内のシステムタイム (CmContainer) と比較する。システムタイム (CmContainer) の方が、プログラム暗号化時のPC時刻より進んでいる (新しい) 場合のみ起動する。

システムタイムチェック (CodeMeterのみ):

- 暗号化タイムチェック
- CmContainer / PCシステムタイムチェック

過去に対する許容時間(分):
15

将来に対する許容時間(分):
15

CmContainer/PCシステムタイムチェック

システムタイム (CmContainer) とPCシステムタイムの差を時間 (Minutes = 分) で設定する。

過去に対する許容時間(分)

PCシステムタイムがシステムタイム (CmContainer) よりどれだけ遅れている (古い) かを分単位で設定する。設定した時間 (分) 以内であれば暗号化したプログラムは起動し、それ以上遅れている場合は起動しない。

将来に対する許容時間(分)

PCシステムタイムがシステムタイム(CmContainer)よりどれだけ進んでいる(早い)かを分単位で設定する。設定した時間(分)以内であれば暗号化したプログラムは起動し、それ以上進んでいる場合は起動しない。

[NOTE]

CmContainerとは、CmDongleとCmActLicenseを総称した表現です。

メンテナンス期間チェック (CodeMeterのみ)：**○ 標準**

コードメータキー(CmContainer)にメンテナンス期間が設定されている場合は、メンテナンス期間をチェックし、設定されていない場合は無視する。(推奨)

○ 要求

メンテナンス期間を必ずチェックする。もし、コードメータキー(CmContainer)にメンテナンス期間が設定されていない場合は、プログラムを起動しない。

認証時間 (CodeMeterのみ)：**□ 認証時間をセット**

このオプションにチェックを入れると、プログラムが起動するたびにコードメータタイムサーバーにアクセスし、コードメータキー(CmContainer)の認証時間とボックス時間を更新します。

□ 認証時間のチェック

このオプションにチェックを入れると、プログラムが起動するたびにコードメータタイムサーバーにアクセスし現在の正確な時刻をチェックし、コードメータキー(CmContainer)の"システムタイム(CmContainer)"と比較します。

認証時間の有効期間(時間)

現在の時刻とコードメータキー(CmContainer)のシステムタイムの差が、「認証時間の有効時間(時間)」で設定された数値(時間)以上に差がある場合はプログラムの起動を中止します。

タイムチェック無しの期間(時間)

指定した数値(時間)の期間、認証時間のチェックを行いません。

アドバンスオプション:

ControlとAboutをメニュー追加

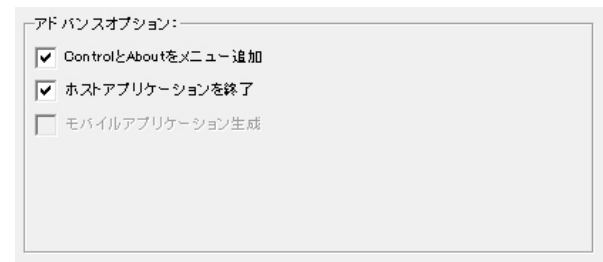
システムメニューとAboutメニューをアプリケーションに追加します。このオプションは、ランタイムチェックが実行されているときに有効です。

ホストアプリケーションを終了

暗号化するファイルがDLLファイルの場合、DLLファイルがプロテクトチェックに失敗すると、そのDLLファイルを使用しているホストアプリケーションが終了します。デフォルトではチェックが入っています。(推奨)

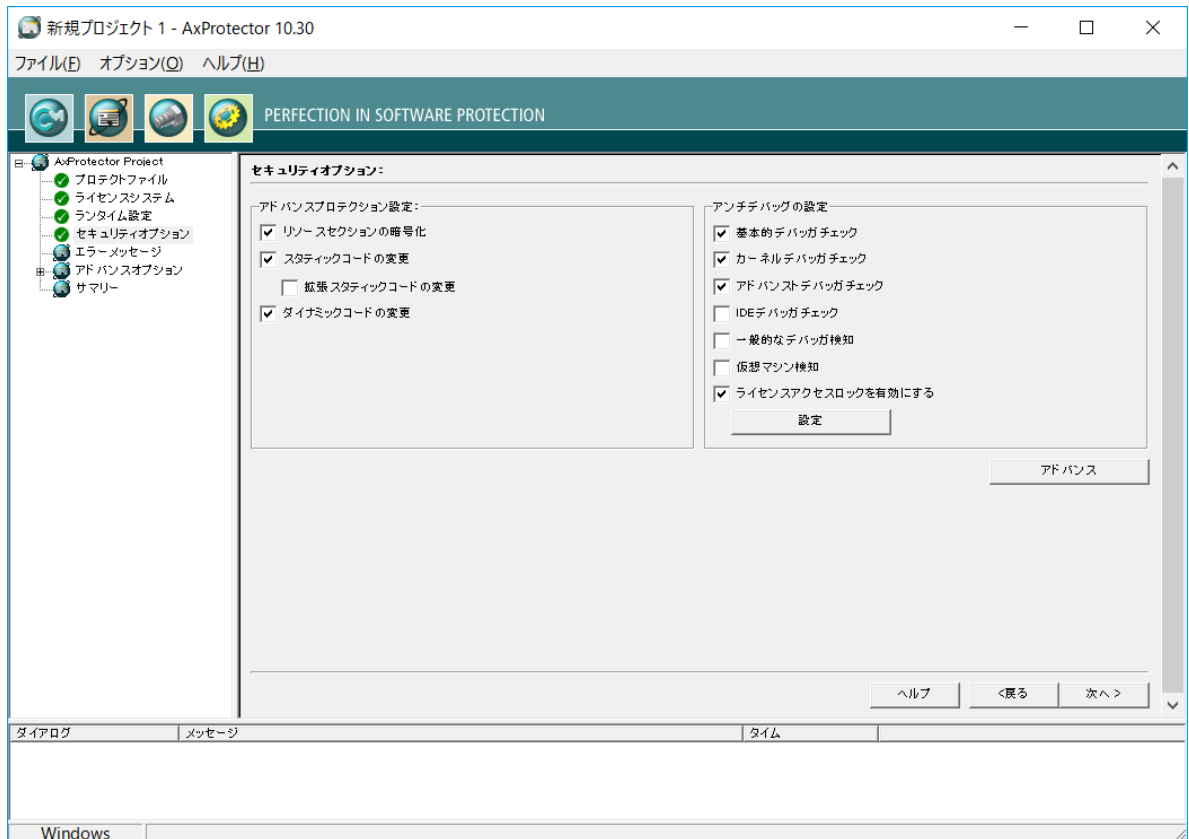
モバイルアプリケーション生成

現在未使用。



4. セキュリティオプション

このセキュリティオプションは、プロテクト強度に関連する設定です。暗号化に対する解析強度を強化する場合に使用します。



アドバンスプロテクション設定：

高度なプロテクション方法を指定します。解析強度を重視する場合はすべてにチェックを入れてください。(ただし、「スタティックコードの変更」と「拡張スタティックコードの変更」は同時に指定できません)

リソースセクションの暗号化

リソースを暗号化します。

スタティックコードの変更

デバッグ、ダンプ、リバースエンジニアリングを不可能にするために、正常にコンパイルされたコードを独自の方法で変更し直します。

拡張スタティックコードの変更

「スタティックコードの変更」の拡張版です。「スタティックコードの変更」に加え、多重のセキュリティメカニズムを追加します。

ダイナミックコードの変更

アプリケーションが実行されている間、コードを変更します。

[NOTE]

「スタティックコードの変更」と「拡張スタティックコードの変更」をアドバンスオプションの「(データ)ファイルの暗号化を有効にする」で同時に使用することはできません。

アンチデバッグの設定

デバッガによる解析を防止する機能です。デフォルト設定で十分にセキュリティ機能が実現できますが、より強度を重視する場合は必要に応じてチェックを追加してください。

□基本的デバッガチェック

一般的なデバッグプログラムを検知します。デバッグプログラムが見つかった場合、アプリケーションは起動しません。

□カーネルデバッガチェック

「SoftICE」のようなカーネルデバッグプログラムを検知します。カーネルデバッグプログラムが見つかった場合、アプリケーションは起動しません。

□アドバンスドデバッガチェック

デバッグプログラムの検知をより強化します。デバッグプログラムが見つかった場合、アプリケーションは起動しません。もし、アプリケーション起動中にデバッグプログラムが検知された場合は、アプリケーションを終了させます。

□IDE デバッガチェック

Visual StudioやDelphiなどの統合開発環境(IDE)のデバッガを禁止します。もし見つかった場合は、アプリケーションが起動しません。

□一般的なデバッガ検知

デバッガがアプリケーションにアタッチされないようなメカニズムを追加します。

□仮想マシン検知

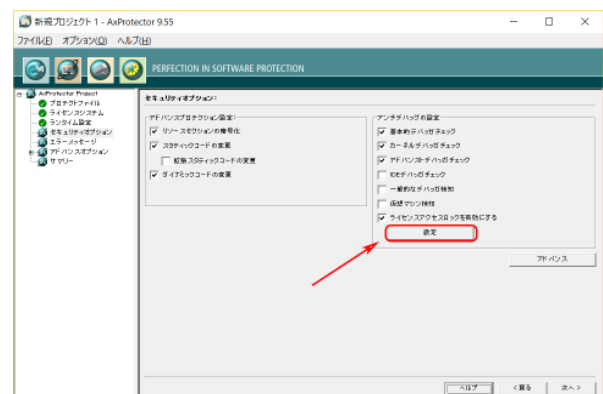
アプリケーションがバーチャルマシン上で起動できないようにします。

□ライセンスアクセスロックを有効にする

デバッガの動作を検知した場合に、コードメータキー (CmDongle) 自身をロックし、アプリケーションが起動できないようにします。コードメータキー (CmDongle) のロックを解除するには、ライセンサーによるリモートプログラミング(更新ファイル)が必要になります。

ファームウェアバージョン1.14以降の場合、デバッガが検知されたら直ちにCmDongleをロックします。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)は、「用意されたロック」が行われます。「用意されたロック」とは、CmDongleを直ちにロックするのではなく、ファームアクセスカウンタ (FAC=Firm Access Counter) を利用し、デバッガを検知するたびに値を"1"ずつ減らし、最終的に"0"になった時点でファームアイテム(Firm Item)がロックします。初期設定では、数値"65535"が割り当てられていますが、ソフトウェアベンダーによって変更することが可能です。ロックされたFirm Itemは、ソフトウェアベンダーのリモートプログラムによって解除することができます。

「設定」ボタンをクリックして、ライセンスアクセスロックモードを選択することができます。



○自動モード

ファームウェアバージョンが1.14以降の場合、ファームアイテムは直ちにロックされます。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)、FAC (Firm Access Counter) の値が"1"減ります。(デフォルト:推奨)

□FACが定義されている場合のみロックする

このオプションにチェックを入れると、ファームウェアバージョンが1.14以降の場合、「用意されたロック」が設定されているかどうかをチェックし、設定されていた場合にファームアイテムをロックします。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)、FACの値が 1 減ります。

○ライセンスコンテナをロック(new)

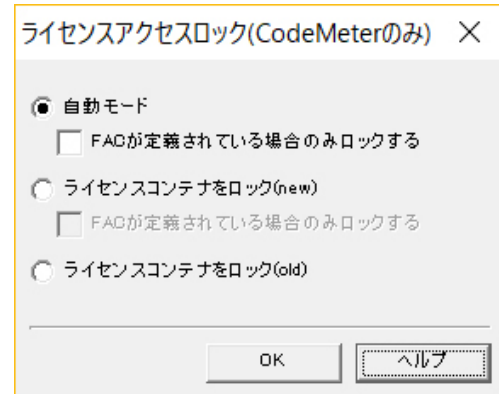
ファームウェアバージョンが1.14以降の場合、ファームアイテムが直ちにロックされます。このオプションがセキュリティ上一番望ましいのですが、必ずファームウェアは1.14以上が必要になります。

□FACが定義されている場合のみロックする

このオプションにチェックを入れると、ファームウェアバージョンが1.14以降の場合、ファームアイテムを直ちにロックします。同時に、「用意されたロック」が設定されているかどうかをチェックし、用意されていた場合、ファームアイテムをロックします。

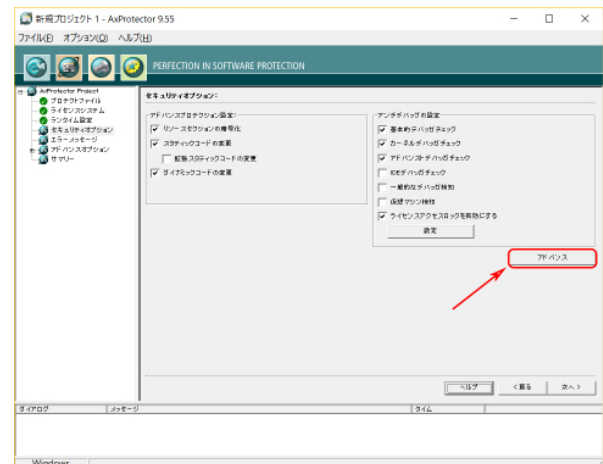
○ライセンスコンテナをロック(old)

すべてのファームウェアバージョンが対象です。
「用意されたロック」が設定されていた場合、FACの値が1減ります。

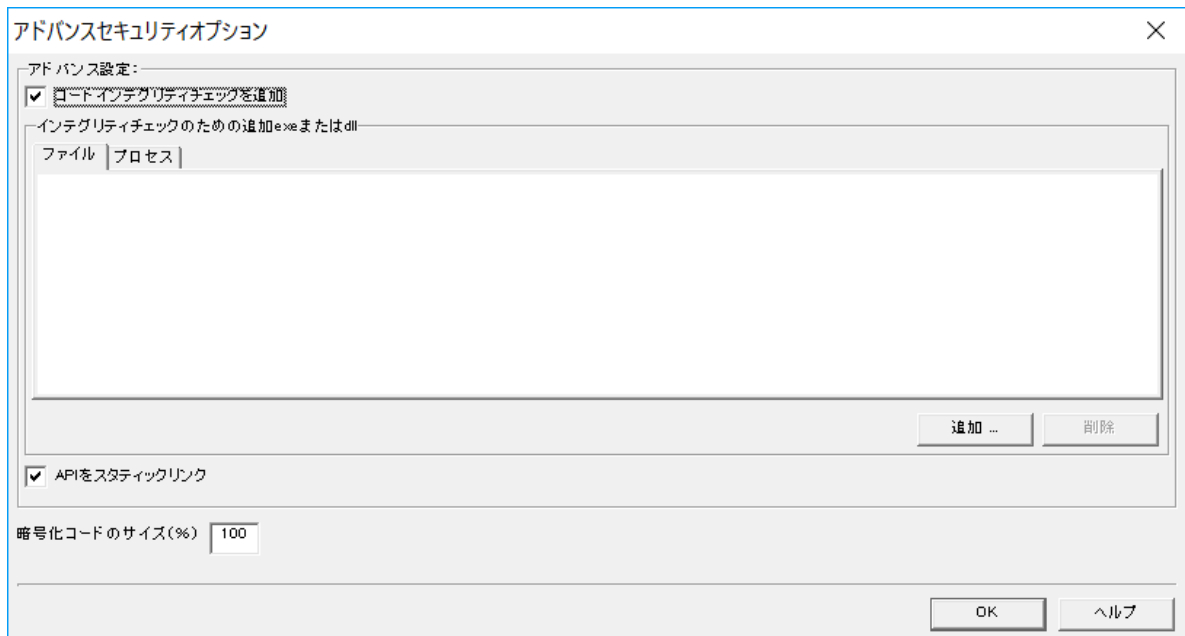


アドバンスセキュリティオプション

[アドバンス]ボタンをクリックするとアドバンスセキュリティオプション設定画面が開きます。



アドバンス設定:

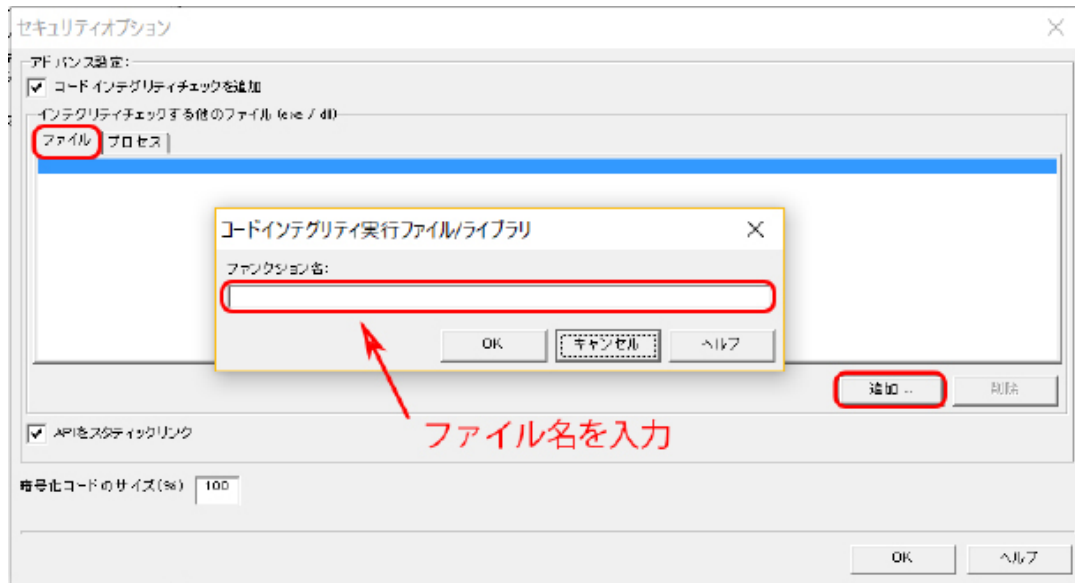


コードインテグリティチェックを追加

非対称認証を利用したコードインテグリティチェックを行います。

チェックを入れると(デフォルト)、アプリケーションプログラムのチェックサム(ハッシュ値)がプライベートキーにより作成および署名され、ハッシュ値とシグニチャがアプリケーションに付加されます。ランタイムチェック時に、AxEngine(ソフトウェア)に存在するパブリックキーを使って、アプリケーションのハッシュ値を再計算し、インテグリティチェックを行います。

さらに、複数のEXEファイルやDLLライブラリを含めたコードインテグリティチェックを行うことも可能です。1つのEXEファイルが、他の関連するEXEファイルやDLLライブラリのインテグリティチェックを行います。



APIをスタティックリンク

APIをスタティックリンクします。ファイルサイズは大きくなりますが、セキュリティ強度は増します。

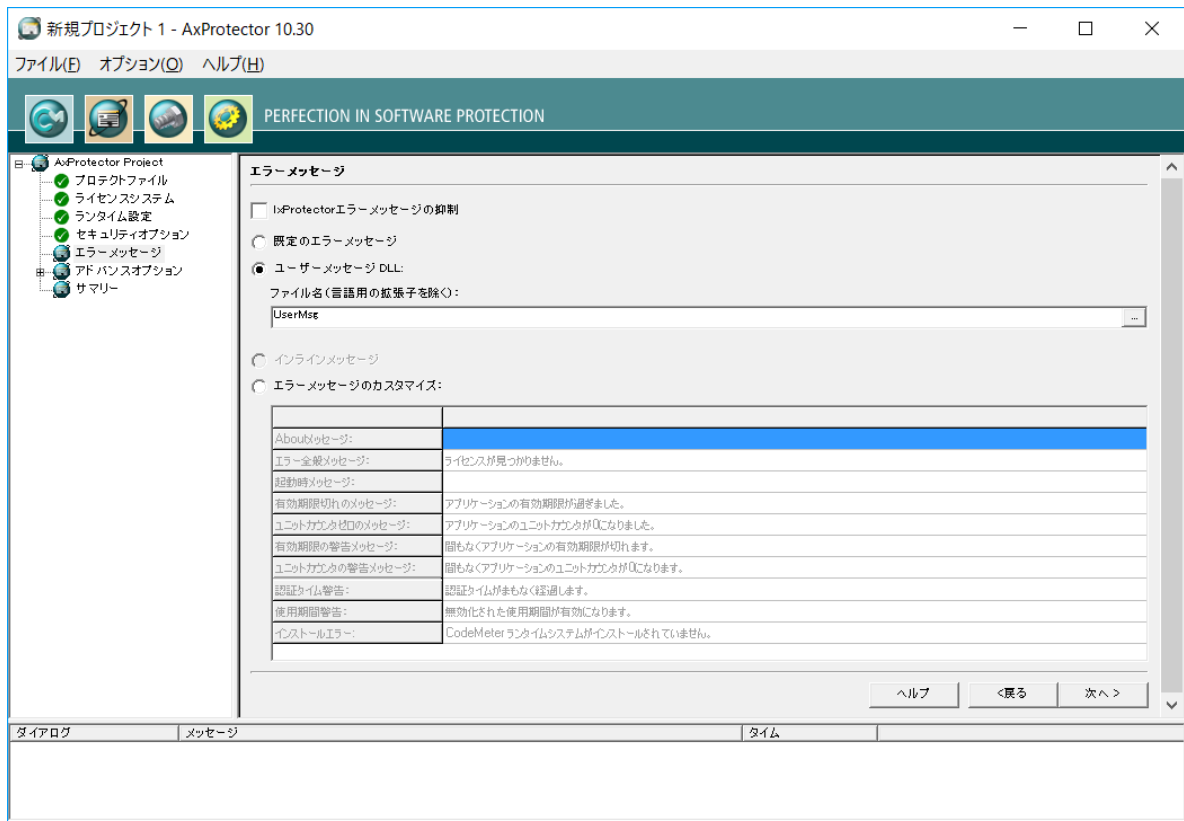
暗号化コードのサイズ(%):

暗号化するコードサイズを%にて設定します。(100%推奨)

5. エラーメッセージ

エラーメッセージを作成します。エラーメッセージの作成には3通りの方法があります。
(Windowsアプリケーション 32bit/64bitの場合)

- 既定のエラーメッセージ (英語デフォルト)
- ユーザーメッセージDLL (UserMsgUs.dllとUserMsgJa.iniファイルを使用)
- エラーメッセージのカスタマイズ (入力フォームから直接メッセージを入力)



IxProtectorエラーメッセージの抑制

IxProtector/WUPIを使用した場合、IxProtector/WUPI側で何かエラーが発生するとエラーメッセージが表示されます。このオプションを選択すると、IxProtector/WUPIによるエラーメッセージを表示させないようにできます。

既定のエラーメッセージ

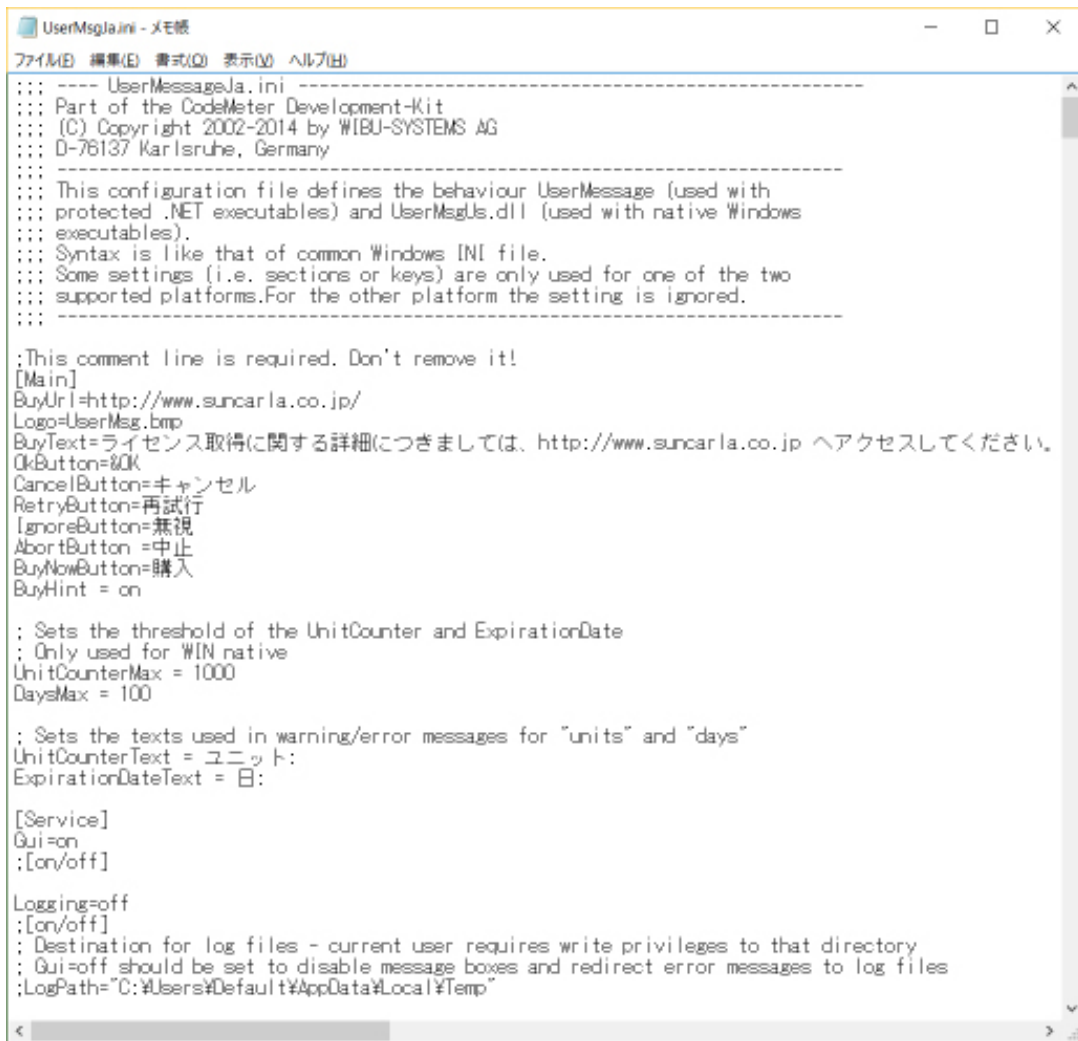
コードメータがあらかじめ用意しているエラーメッセージです。メッセージは英語で表示されます。

ユーザーメッセージ DLL :

BMP画像を含めたメッセージをカスタマイズすることができます。カスタマイズ作業は“UserMsgJa.ini”ファイルの内容を直接編集します。暗号化処理を行うと、暗号化されたプログラムと同じフォルダに“UserMsgJa.ini”が作成されます。この“UserMsgJa.ini”をメモ帳などのエディタで編集してください。編集した“UserMsgJa.ini”やBMP画像ファイルは暗号化されたプログラムと同じフォルダに保存する必要があります。該当するファイルが見つからなかった場合は、コードメータのデフォルトメッセージが表示されます。

【操作例】

1. 「UserMsgJa.ini」ファイルをテキストエディタで開き、必要に応じてメッセージを変更します。



```

UserMsgJa.ini
-----
Part of the CodeMeter Development-Kit
(C) Copyright 2002-2014 by WIBU-SYSTEMS AG
D-76137 Karlsruhe, Germany
-----
This configuration file defines the behaviour UserMessage (used with
protected .NET executables) and UserMsgUs.dll (used with native Windows
executables).
Syntax is like that of common Windows INI file.
Some settings (i.e. sections or keys) are only used for one of the two
supported platforms. For the other platform the setting is ignored.
-----

;This comment line is required. Don't remove it!
[Main]
BuyUrl=http://www.suncarla.co.jp/
Logo=UserMsg.bmp
BuyText=ライセンス取得に関する詳細につきましては、http://www.suncarla.co.jp へアクセスしてください。
OkButton=OK
CancelButton=キャンセル
RetryButton=再試行
IgnoreButton=無視
AbortButton=中止
BuyNowButton=購入
BuyHint = on

; Sets the threshold of the UnitCounter and ExpirationDate
; Only used for WIN native
UnitCounterMax = 1000
DaysMax = 100

; Sets the texts used in warning/error messages for "units" and "days"
UnitCounterText = ユニット:
ExpirationDateText = 日:

[Service]
Gui=on
;[on/off]

Logging=off
;[on/off]
; Destination for log files - current user requires write privileges to that directory
; Gui=off should be set to disable message boxes and redirect error messages to log files
; LogPath="C:\Users\Default\AppData\Local\Temp"

```

2. 暗号化されたプログラムと同じフォルダに、「UserMsgUs.dll」と「UserMsgJa.ini」および使用した画像ファイル(BMP)をおき、コードメータキーを装着しない状態でプログラムを実行するとエラーメッセージが表示されます。

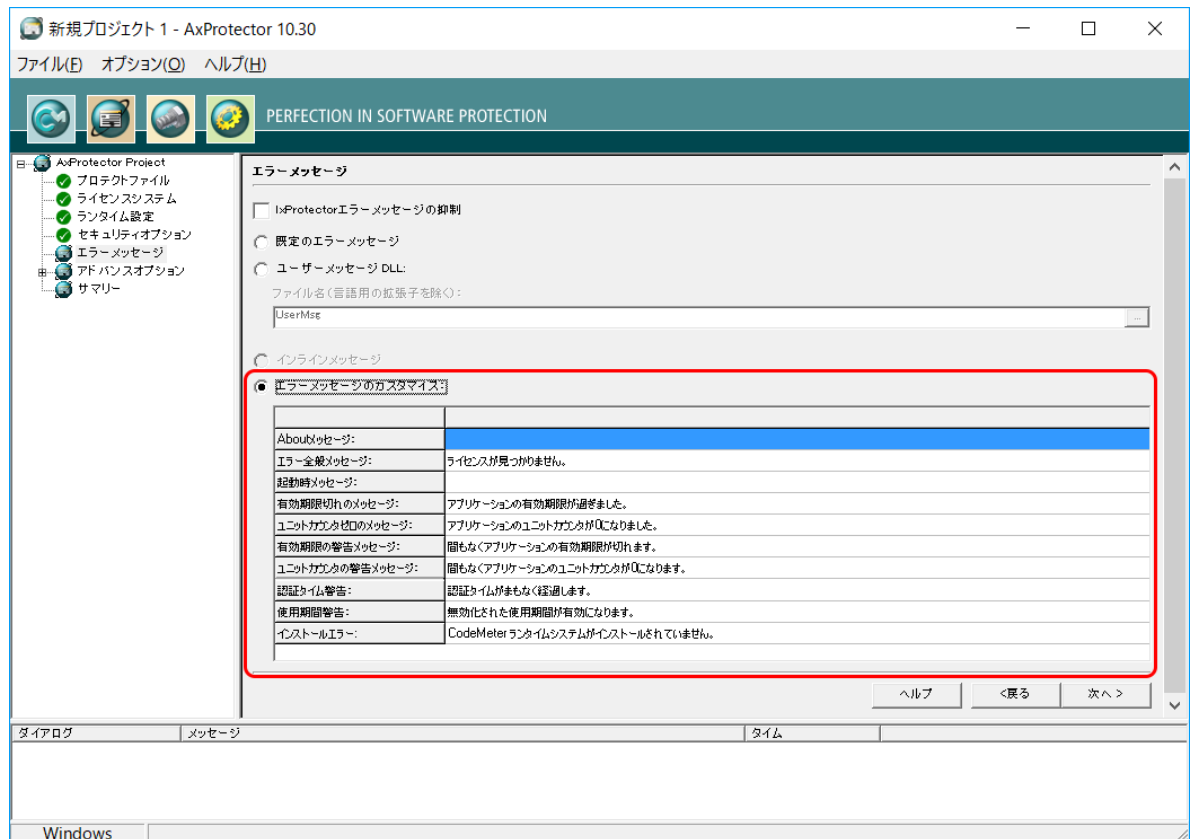


【NOTE】

暗号化されたプログラムを配布する際は、必ず「UserMsgUs.dll」と「UserMsgJa.ini」も一緒に配布し、暗号化されたプログラムと同じフォルダに保存するようにしてください。もし、「UserMsgUs.dll」と「UserMsgJa.ini」が存在しない場合は、コードメータの「既定のエラーメッセージ」(英語デフォルト)が表示されます。

○ エラーメッセージのカスタマイズ：

メッセージボックスのメッセージ部分を変更して、貴社専用のエラーメッセージを作成します。各項目に直接メッセージを入力します。



About メッセージ:

画面には表示されません。

エラー全般メッセージ:

アプリケーション起動時またはランタイムチェック時に、コードメータキーが見つからない場合に表示されるメッセージです。

起動時メッセージ:

アプリケーション起動時に表示されるメッセージです。

有効期限切れのメッセージ:

コードメータキーに設定された有効期限(Expiration Time)または使用期限(Usage Period)が過ぎた時に表示するメッセージです。

ユニットカウンタゼロのメッセージ:

コードメータキーに設定されたユニットカウンタが0の時に表示するメッセージです。

有効期限の警告メッセージ:

コードメータキーに設定された有効期限(Expiration Time)が「警告メッセージを表示する開始点」(「4. ランタイム設定」画面で設定する)に達した時に表示する警告メッセージです。アプリケーションの起動時に表示されます。この警告メッセージは、有効期限が過ぎるまでアプリケーションを起動するたびに表示されます。

ユニットカウンタの警告メッセージ:

コードメータキーに設定されたユニットカウンタが「警告メッセージを表示する開始点」(「4. ランタイム設定」画面で設定する)に達した時に表示する警告メッセージです。アプリケーションの起動時に表示されます。この警告メッセージは、ユニットカウンタが0になるまでアプリケーションを起動するたびに表示されます。

認証タイム警告:

「ランタイム設定」/「アドバンス」の「認証時間」で設定した「認証時間の有効期間(時間)」が間もなく経過する場合に表示する警告メッセージです。

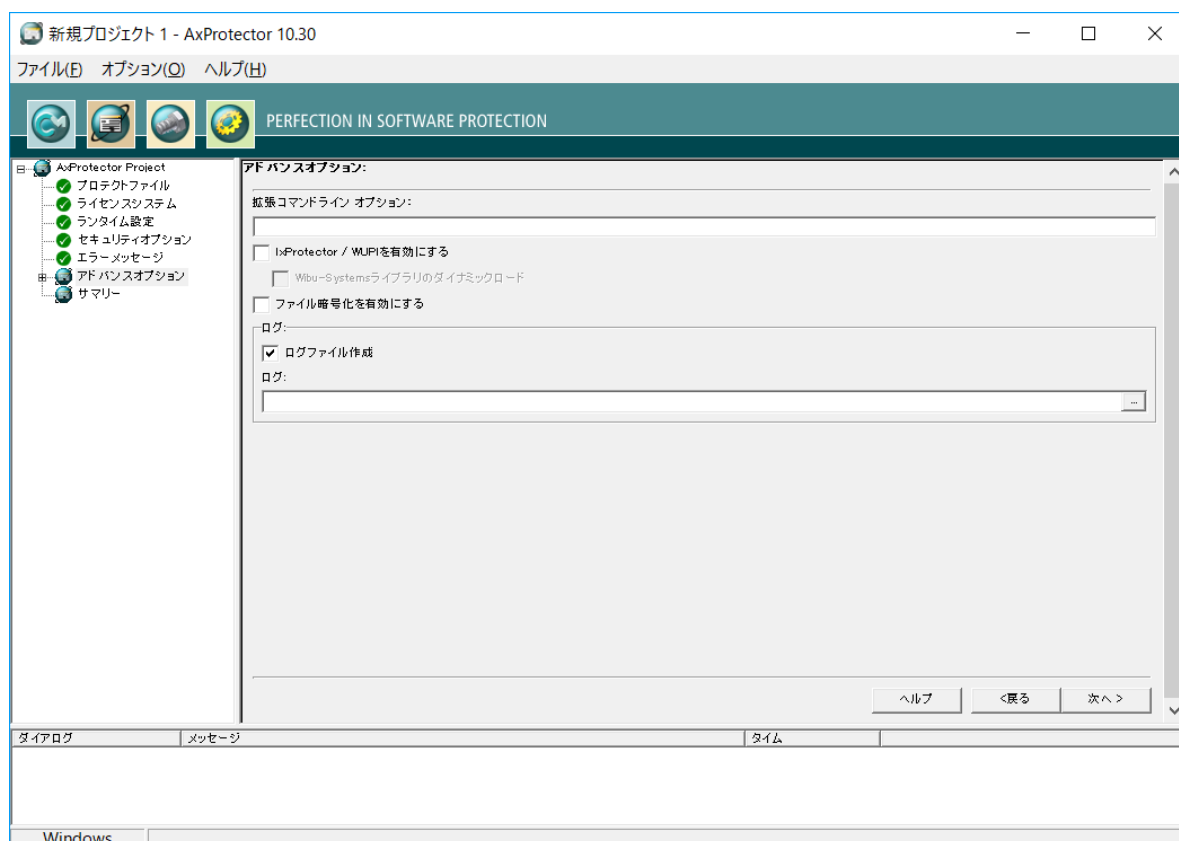
使用期間警告:

コードメータキーに設定された使用期限(Usage Period)が「警告メッセージを表示する開始点」(「4. ランタイム設定」画面で設定する)に達した時に表示する警告メッセージです。アプリケーションの起動時に表示されます。この警告メッセージは、有効期限が過ぎるまでアプリケーションを起動するたびに表示されます。

インストールエラー:

コードメータランタイムがインストールされていない場合に表示します。

6. アドバンスオプション:



拡張コマンドラインオプション

AxProtector操作画面から選択できない拡張機能をコマンドラインでサポートします。

IxProtector / WUPI を有効にする

IxProtector/WUPIを使用する場合は、チェックをいれます。

IxProtectorは、メモリー上で展開されるコードを常に暗号化しておき、必要な時に必要なモジュールだけを復号して実行するという、メモリー上での「オンデマンド復号」を実現する機能です。AxProtectorで暗号化されたコードが、メモリー上でも暗号化されているため、クラッキングに対して非常に強力なセキュリティを実現できます。IxProtectorは、WUPI(Wibu Universal Protection Interface)ファンクションと組み合わせて使用します。(「Chapter 5 IxProtector/WUPIについて」参照)

Wibu-Systemsライブラリのダイナミックロード

VB6でコンパイルしたアプリケーションの場合、またはWibu-Systemsライブラリをダイナミックでロードする場合は、このオプションにチェックを入れます。

[NOTE]

.NETアセンブリとJavaアプリケーションをAxProtectorで暗号化する場合、このIxProtector/WUPI機能を使用しなくても、メモリー上での「オンデマンド復号」機能が自動的に付加されます。従い、「オンデマンド機能」を使用するために、ソースコードを編集する必要はありません。

[NOTE]

Windowsアプリケーション(32/64bit)で「オンデマンド復号」機能を実現するには、IxProtector/WUPIをソースコードに組み込む必要があります。

ファイル暗号化を有効にする

AxProtectorで暗号化されたデータファイルを使用する場合は、ここにチェックを入れます。チェックを入れないと、暗号化されたデータファイルは復号しません。

[NOTE]

暗号化されたデータファイルは、メモリー上で自動的に復号されます。

(復号されるのはあくまでメモリー上であり、ディスク上には常に暗号化された状態で残ります。)

ログファイル作成

暗号化処理のログファイルを作成します。

ログファイル名

ログファイルが作成されるパスとファイル名を指定します。

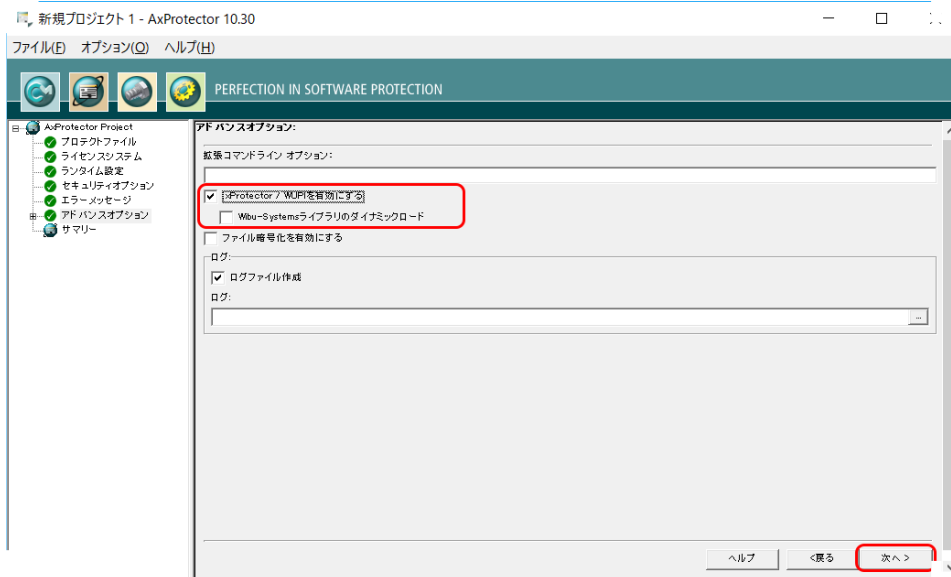
7-1. IxProtector/WUPIの使い方

① IxProtector/WUPIを有効にする

アドバンスオプション画面で、「IxProtector/WUPIを有効にする」にチェックを入れます。

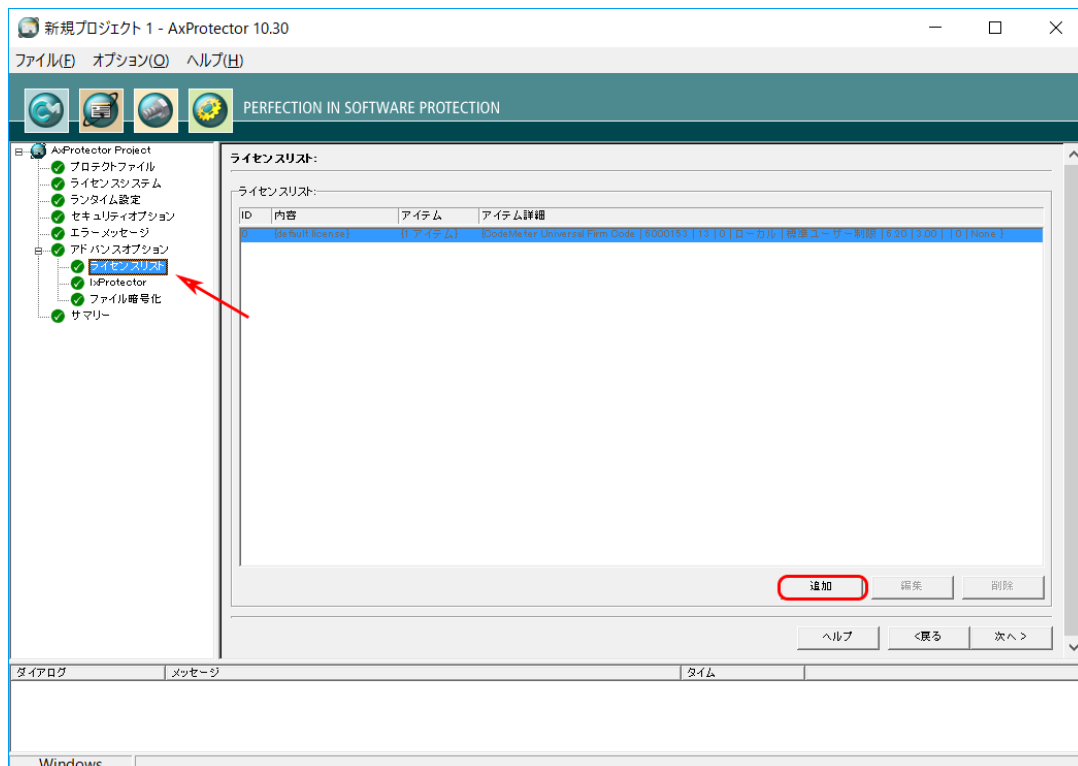
Wibu-Systemsライブラリのダイナミックロード

VB6でコンパイルしたアプリケーションの場合、またはWibu-Systemsライブラリをダイナミックでロードする場合は、このオプションにチェックを入れます。



② ライセンスリストを作成する

「ライセンスリスト」画面が表示されますので、「追加」ボタンをクリックします。IxProtectorは、まずライセンスリストをId単位で作成し、そのIdをWUPIファンクションから呼び出して利用します。



③ ライセンス内容を登録する

Idに対してライセンス内容を登録します。「ライセンスリストの追加」画面で行います。

ライセンスリストの追加

ライセンスリスト:

Id	内容:
1	FC=6000010 FC=1001

ライセンス:

[CodeMeter Universal Firm Code | 6000010 | 1001 | 0 | ローカル | 標準ユーザー制限 | 6.20 | 3.00 | | 0 | None]

ライセンス詳細:

ライセンスシステム:
CodeMeter

ファームコード: 6000010 プロダクトコード: 1001 フィーチャーコード: 0

サブシステム: ローカル ライセンスオプション: 標準ユーザー制限 最小ランタイムバージョン: 6.20 ビルド:

最小ファームウェア: 3.00 リリース日: 2019/06/10 WupiWriteData
 WupiReadData

OK キャンセル ヘルプ

Id:

ライセンスIDを設定する。ライセンスIDは連番で自動表示されます。

内容:

ライセンス内容を記載します。内容はライセンス内容に応じて自由に記載してください。

[ライセンスアイテム詳細]

ライセンスシステム:

使用するライセンスシステムを指定します。コードメータ(CmDongle)、ワイブキー(Wibukey)、コードメータActライセンス(CmActLicense)が選択できます。

ファームコード:

ライセンスIDに割り当てるファームコードを設定します。

プロダクトコード:

ライセンスIDに割り当てるプロダクトコードを設定します。

フィーチャーコード:

ライセンスIDに割り当てるフィーチャコードを設定
します。

2進数から16進数へ

16進数		10進数		2進数			
31	30	29	28	27	26	25	24
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23	22	21	20	19	18	17	16
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	14	13	12	11	10	9	8
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	6	5	4	3	2	1	0
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

OK キャンセル

ライセンスアイテム詳細

ライセンスシステム

リストボックスからCodeMeterとWibuKeyの選択ができます。ライセンスシステムにCmDongleおよびCmActLicenseを使用する場合はCodeMeterを選択します。

[NOTE]

「アイテムの追加」ボタンをクリックして、異なるプロダクトコードを持つCmDongleやCmActLicense、またはWibuKeyのライセンスシステムを追加することができます。生成されるEXEまたはDLLファイルは、ここで指定したライセンスが1つでも存在すると起動することができます。

The 'Add License' dialog box contains the following fields and options:

- ライセンスシステム:** CodeMeter (selected in a dropdown menu)
- ファームコード:** 6000153 (text input)
- プロダクトコード:** 13 (text input)
- フィーチャーコード:** 0 (text input)
- サブシステム:** ローカル (selected in a dropdown menu)
- ライセンスオプション:** ユーザー制限なし (selected in a dropdown menu)
- 最小ドライババージョン:** 6.10 (text input)
- Build:** (empty text input)
- 最小ファームウェア:** 3.00 (text input)
- リリース日:** 2016/12/22 (selected in a dropdown menu)

Buttons at the bottom: OK, キャンセル, ヘルプ

□ファームコード：

貴社のファームコード(Firm Code)を入力します。

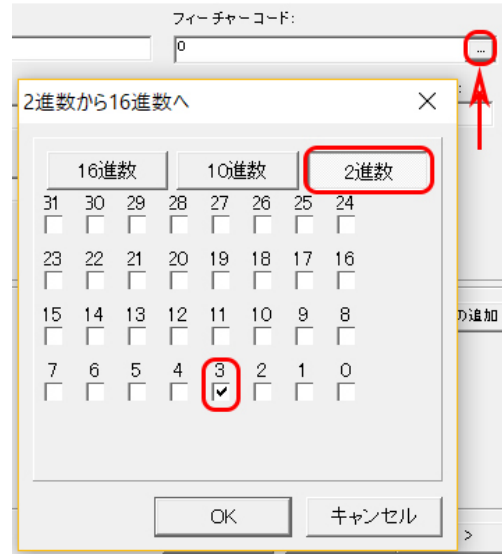
□プロダクトコード：

プロダクトコード(Product Code)を入力します。有効なプロダクトコードの範囲は、1 ~ 4294967295 (32bits)の整数値です。

□フィーチャーコード：

フィーチャーコードを入力します。フィーチャーコードは、おもにバージョン管理に利用します。

例えば、ファームコード=10、プロダクトコード=13、バージョン=3という設定でプログラムを暗号化したい場合、フィーチャーコードに"8"を指定することで実現します。フィーチャーコードの「2進数」ボタンをクリックすると、0 ~ 31までのビットが表示されますので、設定したいバージョンを選択してOKボタンをクリックすると、自動的に10進数に変換します。フィーチャーコードは32ビットですので、実際には32バージョン分を管理することができます。



0x000000000000000000000000000000001000(2進数) = 8 (10進数)

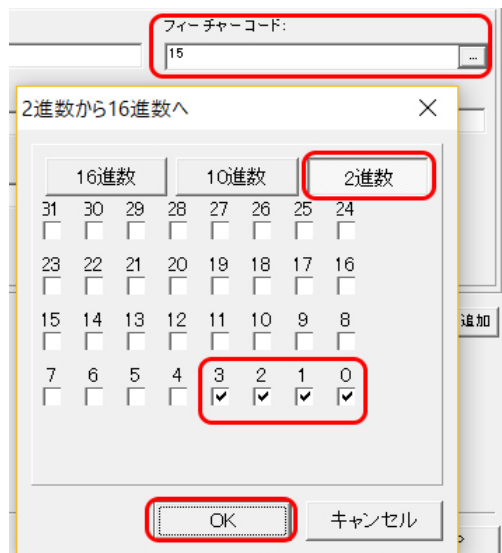
ファームコード=10、プロダクトコード=13、フィーチャーコード=8で暗号化されたプログラムを起動するには、同じコード(ファームコード=10、プロダクトコード=13、フィーチャーコード=8)を持つCmDongleが必要になります。

このフィーチャーコード機能を利用すると、1個のエントリで複数バージョンのライセンス管理が可能になります。例えば、次の4つのバージョンのアプリケーションを1個のエントリで管理する場合、

- バージョン0 (Beta版) ファームコード=10、プロダクトコード=13、フィーチャーコード=1
- バージョン1 ファームコード=10、プロダクトコード=13、フィーチャーコード=2
- バージョン2 ファームコード=10、プロダクトコード=13、フィーチャーコード=4
- バージョン3 ファームコード=10、プロダクトコード=13、フィーチャーコード=8

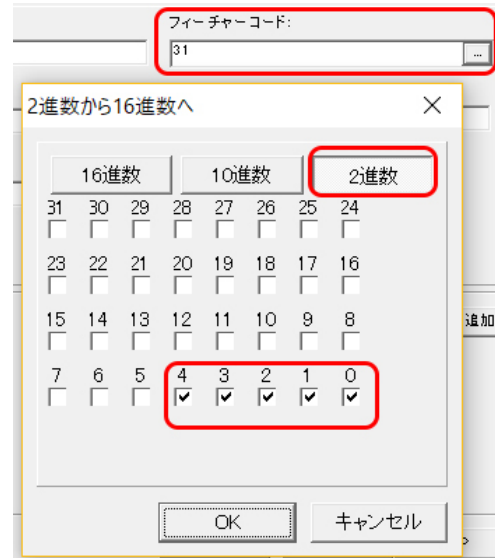
0x000000000000000000000000000000001111 = 15 (10進数)

CmDongle側には、ファームコード=10、プロダクトコード=13、フィーチャーコード=15を登録することで実現します。[2進数から16進数へ]/[2進数]ダイアログで、0,1,2,3にチェックを入れてOKボタンをクリックすると、フィーチャーコード値が10進数に自動計算されてフィーチャーコード欄に入力されます。



また、後日、バージョン4 (ファームコード=10、プロダクトコード=13、フィーチャーコード=16) をリリースした場合、CmDongle側のフィーチャーコードを"15"から"31"に変更することで、5つのバージョンを起動させることが可能になります。

0x0000000000000000000000000000000011111 = 31 (10進数)



CmDongle側のフィーチャーコードの変更(更新)は、コードメータのリモートアップデート機能を利用すると、メール添付のやりとりだけで実現できます。現物(ハードウェア)を送ったり、送り返したりする必要がありません。

□サブシステム:

暗号化されたプログラムが検索する範囲(サブシステム)を割り当てます。

「ローカル」

ローカルのみアクセスする

「ネットワーク」

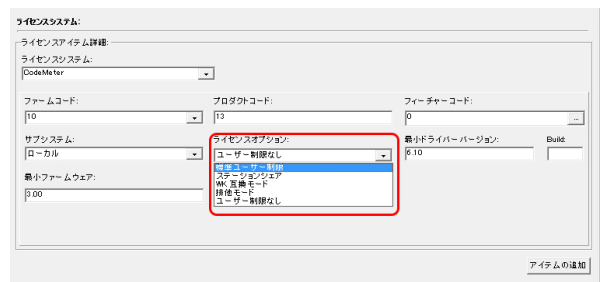
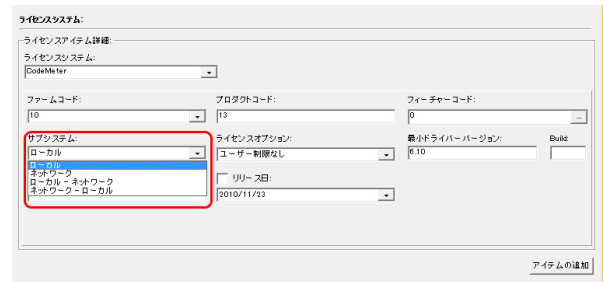
ネットワークのみアクセスする

「ローカルネットワーク」

最初にローカルアクセスを行い、次にネットワークアクセスを行う

「ネットワークローカル」

最初にネットワークアクセスを行い、次にローカルアクセスを行う



□ライセンスオプション:

○標準ユーザー制限(アプリケーション単位で制限)(Normal User Limit)

実行するアプリケーションごとに1つのライセンスを割り当てます。例えば、同じアプリケーションを同時に2回起動する場合は2つのライセンスが必要になります。この原則はコードメータキー(CmDongle)がローカル上にある場合もネットワーク(LAN)上にある場合も同じように適用されます。(1アプリケーション = 1ライセンスの考え方)

○ステーションシェア(クライアント単位で制限)(Station Share)

1台のPCで同一のアプリケーションを同時に複数回起動した場合でも1ライセンスとして扱われます。(1PC = 1ライセンスの考え方)

○WK互換モード(ネットワーク上では標準ユーザー制限、ローカル上は制限なし)

ネットワーク(LAN)上のコードメータキー(CmDongle)に対しては「標準ユーザー制限(1アプリケーション = 1ライセンス)」として動作しますが、ローカルPC上のコードメータキー(CmDongle)に対しては制限がありません。

* CmDongleをローカルPCに装着すると、そのPC上では同一のアプリケーションが無限に起動できるが、そのCmDongleをコードメータサーバーに装着すると、「標準ユーザー制限(1アプリケーション=1ライセンス)」として機能する。

○排他モード(クライアントごとに1つのアプリケーションのみ許可)(Exclusive Mode)

1台のPCで同一アプリケーションの重複起動を禁止します。

○ユーザー制限なし(ユーザー数無制限)(No user Limit)

起動に必要なコードメータキー(CmDongle)がネットワーク(LAN)上で見つければ、ライセンス数の制限にかかわらずアプリケーションが起動します。ライセンス数の制約を受けません。

□最小ドライバーバージョン:

コードメータランタイムキットの最小バージョンを指定します。ここで指定したバージョンより古いバージョンがインストールされていると、暗号化されたプログラムは起動しません。コードメータランタイムキットは、OS環境に応じて定期的にバージョンアップしており、貴社プログラムが古いバージョンで動作し、誤動作するのを防ぐことができます。

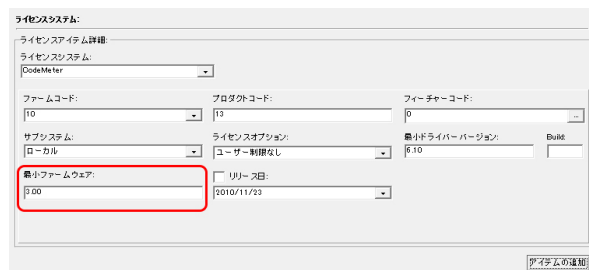
The screenshot shows the 'ライセンスシステム' (License System) configuration window. The 'ライセンスシステム詳細' (License System Details) section includes the following fields:

Field	Value
ライセンスシステム	CodeMeter
ファームコード	110
プロダクトコード	13
フィーチャコード	0
サブシステム	ローカル
ライセンスオプション	ユーザー制限なし
最小ドライバーバージョン	6.10
最小ファームウェア	0.00
有効期限	2010/11/23

The '最小ドライバーバージョン' (Minimum Driver Version) field is highlighted with a red box. A 'Build' button is visible to the right of the 'Minimum Driver Version' field. A 'ライセンスの通知' (License Notification) button is located at the bottom right of the window.

□最小ファームウェア:

CmDongleの要求ファームウェアの最小バージョンを指定します。CmDongle側のファームウェアが、ここで設定した最小ファームウェアのバージョンより低い場合は、暗号化されたプログラムは起動しません。



[注意]

デフォルトでは最小ファームウェアが"3.00"に設定されますが、従来からのCmDongleと互換性をもたせるためには最小ファームウェアを"1.14"または"1.18" (「リリース日」機能を使用する場合) にすることをお勧め致します。

[NOTE]

Universal FirmCode=60xxxxxを使用する場合は、ファームウェアバージョンが3.00以上必要になります。FirmCode=10xxxxの場合は、ファームウェアバージョン1.14以降で動作します。

[NOTE]

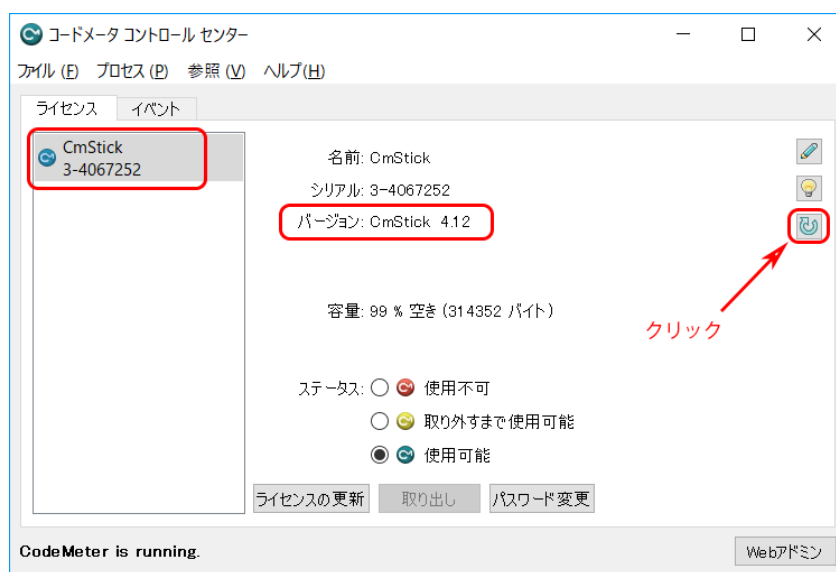
CmDongleファームウェアのバージョンアップは、コードメータコントロールセンター上でユーザーが更新することができます。ただし、CmDongleのシリアル番号によって、最大可能なバージョンが異なります。

シリアル番号 1-xxxxxxxのCmDongleはファームウェア1.18まで可能。

シリアル番号 2-xxxxxxxのCmDongleはファームウェア2.04まで可能。

シリアル番号 3-xxxxxxxのCmDongleはファームウェア3.00以降可能。

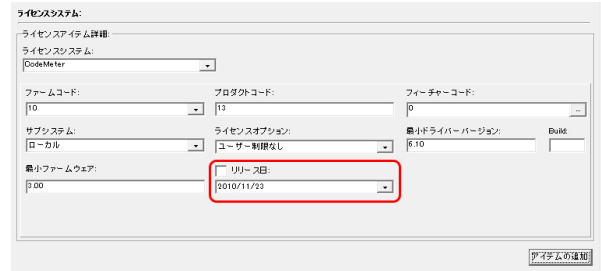
CmDongleのファームウェアの更新は、コードメータコントロールセンター画面の「ファームウェアの更新」ボタンをクリックすると自動的に最新バージョンに更新されます。更新作業は、インターネットに接続する必要があります。



□リリース日:

プログラムのリリース日を設定します。リリース日は、CmContainer内に登録されたメンテナンス期間と連動し、リリース日がメンテナンス期間内であればプログラムが起動し、メンテナンス期間以外の場合はプログラムが起動しません。

この機能は、ユーザーとのプログラムのメンテナンス契約をコントロールするために役立つ機能です。



[注意]

リリース日機能を利用するには、最小ファームウェアは1.18以上が必要になります。

□ WupiWriteData:

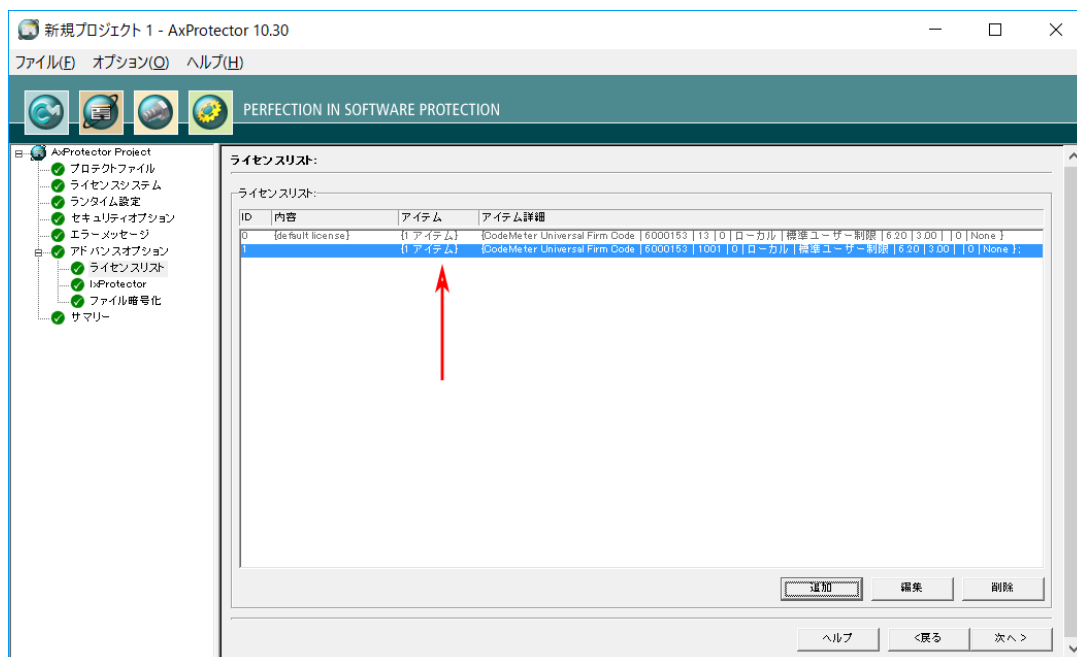
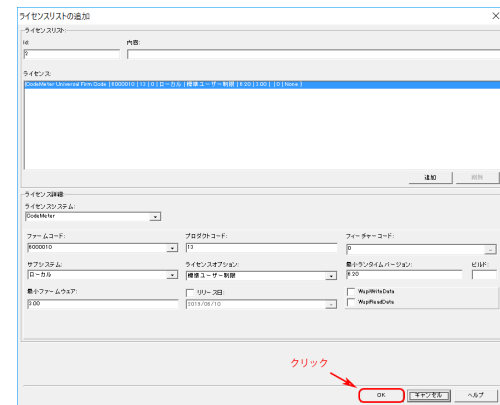
CmContainerからデータを書き込む場合はチェックを入れます。

□ WupiReadData:

CmContainerからデータを読み込む場合はチェックを入れます。

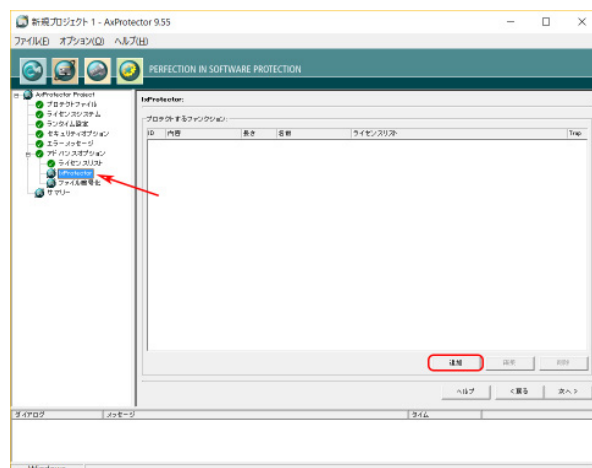


「OK」ボタンをクリックすると、ライセンス内容がライセンスリストに追加登録されます。



④ IxProtector を登録する

IxProtector画面で、「追加」ボタンをクリックして、IxProtectorで暗号化するファンクションを登録します。



⑤ ファンクション名を登録する

「ファンクションの追加」画面でファンクションを登録します。

Id:

ファンクションIDを登録します。ファンクションIDは自動的に連番登録されます。

内容:

ファンクションの内容を登録します。

長さ (%):

暗号化する範囲をパーセンテージ(%)で指定します。%を使用しないで、直接数値を設定すると、設定したバイト数分が暗号化されます。

ファンクション名:

ソースコードの中で実際に使われているファンクション名を正確に入力します。

(例)

CWupiCalculatorDlg::CalcSimpleOperation

OnButtonCalcMemClear

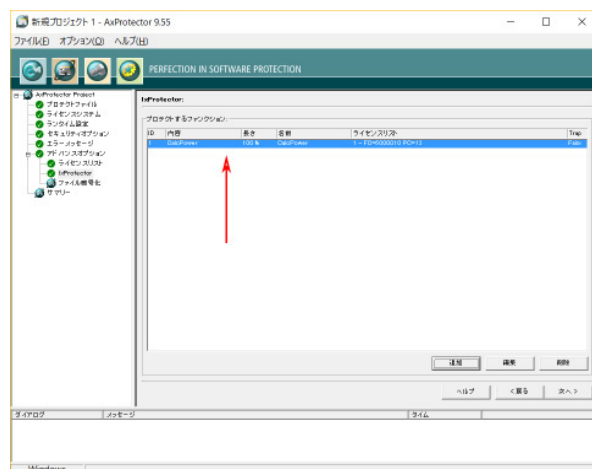
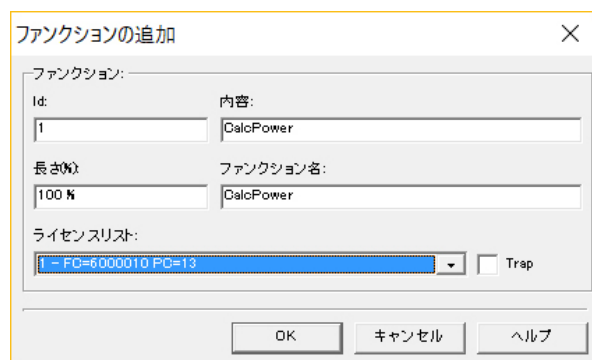
CWupiCalculatorDlg::CalcAngle

ライセンスリスト:

ライセンスリストから使用するライセンスを指定します。

登録後、「OK」ボタンをクリックすると、IxProtectorにファンクションが登録されます。

「次へ」ボタンをクリックして次に進めます。

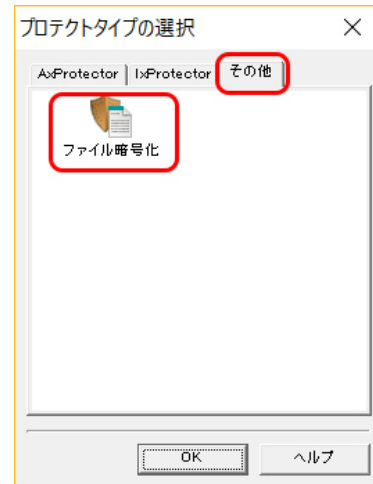


7-2. ファイル暗号化の使い方

この「ファイル暗号化を有効にする」オプションは、プログラムが使用(または制御)するデータファイルをメモリー上で復号化・暗号化する機能です。復号化・暗号化はあくまでメモリー上で行われ、ディスク上のデータファイルは常に暗号化されているため、データファイルのコピープロテクトが可能です。

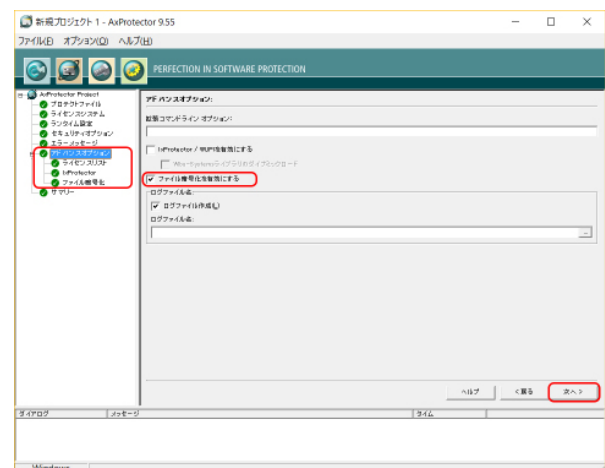
なお、データファイルはAxProtectorの"ファイル暗号化"で最初に暗号化しておきます。詳しくは「Chapter4: 4-6.データファイルを暗号化する」を参照してください。

AxProtector/「アドバンスオプション」/「ファイル暗号化を有効にする」オプションは、暗号化するプログラムに「データファイルの復号・暗号化機能」を持たせるかどうかを選択するオプションです。このオプションを有効にしない場合、データファイルの復号・暗号化は行われません。



① ファイル暗号化を有効にする

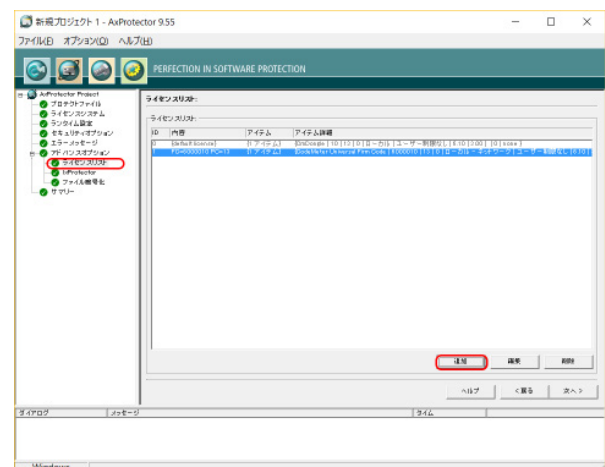
「アドバンスオプション」画面で、「ファイル暗号化を有効にする」にチェックを入れ、「次へ」をクリックする。



② ライセンスリストを作成する

「ライセンスリスト」画面が表示されますので、「追加」ボタンをクリックします。すでに、ライセンスリストにライセンスが登録されている場合は不要です。

ライセンスリストに登録されているライセンスは、「IxProtector」と「ファイル暗号化」の両方で使用可能です。IxProtectorを使用しない場合は、IxProtector画面がグレースアウトしますので、「次へ」ボタンをクリックして次に進めます。



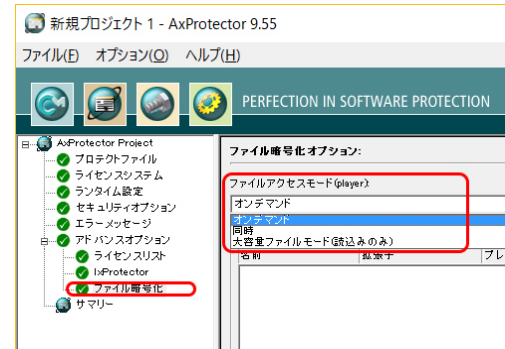
③ ファイルアクセスモードを指定する

「ファイル暗号化オプション」画面で、ファイルアクセスモード (Player) を指定します。ファイルアクセスモードには次の 3 種類があります。

オンデマンド

同時

大容量ファイルモード (読み込みのみ)



オンデマンド:

「オンデマンド」モードでは、プログラム (Player) は、ファイル全体のメモリーを確保しますが、読み込むのは必要な部分だけです。読み込みと復号は 4k バイトブロック単位で行い、必要な時に必要なブロックをロードし復号します。一度復号されたブロックは、メモリー上に残るため、2 回目以降はメモリー上のブロックが使用されます。この「オンデマンド」モードはロードされるファイル全体のメモリーが必要になりますが、一度復号されたデータを再利用できるため、パフォーマンスに優れます。このモードは、データファイルを読み書き (Read/Write) する場合に適します。

同時 (一括):

「同時 (一括)」モードでは、プログラム (Player) は、ファイル全体のメモリーを確保し、ファイル全体をメモリーに読み込んでから一度に復号します。復号されたデータはメモリー上に残り、必要な都度メモリー上から読み出されます。ファイル全体のメモリーが必要になりますが、復号済のデータを利用できるためパフォーマンスに優れます。「オンデマンド」モードと異なる点は、最初のアクセス時にファイル全体を復号するための時間が必要となることです。このモードは、「オンデマンド」モード同様、データファイルを読み書き (Read/Write) する場合に適します。

大容量ファイルモード (読み込みのみ):

この「大容量ファイルモード (読み込みのみ)」モードでは、ファイル全体のメモリーを確保せず、必要な部分を読んで復号します。復号されたデータはメモリーに残らないため、メモリーの消費量を抑えることができます。このモードは、リードオンリー (読み込みのみ) のデータファイルに適します。

④ ファイルタイプの定義を行う

「ファイル暗号化オプション」画面で、「追加」ボタンをクリックし、ファイルタイプの追加を行います。

名前:

ファイルタイプ名を定義します。この名前は暗号化処理には影響しません。単なる参照名です。

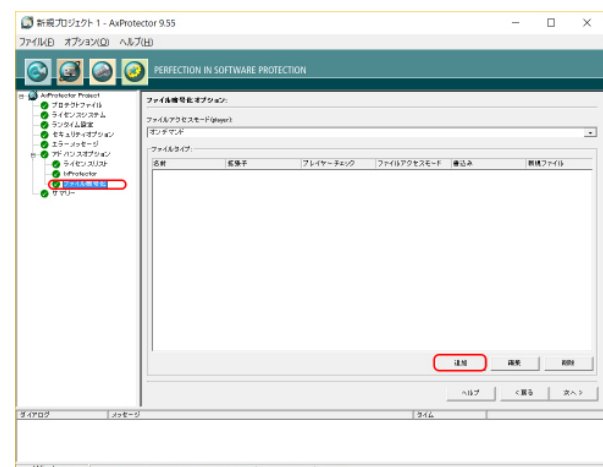
拡張子:

作成するファイルの拡張子を指定します。

(例) txt

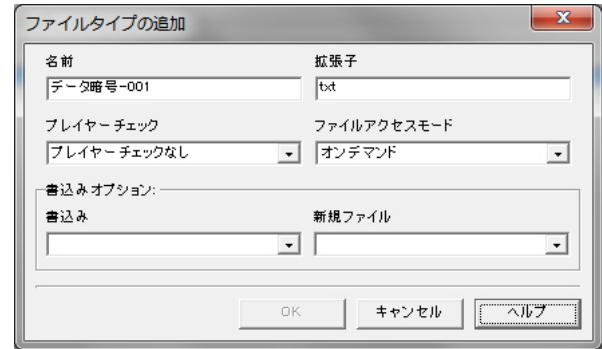
プレイヤーチェック:

プログラム (プレイヤー) 自身のチェックを指定します。特にプレイヤーのチェックを行わない場合は、「プレイヤーチェックなし」を選択します。



ファイルアクセスモード

「オンデマンド」、「同時(一括)」、「大容量ファイルモード(読み込みのみ)」から選択します。

**書き込みオプション:****書き込み:**

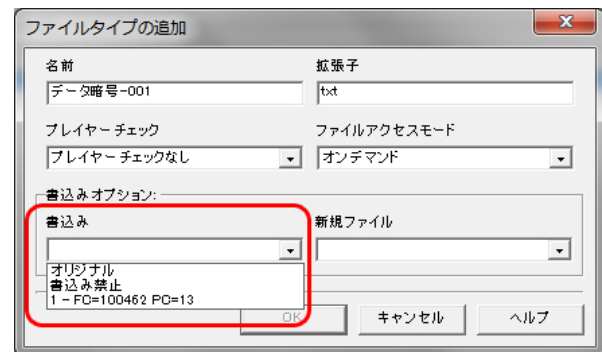
暗号化されていたファイルをプレイヤーで開いた後に編集した場合、どのようなセキュリティ属性で保存するかを指定できます。セキュリティ属性には、「オリジナル」、「書き込み禁止」、「(ライセンスリスト)」の3通りがあります。

オリジナル

あらかじめ設定されていたライセンス仕様(ファームコード、プロダクトコードなど)にて保存する。

書き込み禁止

ファイルへの書き込みを禁止する。
(編集保存禁止)

**(ライセンスリスト)**

指定したライセンス仕様で保存する。ライセンスリストで追加されたライセンスが表示されます。

新規ファイル:

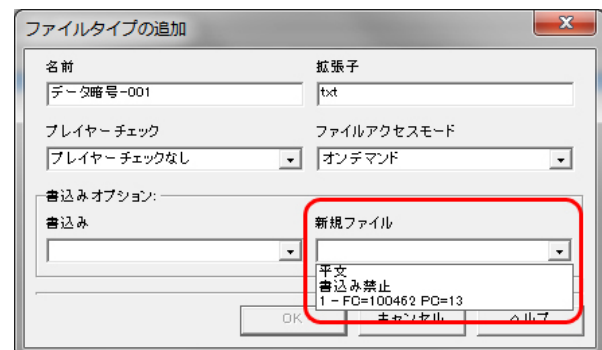
プレイヤーで新しく作成する新規ファイルのセキュリティ属性を指定します。

平文

新規ファイルは平文(暗号化されない)として保存する。

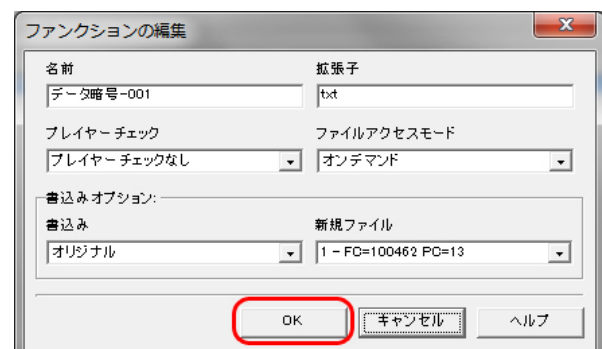
書き込み禁止

新規ファイルは作成しない。新規保存できない。

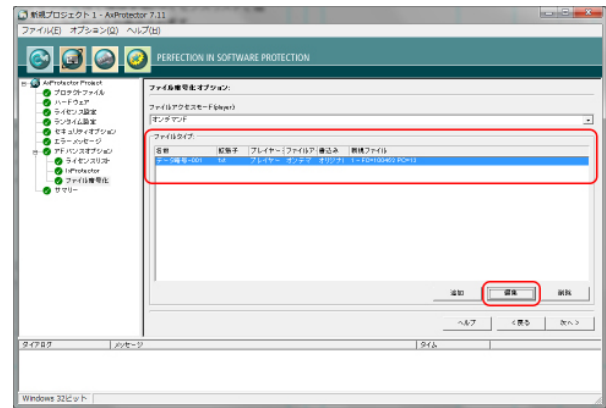
**(ライセンスリスト)**

指定したライセンス仕様で作成保存する。ライセンスリストで追加されたライセンスが表示されます。

ファイルタイプの追加が終了したら「OK」ボタンをクリックして、「ファイル暗号化オプション」画面に戻ります。追加したファイルタイプが登録されているのを確認してください。ここで登録されたデータファイルがプログラム(プレイヤー)によってセキュリティ制御(復号・暗号化)されます。

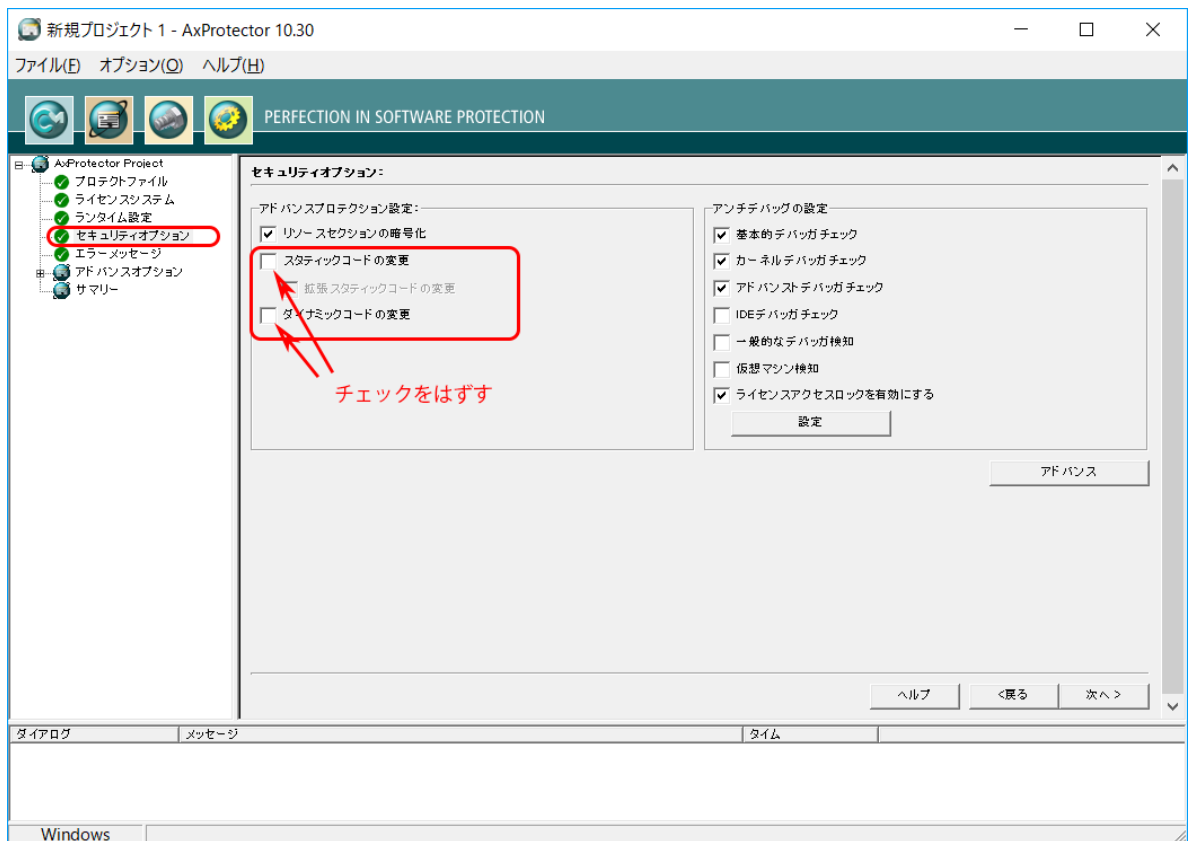


なお、登録したファイルタイプは「編集」ボタンをクリックすることで編集が可能です。



[注意]

制御するプログラムをAxProtectorで暗号化する際、「4. セキュリティオプション」で、「スタティックコードの変更」と「ダイナミックコードの変更」のオプションをはずす必要があります。デフォルトではチェックが入っています。特に「スタティックコードの変更」にチェックを入れたまま暗号化処理を行うと処理の途中でエラーが発生しますのでご注意ください。



7. サマリー：

設定したセキュリティ内容の概要を表示します。設定内容は、コマンドラインによるオプションパラメータで表示されます。

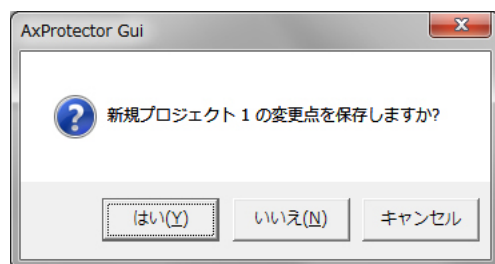
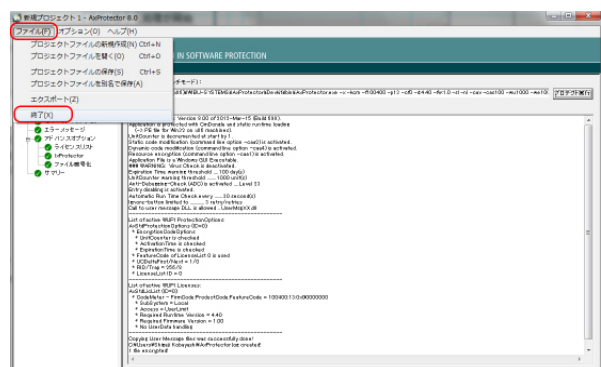
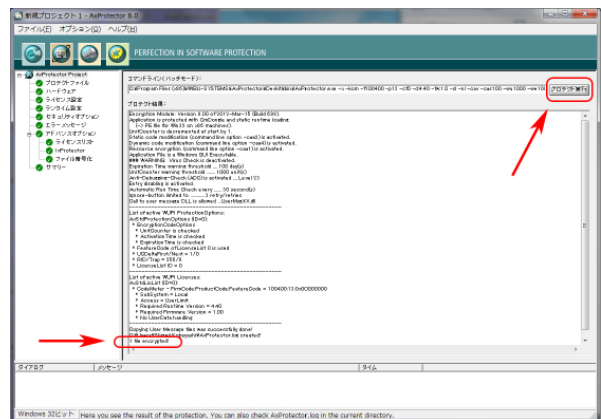
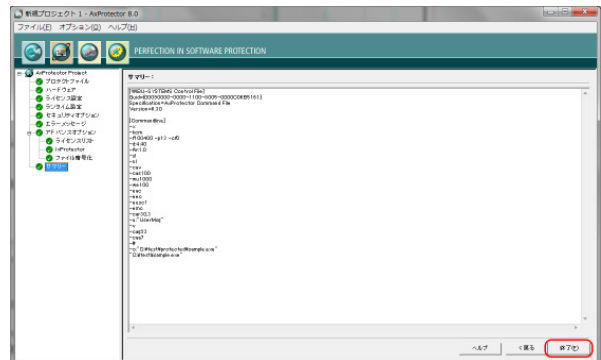
設定に問題がなければ、右下の「終了」ボタンがアクティブになります。もし、設定に問題がある場合は、「終了」ボタンが非アクティブの状態になります。

「終了」ボタンをクリックすると、暗号化処理が開始されます。暗号化処理に成功すると、「プロテクト結果」画面の最下行に "1 file encrypted!" が表示されます。

また、プロテクトオプションのコマンドラインからは、設定したオプションパラメータが表示されます。このコマンドラインを直接入力して、オプションパラメータを変更および追加をして暗号化処理を行うことも可能です。右部の「プロテクト実行」ボタンをクリックすると暗号化処理が開始されます。

[ファイル]メニューから[終了]を選択してAxProtectorを閉じます。

閉じる際、プロジェクトファイルを保存するか画面が表示されます。プロジェクトファイルとは、各項目で設定したセキュリティ内容を保存しておくものです。どのファイルをどのようなセキュリティ内容で暗号化したかの管理ができます。また、プロジェクトファイルを直接開くことで、最初から各項目を入力する手間が省けます。



8. プロジェクトファイルを開く

AxProtectorを起動したあと、[ファイル]メニューから[プロジェクトファイルを開く]を選択し、保存されているプロジェクトファイルを開きます。プロジェクトファイルを開くと、プロジェクトファイルに設定されている内容が反映されます。

また、直近に使用したファイルが最下行に表示されますので、そこからファイルを開くこともできます。



4-5. .NET アセンブリ



1. プロテクトファイル

暗号化する前のオリジナルファイル名と、暗号化生成されるファイル名を指定します。AxProtectorを起動した直後はこの画面が表示されます。

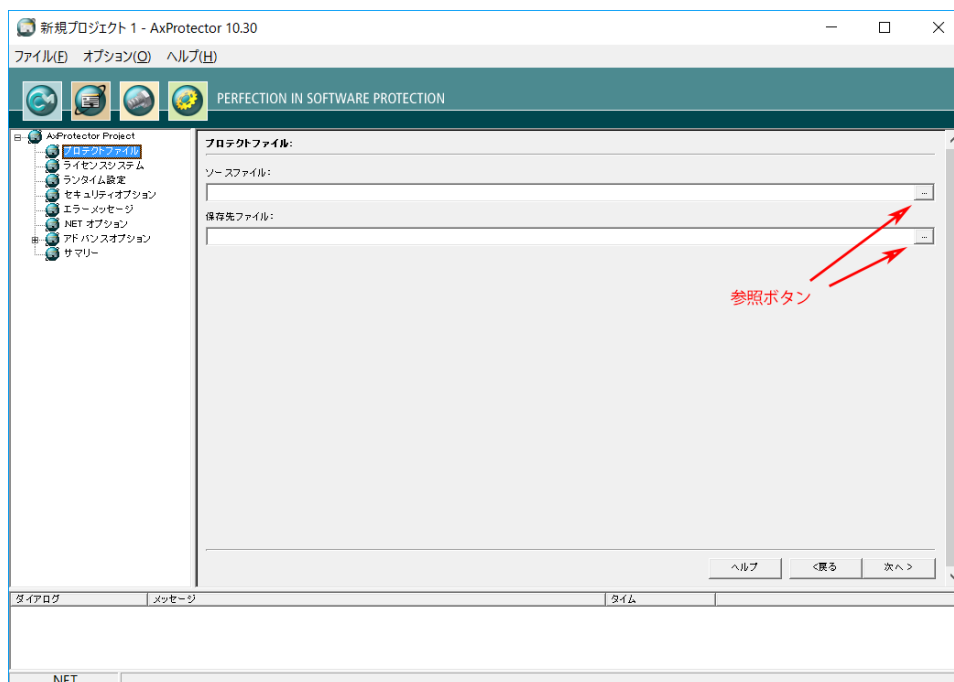
ソースファイル:

暗号化する前のオリジナルファイル名を指定します。右部の参照ボタンからファイルを指定することもできます。

保存先ファイル:

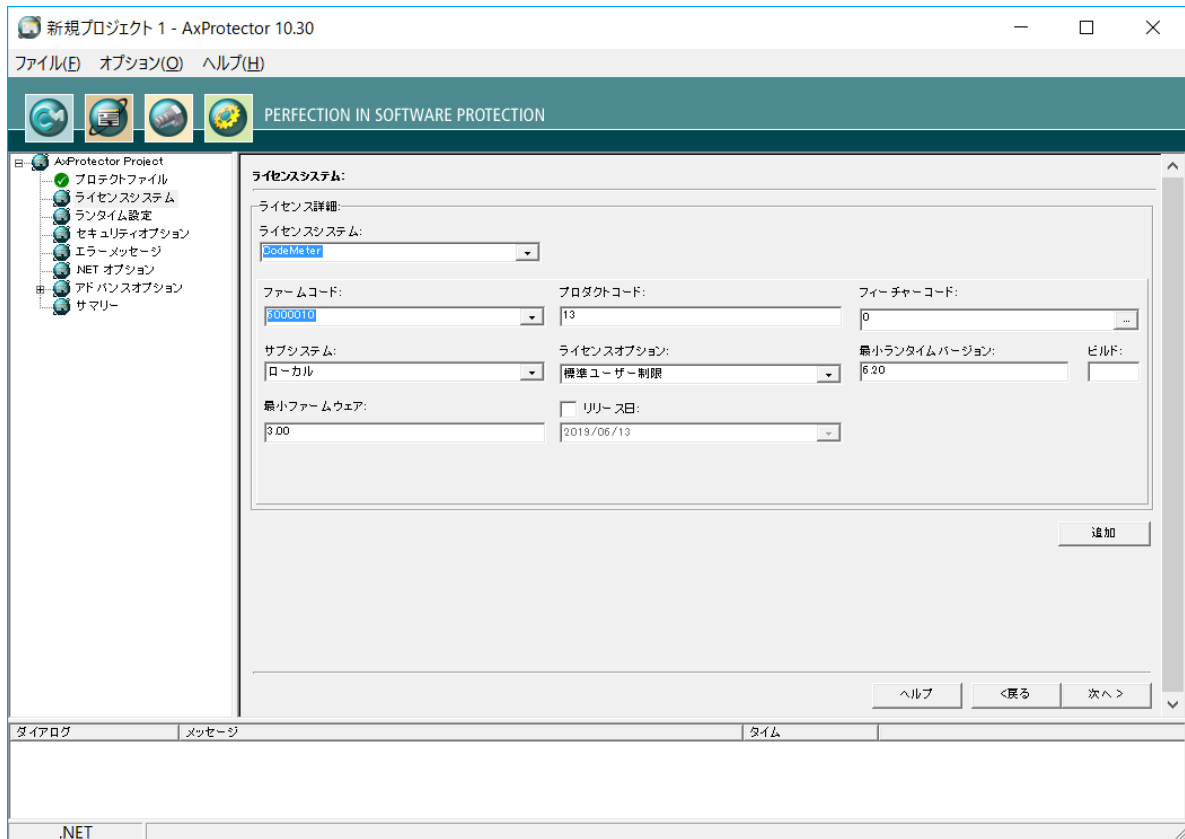
暗号化生成されるファイル名とその保存先のフォルダ名を指定します。右部の参照ボタンからファイルを指定することもできます。

ソースファイルと保存先ファイルを同一にすると、ソースファイルが暗号化生成されるファイルに上書きされますのでご注意ください。ファイル名が同じ場合は別フォルダに保存するか、同じフォルダに保存する場合は、ファイル名を変更するようにしてください。なお、ソースファイルを指定すると、同一フォルダの中にprotectedフォルダが自動的に作成されます。



2. ライセンスシステム

プロテクトに使用するハードウェアキーを指定します。AxProtectorはプロテクトのハードウェアキーとして、コードメータ (CmDongle)、コードメータAct (CmActLicense)、ワイブキー (WibuKey) を使用することができます。

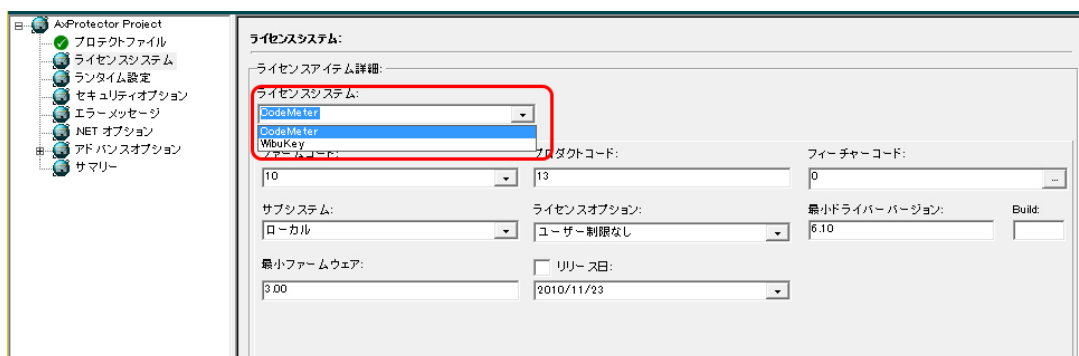


□ ライセンスシステム

リストボックスからCodeMeterとWibuKeyの選択ができます。ライセンスシステムにCmDongleおよびCmActLicenseを使用する場合はCodeMeterを選択します。

[NOTE]

「アイテムの追加」ボタンをクリックして、異なるプロダクトコードを持つCmDongleやCmActLicense、またはWibuKeyのライセンスシステムを追加することができます。生成されるEXEまたはDLLファイルは、ここで指定したライセンスが1つでも存在すると起動することができます。



□サブシステム:

暗号化されたプログラムが検索する範囲(サブシステム)を割り当てます。

ライセンスシステム:

ライセンスアイテム詳細:

ライセンスシステム:
CodeMeter

ファームコード: 10 プロダクトコード: 13 フィーチャーコード: 0

サブシステム:
ローカル
ローカル
ネットワーク
ローカル - ネットワーク
ネットワーク - ローカル

ライセンスオプション:
ユーザー制限なし

最小ドライババージョン: 6.10 Build:

リリース日:
2010/11/23

アイテムの追加

「ローカル」

ローカルのみアクセスする

「ネットワーク」

ネットワークのみアクセスする

「ローカル-ネットワーク」

最初にローカルアクセスを行い、次にネットワークアクセスを行う

「ネットワーク-ローカル」

最初にネットワークアクセスを行い、次にローカルアクセスを行う

□ライセンスオプション:

○ 標準ユーザー制限 (アプリケーション単位で制限) (Normal User Limit)

実行するアプリケーションごとに1つのライセンスを割り当てます。例えば、同じアプリケーションを同時に2回起動する場合は2つのライセンスが必要になります。この原則はコードメータキー (CmDongle) がローカル上にある場合もネットワーク(LAN)上にある場合も同じように適用されます。(1アプリケーション = 1ライセンスの考え方)

○ ステーションシェア (クライアント単位で制限) (Station Share)

1台のPCで同一のアプリケーションを同時に複数回起動した場合でも1ライセンスとして扱われます。(1PC = 1ライセンスの考え方)

○ WK互換モード (ネットワーク上では標準ユーザー制限、ローカル上は制限なし)

ネットワーク(LAN)上のコードメータキー (CmDongle) に対しては「標準ユーザー制限 (1アプリケーション = 1ライセンス)」として動作しますが、ローカルPC上のコードメータキー (CmDongle) に対しては制限がありません。

* CmDongleをローカルPCに装着すると、そのPC上では同一のアプリケーションが無限に起動できるが、そのCmDongleをコードメータサーバーに装着すると、「標準ユーザー制限 (1アプリケーション=1ライセンス)」として機能する。

○ 排他モード (クライアントごとに1つのアプリケーションのみ許可) (Exclusive Mode)

1台のPCで同一アプリケーションの重複起動を禁止します。

○ ユーザー制限なし (ユーザー数無制限) (No user Limit)

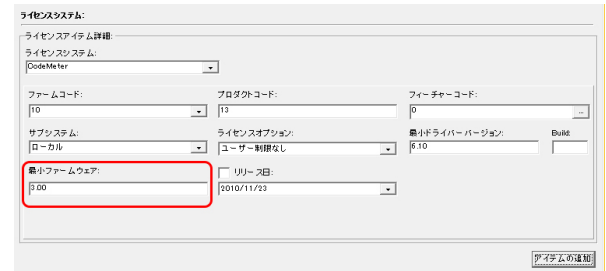
起動に必要なコードメータキー (CmDongle) がネットワーク(LAN)上で見つければ、ライセンス数の制限にかかわらずアプリケーションが起動します。ライセンス数の制約を受けません。

□最小ドライババージョン:

コードメータランタイムキットの最小バージョンを指定します。ここで指定したバージョンより古いバージョンがインストールされていると、暗号化されたプログラムは起動しません。コードメータランタイムキットは、OS環境に応じて定期的にバージョンアップしており、貴社プログラムが古いバージョンで動作し、誤動作するのを防ぐことができます。

□最小ファームウェア:

CmDongleの要求ファームウェアの最小バージョンを指定します。CmDongle側のファームウェアが、ここで設定した最小ファームウェアのバージョンより低い場合は、暗号化されたプログラムは起動しません。



[注意]

デフォルトでは最小ファームウェアが"3.00"に設定されますが、従来からのCmDongleと互換性をもたせるためには最小ファームウェアを"1.14"または"1.18"('リリース日'機能を使用する場合)にすることを勧め致します。

[NOTE]

Universal FirmCode=60xxxxを使用する場合は、ファームウェアバージョンが3.00以上必要になります。FirmCode=10xxxxの場合は、ファームウェアバージョン1.14以降で動作します。

[NOTE]

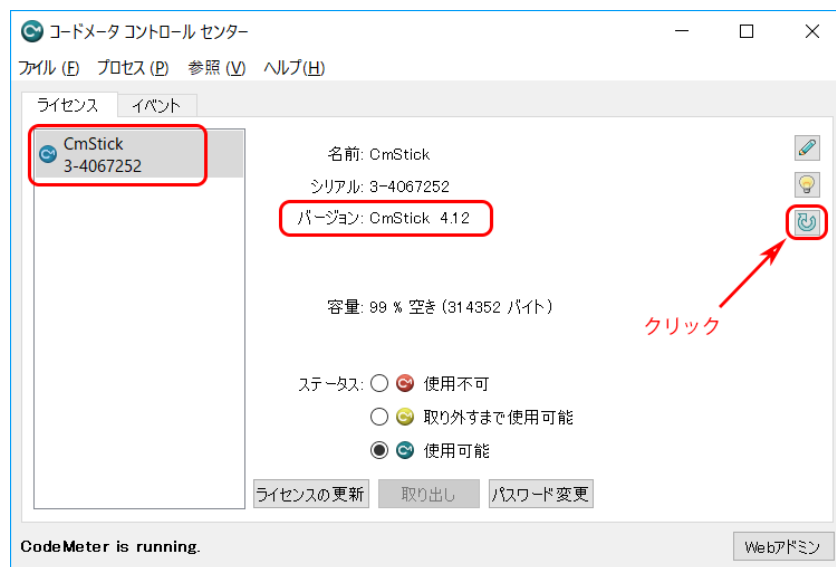
CmDongleファームウェアのバージョンアップは、コードメータコントロールセンター上でユーザーが更新することができます。ただし、CmDongleのシリアル番号によって、最大可能なバージョンが異なります。

シリアル番号 1-xxxxxxxのCmDongleはファームウェア1.18まで可能。

シリアル番号 2-xxxxxxxのCmDongleはファームウェア2.04まで可能。

シリアル番号 3-xxxxxxxのCmDongleはファームウェア3.00以降可能。

CmDongleのファームウェアの更新は、コードメータコントロールセンター画面の「ファームウェアの更新」ボタンをクリックすると自動的に最新バージョンに更新されます。更新作業は、インターネットに接続する必要があります。



□リリース日:

プログラムのリリース日を設定します。リリース日は、CmContainer内に登録されたメンテナンス期間と連動し、リリース日がメンテナンス期間内であればプログラムが起動し、メンテナンス期間以外の場合はプログラムが起動しません。

この機能は、ユーザーとのプログラムのメンテナンス契約をコントロールするために役立つ機能です。

ライセンスシステム:
ライセンスシステム詳細:
ライセンスシステム: CodeMeter
ファームコード: 10 プロダクトコード: 0 フィーチャーコード: 0
サブシステム: ローカル ライセンスオプション: ユーザー制限なし 最小ランタイムバージョン: 6.10 ビルド:
最小ファームウェア: 0.00 リリース日: 2010/11/23
ライセンスの追加

[注意]

リリース日機能を利用するには、最小ファームウェアは1.18以上が必要になります。

□「追加」ボタンについて

複数のライセンスシステムを追加する場合は、「追加」ボタンをクリックしてライセンスシステムを追加します。

ライセンスシステム:
ライセンスシステム詳細:
ライセンスシステム: CodeMeter
ファームコード: 1000153 プロダクトコード: 10 フィーチャーコード: 0
サブシステム: ローカル ライセンスオプション: 標準ユーザー制限 最小ランタイムバージョン: 6.20 ビルド:
最小ファームウェア: 0.00 リリース日: 2019/06/10
追加

ライセンスの追加
ライセンス詳細:
ライセンスシステム: CodeMeter
ファームコード: 6000153 プロダクトコード: 1000 フィーチャーコード: 0
サブシステム: ローカル ライセンスオプション: 標準ユーザー制限 最小ランタイムバージョン: 6.20 ビルド:
最小ファームウェア: 3.00 リリース日: 2019/06/10
OK キャンセル ヘルプ

3. ランタイム設定

使用制限機能に関する動作を設定します。



ランタイムチェック:

ランタイムチェック(コードメータキーの定期的チェック)のインターバル時間を設定します。

ランタイムチェックを有効

ランタイムチェック機能を有効にします。

インターバル時間(時:分:秒):

ランタイムチェックにおいてコードメータキーのチェックが行なわれてから次のチェックが行なわれるまでのインターバルを時分秒で設定します。デフォルトは30秒(00:00:30)が設定されています。

エラー許容回数:

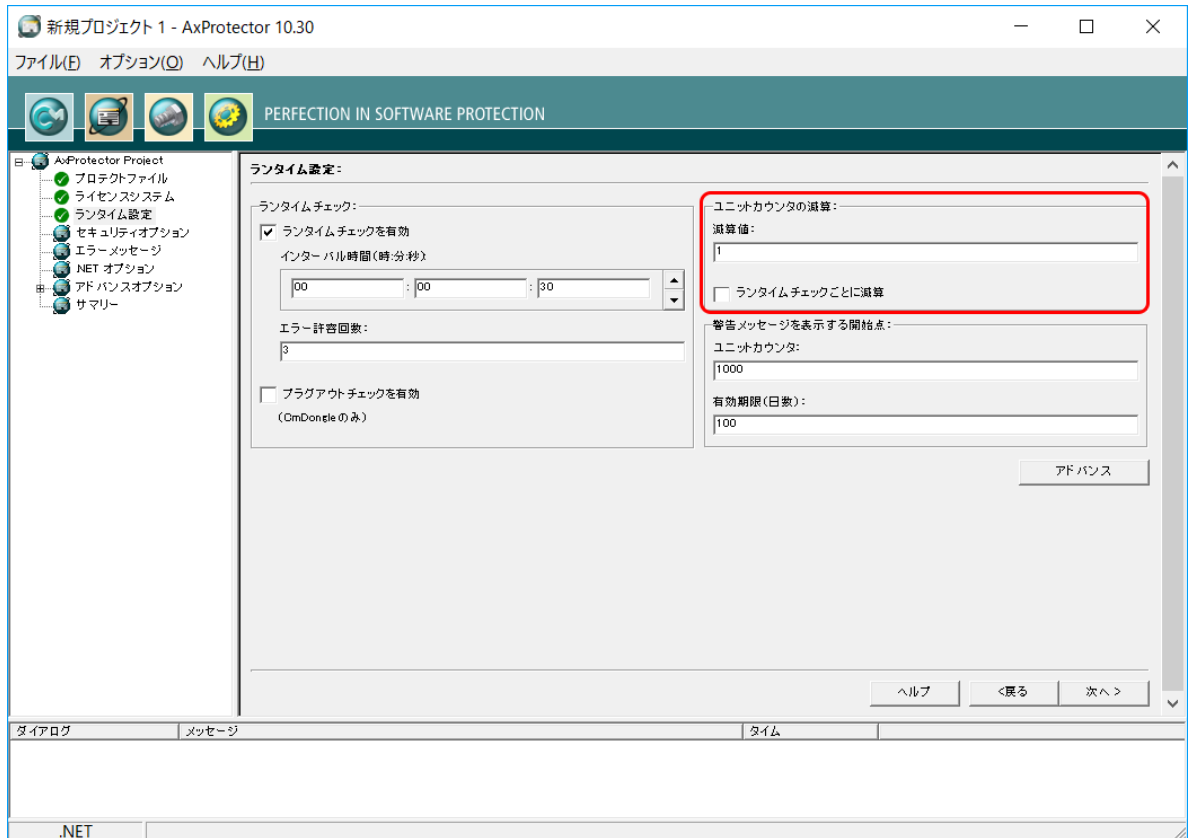
この機能は、コードメータキーをアプリケーション稼働中に抜くなど、何らかの理由でランタイムチェックが失敗した場合(チェックエラーの場合)、ただちにアプリケーションを終了させず、許可した回数だけアプリケーションを続行させる機能です。例えば、ここで3(デフォルト)を設定すると、ランタイムチェックエラーになっても3回まではアプリケーションを続行させることが可能です。しかし、4回目でエラーになるとアプリケーションを終了させます。その際、強制的にアプリケーションを終了させるのではなく、「中止」が「再試行」の選択メッセージを出します。コードメータキーを装着して「再試行」をクリックすると、アプリケーションは続行します。コードメータキーを装着せずに「中止」をクリックするとアプリケーションは終了します。この機能を使わず、1回目のランタイムチェックエラーでアプリケーションを終了させたい場合は0を設定します。

プラグアウトチェックを有効(CmDongleのみ)

コードメータキーをPCから取り外すとすぐにエラーを表示させます。「プラグアウト」を監視する。

ユニットカウンタの減算:

コードメタキーのユニットカウンタの減算値を設定します。ユニットカウンタを使用すると、コードメタキーのチェックが行なわれるたびにユニットカウンタチェックが行なわれ、「減算値」で設定した数値が減算されます。ユニットカウンタが0(ゼロ)になるとアプリケーションの起動ができなくなります。



減算値:

1回のチェックで減算する値を設定します。デフォルト値は"1"です。

ランタイムチェックごとに減算

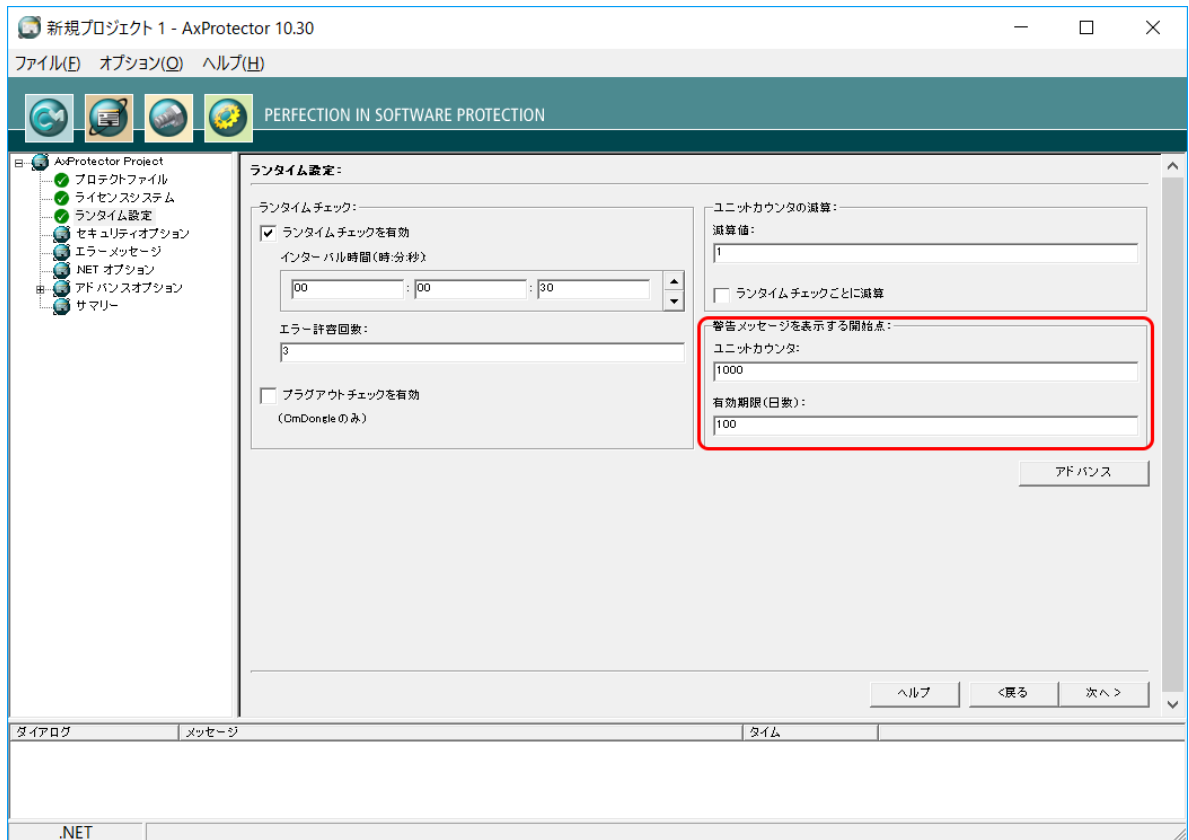
ランタイムチェックが行なわれるごとにユニットカウンタの減算を行いません。ランタイムチェック時に減算を行わない場合はチェックをはずします。この機能はランタイムチェックのインターバル時間と組み合わせることで、プログラムの使用可能時間を設定することができます。

[NOTE]

プログラムの使用可能時間は、使用期間 (Usage Period)機能でも設定できます。

警告メッセージを表示する開始点：

ユニットカウンタや有効期限(日数)の残りが一定の値を下回った場合に警告メッセージを表示させるオプションです。



□ ユニットカウンタ：

ユニットカウンタがあと残りいくつになると警告メッセージを出すかの開始点を設定します。警告メッセージは開始点から0になるまでアプリケーションを起動するたびに表示されます。出力される警告メッセージは、ユニットカウンタ警告用メッセージです。

(デフォルト値は1000回：ユニットカウンタが1000になると警告メッセージを表示し始める)

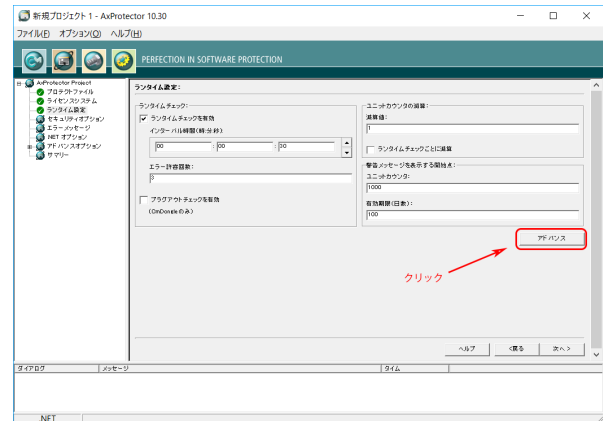
□ 有効期限(日数)：

有効期限まであと何日になると警告メッセージを出すかの開始点を設定します。警告メッセージは有効期限に達するまで、アプリケーションを起動するたびに表示されます。出力される警告メッセージは有効期限警告用メッセージです。

(デフォルト値は100日：有効期限(日数)が100日になると警告メッセージを表示し始める)

アドバンス:

[アドバンス]ボタンをクリックすると「ランタイムの設定」(アドバンス設定)画面が開きます。



アドバンスランタイム設定

ユニットカウンタチェック: <input checked="" type="radio"/> 標準 <input type="radio"/> 要求 (CodeMeterのみ) <input type="radio"/> 無視 (CodeMeterのみ)	有効期限チェック: <input checked="" type="radio"/> 標準 <input type="radio"/> 要求 (CodeMeterのみ) <input type="radio"/> 無視 (CodeMeterのみ)	アクティベーションタイムチェック (CodeMeterのみ) <input checked="" type="radio"/> 標準 <input type="radio"/> 要求 <input type="radio"/> 無視
システムタイムチェック (CodeMeterのみ): <input type="checkbox"/> 暗号化タイムチェック <input type="checkbox"/> CmContainer / PCシステムタイムチェック 過去に対する許容時間(分): <input type="text" value="15"/> 将来に対する許容時間(分): <input type="text" value="15"/>	メンテナンス期間チェック (CodeMeterのみ) <input checked="" type="radio"/> 標準 <input type="radio"/> 要求	
認証時間 (CodeMeterのみ) <input type="checkbox"/> 認証時間をセット <input type="checkbox"/> 認証時間のチェック 認証時間の有効期間(時間): <input type="text" value="100"/> タイムチェック無しの期間(時間): <input type="text" value="0"/>	アドバンスオプション: <input type="checkbox"/> ControlとAboutをメニュー追加 <input checked="" type="checkbox"/> ホストアプリケーションを終了 <input type="checkbox"/> モバイルアプリケーション生成	

OK ヘルプ

ユニットカウンタチェック:

○標準

コードメータキー (CmDongle) にユニットカウンタが設定されている場合は、ユニットカウンタ値をチェックし、設定されていない場合は無視する。(推奨)

○要求 (CodeMeterのみ)

ユニットカウンタを必ずチェックする。もし、コードメータキー (CmDongle) にユニットカウンタが設定されていない場合は、プログラムを起動しない。

○無視 (CodeMeterのみ)

ユニットカウンタを無視する。コードメータキー (CmDongle) にユニットカウンタが設定されていても、プログラムは無視する。実質上、ユニットカウンタ機能なしの状態になる。

有効期限チェック:**○標準**

コードメータキー (CmContainer) に有効期限が設定されている場合は、有効期限をチェックし、設定されていない場合は無視する。(推奨)

○要求 (CodeMeterのみ)

有効期限を必ずチェックする。もし、コードメータキー (CmContainer) に有効期限が設定されていない場合は、プログラムを起動しない。

○無視 (CodeMeterのみ)

有効期限を無視する。コードメータキー (CmContainer) に有効期限が設定されていても、プログラムは無視する。実質上、有効期限なしの状態になる。

アクティベーションチェック(CodeMeterのみ):**○標準**

コードメータキー (CmContainer) にアクティベーションタイム (使用開始日) が設定されている場合は、アクティベーションタイムをチェックし、設定されていない場合は無視する。(推奨)

○要求

アクティベーションタイムを必ずチェックする。もし、コードメータキー (CmContainer) にアクティベーションタイムが設定されていない場合は、プログラムを起動しない。

○無視

アクティベーションタイムを無視する。コードメータキー (CmContainer) にアクティベーションタイムが設定されていても、プログラムは無視する。実質上、アクティベーションタイムなしの状態。

システムタイムチェック (CodeMeterのみ):**□ 暗号化タイムチェック**

プログラム暗号化時のPC時刻 (PCシステムタイム) を記憶し、プログラム起動時に、CmDongle (CmContainer) 内のシステムタイム (CmContainer) と比較する。システムタイム (CmContainer) の方が、プログラム暗号化時のPC時刻より進んでいる (新しい) 場合のみ起動する。

□ CmContainer/PCシステムタイムチェック

システムタイム (CmContainer) とPCシステムタイムの差を時間 (Minutes = 分) で設定する。

過去に対する許容時間 (分)

PCシステムタイムがシステムタイム (CmContainer) よりどれだけ遅れている (古い) かを分単位で設定する。設定した時間 (分) 以内であれば暗号化したプログラムは起動し、それ以上遅れている場合は起動しない。

システムタイムチェック (CodeMeterのみ):

暗号化タイムチェック

CmContainer / PCシステムタイムチェック

過去に対する許容時間(分):

15

将来に対する許容時間(分):

15

将来に対する許容時間(分)

PCシステムタイムがシステムタイム(CmContainer)よりどれだけ進んでいる(早い)かを分単位で設定する。設定した時間(分)以内であれば暗号化したプログラムは起動し、それ以上進んでいる場合は起動しない。

[NOTE]

CmContainerとは、CmDongleとCmActLicenseを総称した表現です。

メンテナンス期間チェック(CodeMeterのみ)：

○ 標準

コードメータキー(CmDongle)にメンテナンス期間が設定されている場合は、メンテナンス期間をチェックし、設定されていない場合は無視する。(推奨)

○ 要求

メンテナンス期間を必ずチェックする。もし、コードメータキー(CmDongle)にメンテナンス期間が設定されていない場合は、プログラムを起動しない。

認証時間(CodeMeterのみ)：

□ 認証時間をセット

このオプションにチェックを入れると、プログラムが起動するたびにコードメータタイムサーバーにアクセスし、コードメータキー(CmContainer)の認証時間とボックス時間を更新します。

□ 認証時間のチェック

このオプションにチェックを入れると、プログラムが起動するたびにコードメータタイムサーバーにアクセスし現在の正確な時刻をチェックし、コードメータキー(CmContainer)の"システムタイム(CmContainer)"と比較します。

認証時間の有効期間(時間)

現在の時刻とコードメータキー(CmContainer)のシステムタイムの差が、「認証時間の有効時間(時間)」で設定された数値(時間)以上に差がある場合はプログラムの起動を中止します。

タイムチェック無しの期間(時間)

指定した数値(時間)の期間、認証時間のチェックを行いません。

アドバンスオプション:

 ControlとAboutをメニュー追加

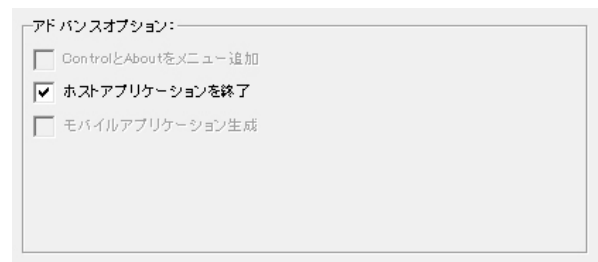
(使用不可)

 ホストアプリケーションを終了

暗号化するファイルがDLLファイルの場合、DLLファイルがプロテクトチェックに失敗すると、そのDLLファイルを使用しているホストアプリケーションが終了します。デフォルトではチェックが入っています。(推奨)

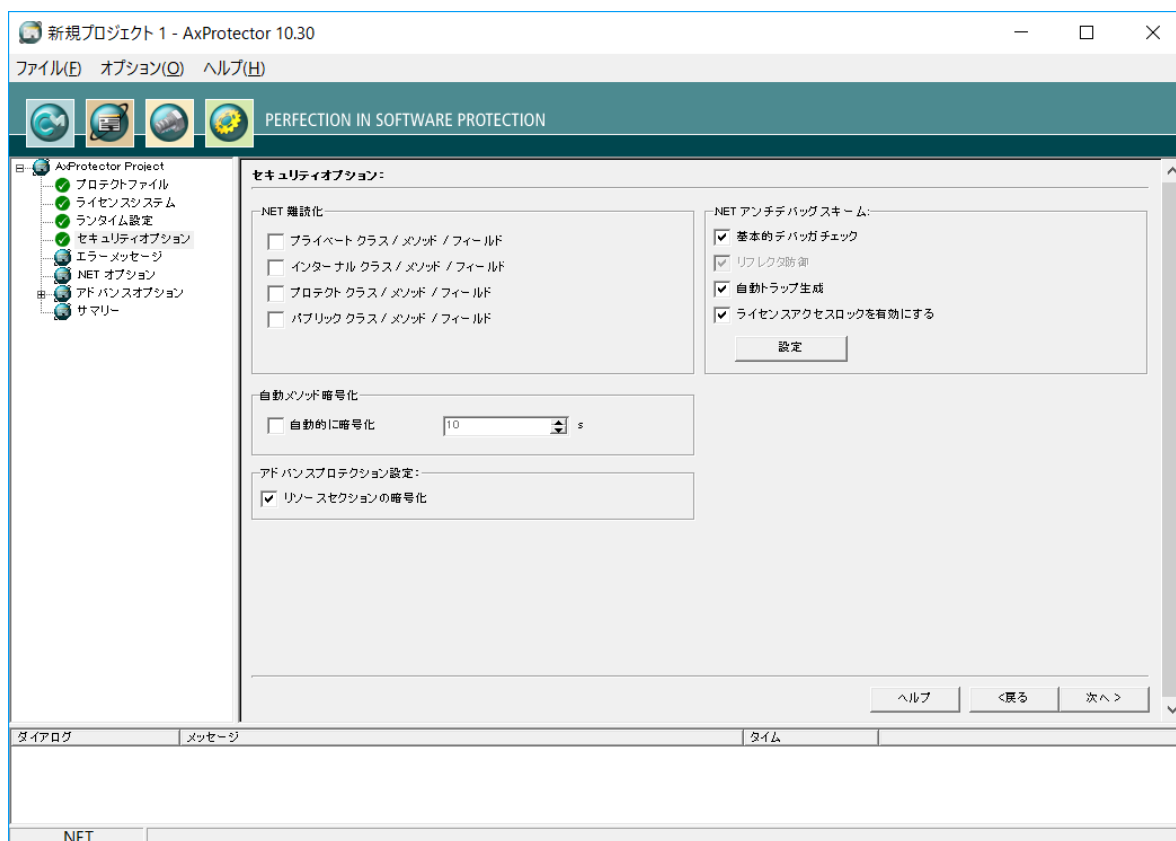
 モバイルアプリケーション生成

(使用不可)



4. セキュリティオプション

このセキュリティオプションは、プロテクト強度に関連する設定です。暗号化に対する解析強度を強化する場合に使用します。



.NET難読化：

各要素を難読化し解析不可能にします。難読化する要素にはクラス、メソッド、フィールドが含まれません。

プライベートクラス/メソッド/フィールド

プライベート要素を難読化します。

インターナルクラス/メソッド/フィールド

インターナル要素を難読化します。

プロテクトクラス/メソッド/フィールド

プロテクト要素を難読化します。

パブリッククラス/メソッド/フィールド

パブリック要素を難読化します。

[NOTE]

この.NET難読化オプションを選択しなくてもソースコードは難読化されます。このオプションは、解析をさらに複雑にするために各要素を難読化するためのものです。

.NETアンチデバッグスキーム：

デバッガープログラムはソフトウェアのエラーやバグを見つけるために役立ちますが、反面、ハッカーによる解析も可能にします。このスキームでは、デバッガープログラムに対するポリシーを設定します。

基本的デバッガチェック

デバッガーがアプリケーションにアタッチされているかをチェックし、デバッガーが検知された場合は、アプリケーションの起動を禁止するか、すでに起動している場合は終了させます。

リフレクタ防御

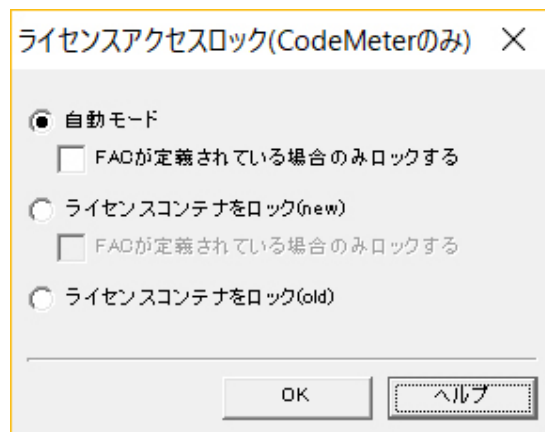
逆コンパイルを防止するために、リフレクタ防御が自動的に組み込まれます。

自動トラップ生成

ハッカーによる解析をより困難にするために、ハッカートラップが自動的に組み込まれます。

ライセンスアクセスロックを有効にする

デバッガーが検知されると、使用中のCmContainerのファームアイテムへのアクセスをロックします。ロックの内容は「設定」ボタンをクリックし、「ライセンスロック(CodeMeterのみ)」ダイアログ画面で設定します。



ファームウェアバージョン1.14以降の場合、デバッガーが検知されたら直ちにCmDongleをロックします。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)は、「用意されたロック」が行われます。「用意されたロック」とは、CmDongleを直ちにロックするのではなく、ファームアクセスカウンタ(FAC=Firm Access Counter)を利用し、デバッガーを検知するたびに値を"1"ずつ減らし、最終的に"0"になった時点でファームアイテム(Firm Item)がロックします。初期設定では、数値"65535"が割り当てられていますが、ソフトウェアベンダーによって変更することが可能です。ロックされたFirm Itemは、ソフトウェアベンダーのリモートプログラムによって解除することができます。

自動モード

ファームウェアバージョンが1.14以降の場合、ファームアイテムは直ちにロックされます。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)、FAC (Firm Access Counter) の値が"1"減ります。(デフォルト:推奨)

FACが定義されている場合のみロックする

このオプションにチェックを入れると、ファームウェアバージョンが1.14以降の場合、「用意されたロック」が設定されているかどうかをチェックし、設定されていた場合にファームアイテムをロックします。ファームウェアバージョンが1.14より以前の場合(1.14は含まない)、FACの値が1減ります。

○ライセンスコンテナをロック(new)

ファームウェアバージョンが1.14以降の場合、ファームアイテムが直ちにロックされます。このオプションがセキュリティ上一番望ましいのですが、必ずファームウェアは1.14以上が必要になります。

□FACが定義されている場合のみロックする

このオプションにチェックを入れると、ファームウェアバージョンが1.14以降の場合、ファームアイテムを直ちにロックします。同時に、「用意されたロック」が設定されているかどうかをチェックし、用意されていた場合、ファームアイテムをロックします。

○ライセンスコンテナをロック(old)

すべてのファームウェアバージョンが対象です。

「用意されたロック」が設定されていた場合、FACの値が1減ります。

自動メソッド暗号：

指定した数秒後に再暗号化を行います。

自動メソッド暗号

自動的に暗号化

0 s

アドバンスプロテクション設定：

リソースセクションを暗号化します。

アプリケーション起動時に、暗号化されたリソースがPCメモリーにロードされ、「オンデマンド」で復号されます。

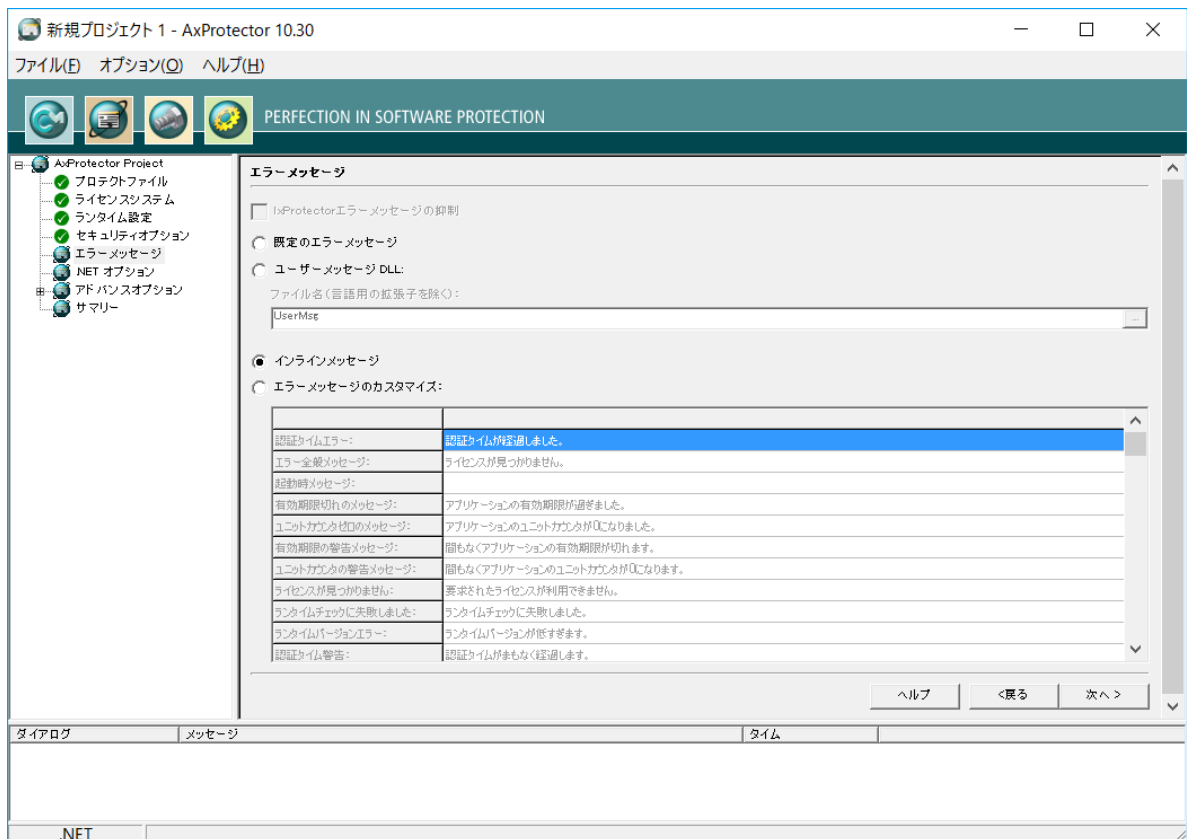
アドバンスプロテクション設定：

リソースセクションの暗号化

5. エラーメッセージ

エラーメッセージを作成します。.NETアセンブリの場合、エラーメッセージの作成には2通りの方法があります。

- 既定のエラーメッセージ
- インラインメッセージ



IxProtectorエラーメッセージの抑制
(選択不可)

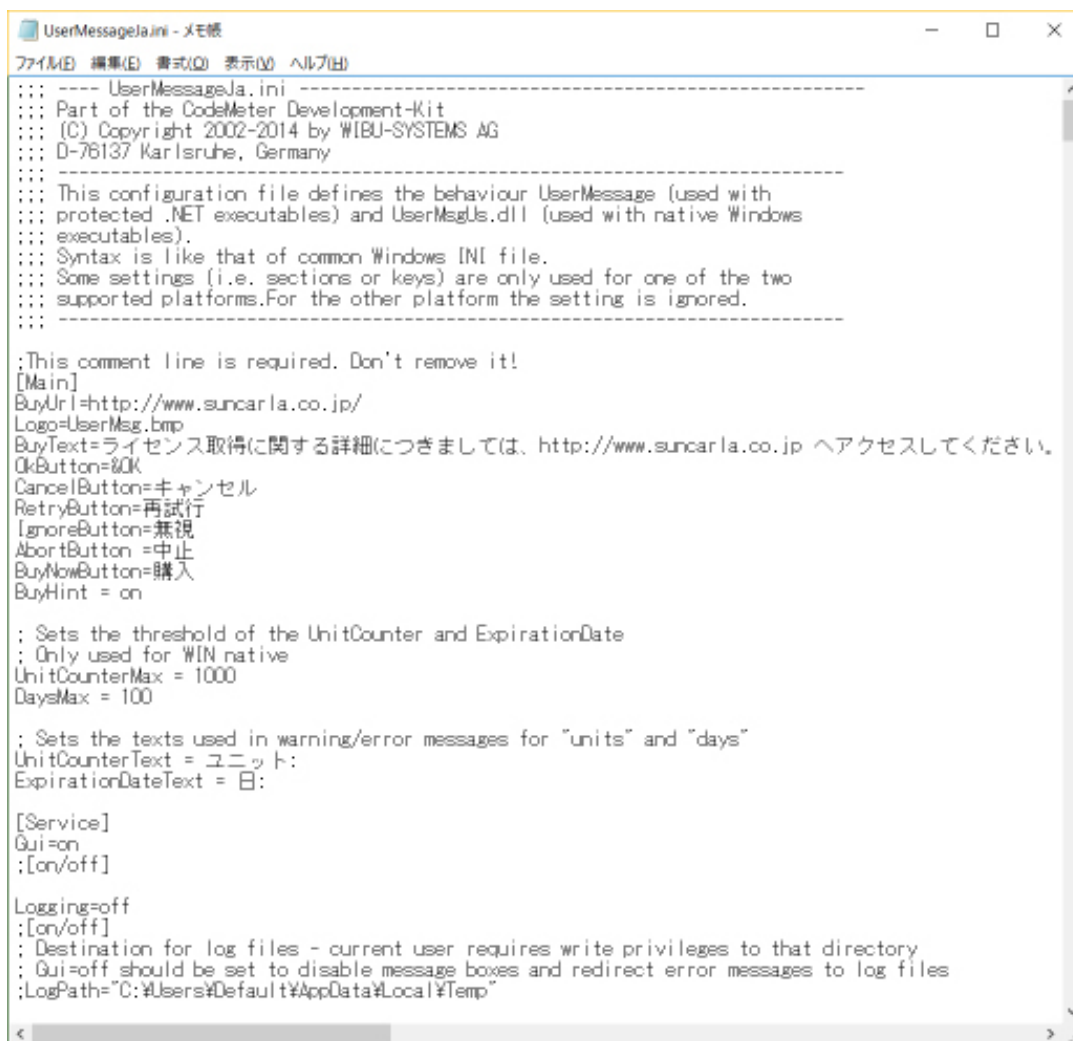
既定のエラーメッセージ
コードメータがあらかじめ用意しているエラーメッセージです。

ユーザーメッセージ DLL :
.NETアセンブリの場合、この「ユーザーメッセージDLL」を使用せずに、次の「インラインメッセージ」を使用してください。

インラインメッセージ :
BMP画像を含めたメッセージをカスタマイズすることができます。カスタマイズ作業は“UserMessageJa.ini” ファイルの内容を直接編集します。暗号化処理を行うと、暗号化されたプログラムと同じフォルダに“UserMessageJa.ini”が作成されます。この“UserMessageJa.ini”をメモ帳などのエディタで編集してください。編集した“UserMessageJa.ini” やBMP画像ファイルは暗号化されたプログラムと同じフォルダに保存する必要があります。該当するファイルが見つからなかった場合は、コードメータの「規定のエラーメッセージ」が表示されます。

【操作例】

1. 「UserMessageJa.ini」 ファイルをテキストエディタで開き、必要に応じてメッセージを変更します。



```
----- UserMessageJa.ini -----
: Part of the CodeMeter Development-Kit
: (C) Copyright 2002-2014 by WIBU-SYSTEMS AG
: D-76137 Karlsruhe, Germany
-----
: This configuration file defines the behaviour UserMessage (used with
: protected .NET executables) and UserMsgUs.dll (used with native Windows
: executables).
: Syntax is like that of common Windows [INI file].
: Some settings (i.e. sections or keys) are only used for one of the two
: supported platforms. For the other platform the setting is ignored.
-----

: This comment line is required. Don't remove it!
[Main]
BuyUrl=http://www.suncarla.co.jp/
Logo=UserMsg.bmp
BuyText=ライセンス取得に関する詳細につきましては、http://www.suncarla.co.jp へアクセスしてください。
OkButton=OK
CancelButton=キャンセル
RetryButton=再試行
IgnoreButton=無視
AbortButton=中止
BuyNowButton=購入
BuyHint = on

: Sets the threshold of the UnitCounter and ExpirationDate
: Only used for WIN native
UnitCounterMax = 1000
DaysMax = 100

: Sets the texts used in warning/error messages for "units" and "days"
UnitCounterText = ユニット:
ExpirationDateText = 日:

[Service]
Gui=on
:[on/off]

Logging=off
:[on/off]
: Destination for log files - current user requires write privileges to that directory
: Gui=off should be set to disable message boxes and redirect error messages to log files
: LogPath="C:\Users\%Default%\AppData\Local\%Temp"
```

2. 暗号化されたプログラムと同じフォルダに、「UserMessageJa.ini」と使用した画像ファイル (BMP)をおき、コードメータキーを装着しない状態でプログラムを実行するとエラーメッセージが表示されます。

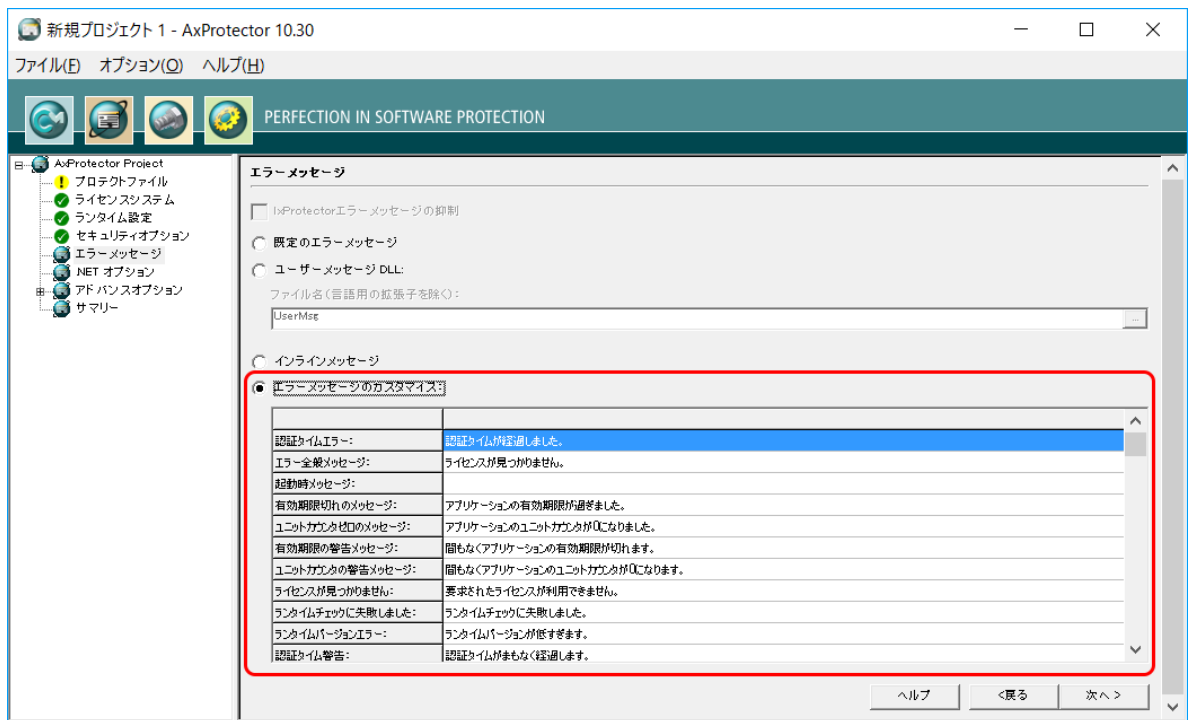


【NOTE】

暗号化されたプログラムを配布する際は、必ず「UserMessageJa.ini」も一緒に配布し、暗号化されたプログラムと同じフォルダに保存するようにしてください。

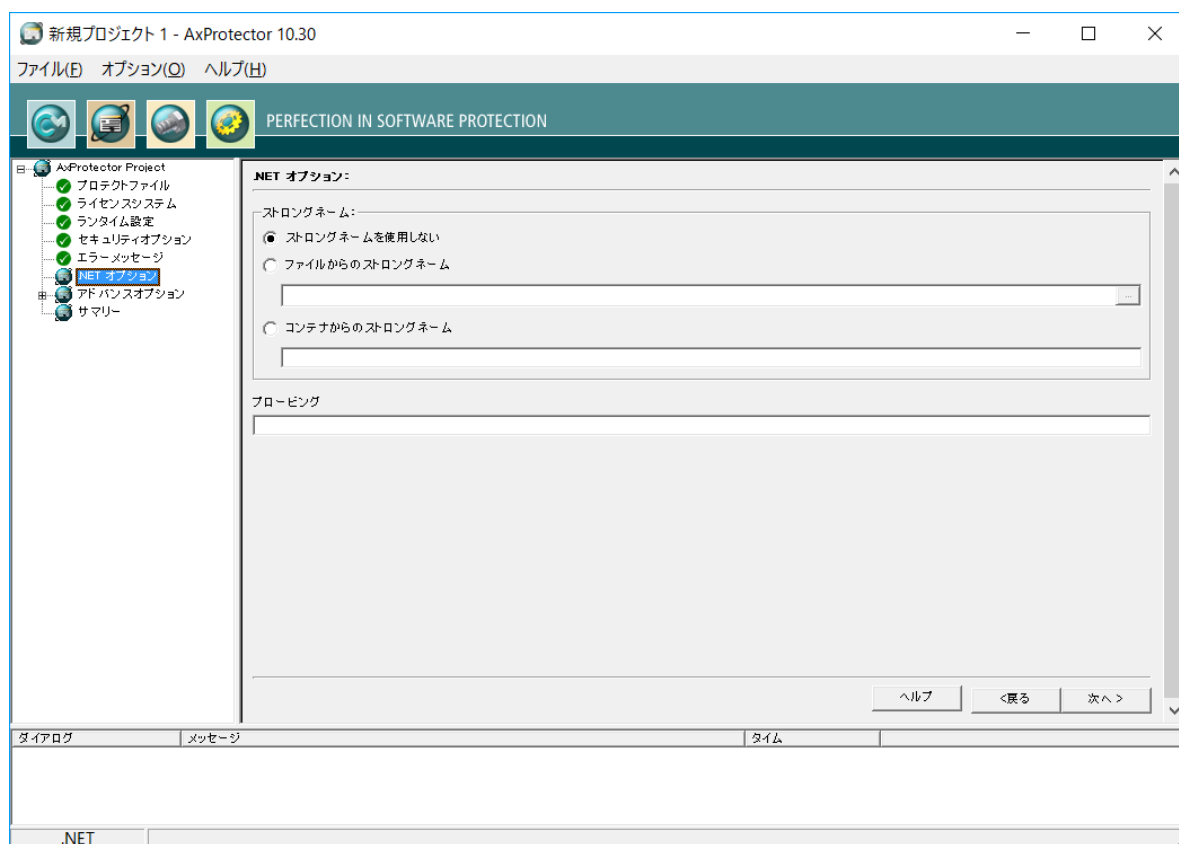
○ エラーメッセージのカスタマイズ：

.NETアセンブリの場合、「エラーメッセージのカスタマイズ」は使用できません。「インラインメッセージ」をご使用ください。



6. .NET オプション

AxProtectorによるデジタル署名を行うことができます。



○ ストロングネームを使用しない

AxProtectorによるデジタル署名は行いません。

○ キーペアファイルを指定してストロングネームを使用：

ファイルで定義されているキーペアでデジタル署名を行います。
キーペアが保存されているファイルを指定します。

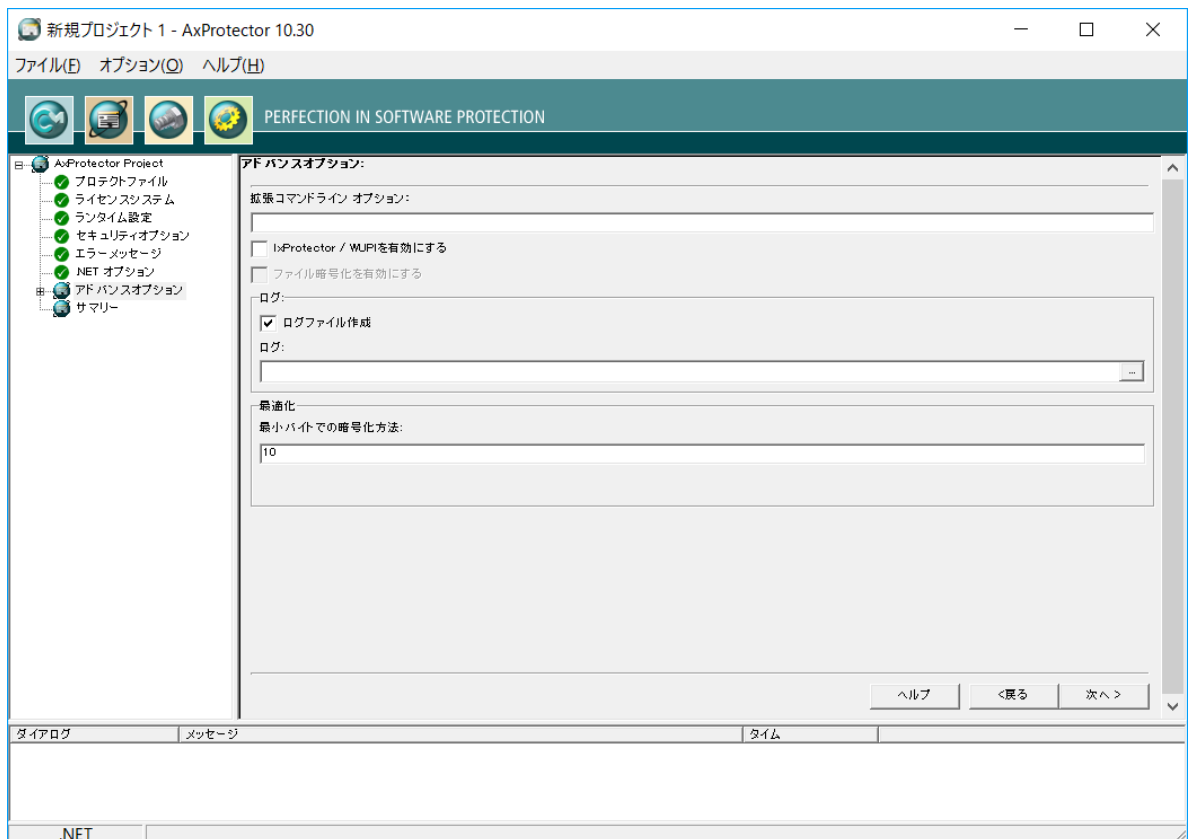
○ コンテナを指定してストロングネームを使用：

コンテナで定義されているキーペアでデジタル署名を行います。
コンテナに保存されているファイルを指定します。

プロービング：

デジタル署名を行うプログラム(アセンブリ)のパスを指定します。
複数ある場合は ";" で指定します。
また、app.config ファイルにまとめて指定することも可能です。

7. アドバンスオプション



拡張コマンドラインオプション

AxProtector操作画面から選択できない拡張機能をコマンドラインでサポートします。

IxProtector / WUPI を有効にする

IxProtector/WUPIを使用する場合は、チェックをいれます。

ファイル暗号化を有効にする (選択不可)

.NETアセンブリの場合、データファイル暗号化機能は使用できません。.NETアセンブリ上でデータファイルの暗号化/復号を実現するには、コードメータCore APIの暗号化ファンクションを利用します。(CmCrypt()など)

ログファイル作成

暗号化処理のログファイルを作成します。

ログファイル名

ログファイルが作成されるパスとファイル名を指定します。ファイル名だけを指定した場合は、
 %¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin
 に作成されます。

最適化

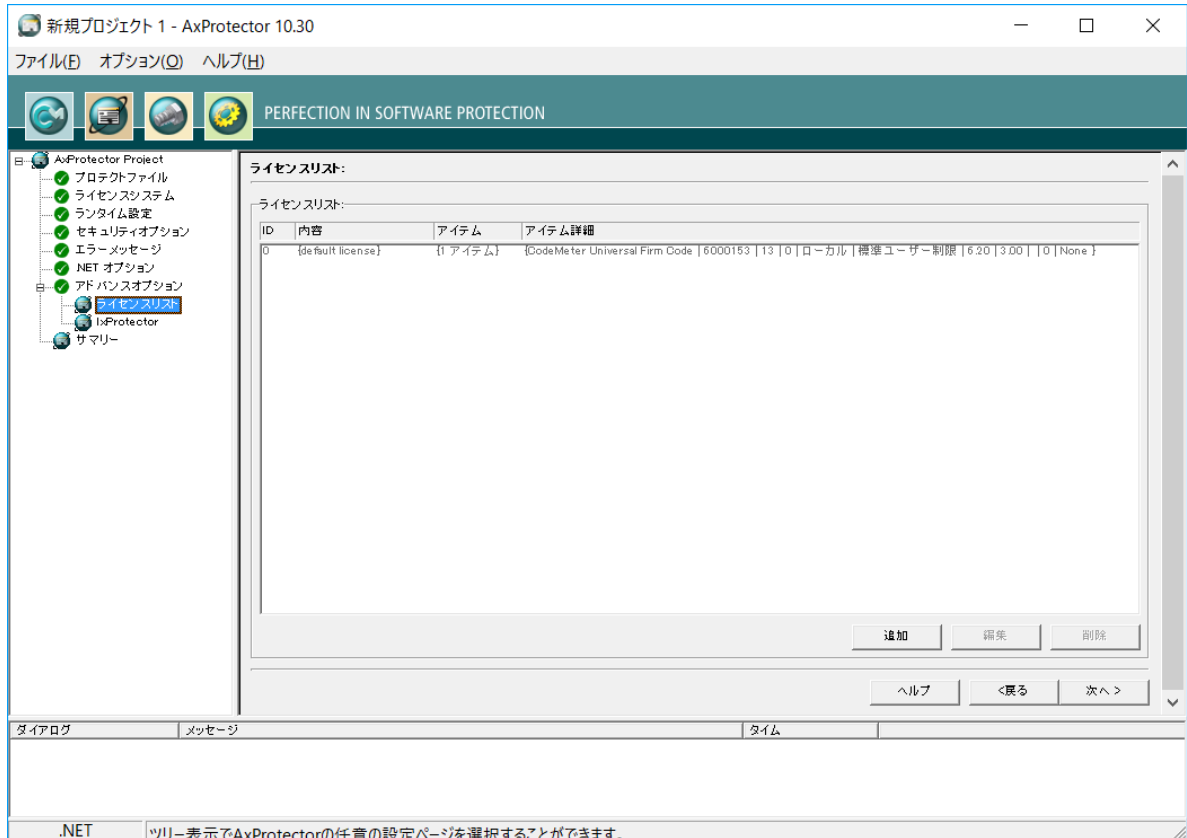
最小バイトでの暗号化方法:

暗号化対象になるメソッドの最小バイト数を指定します。(デフォルト値は10バイト)
 メソッドのサイズが指定したバイト数以下の場合は暗号化しません。

7-1. ライセンスリスト

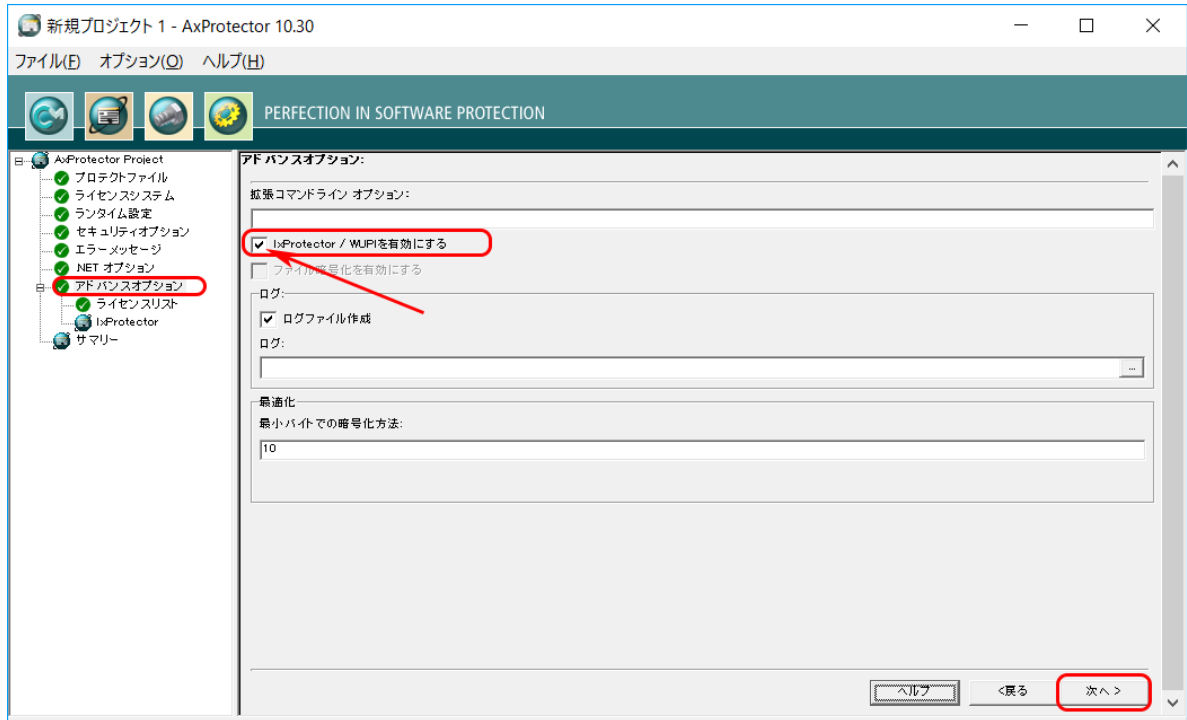
ライセンスリストは、1つのアセンブリから、複数のファームコード/プロダクトコード等のチェック、またはCmActLicenseやWibuKeyドングルのチェックを行う場合に利用します。

ライセンスリストに登録された内容を実行するには、IxProtector/WUPIを利用します。



① lxProtector/WUPI を有効にする

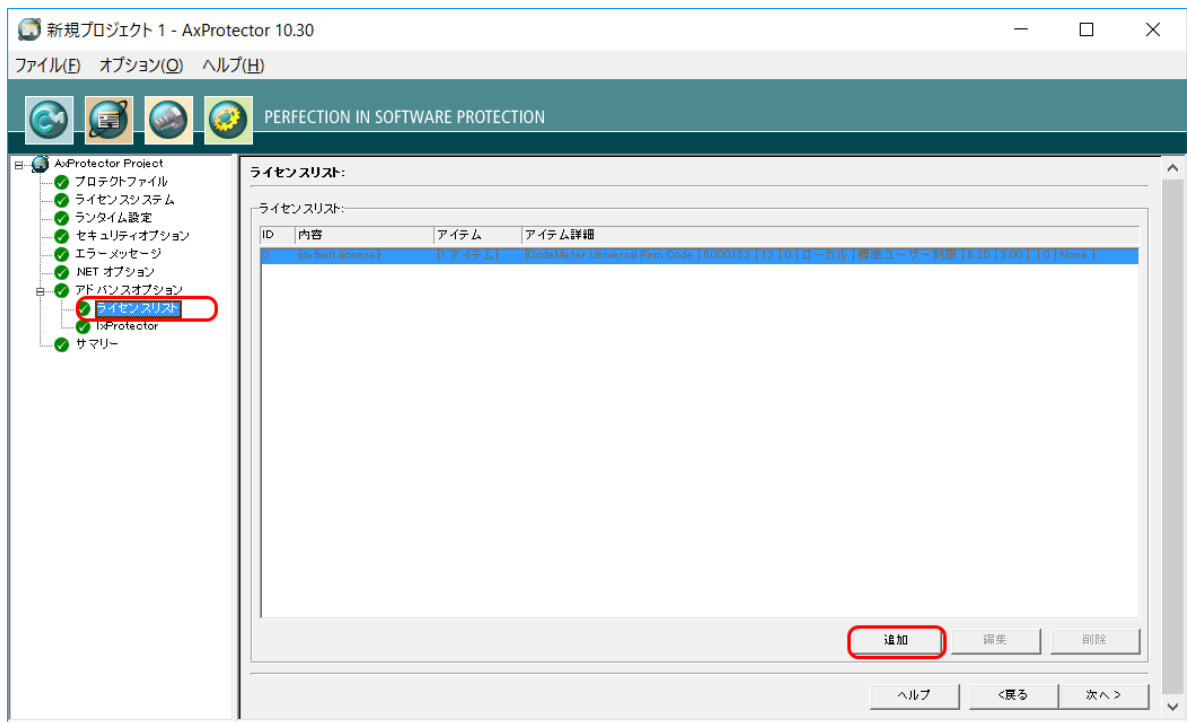
まず、アドバンスオプション画面で、「lxProtector/WUPIを有効にする」にチェックを入れ、「次へ」をクリックする。



② ライセンスリストを作成する

「ライセンスリスト」を選択し、「追加」ボタンをクリックします。

lxProtectorは、まずライセンスリストをId単位で作成し、そのIdを利用します。



③ ライセンス内容を登録する

Idに対してライセンス内容を登録します。「ライセンスリストの追加」画面で行います。

Id:

ライセンスIDを設定する。ライセンスIDは連番で自動表示されます。

内容:

ライセンス内容を記載します。内容はライセンス内容に応じて自由に記載してください。

[ライセンスアイテム詳細]

ライセンスシステム:

使用するライセンスシステムを指定します。コードメータ(CmDongle)、ワイブキー(Wibukey)、コードメータActライセンス(CmActLicense)が選択できます。

ファームコード:

ライセンスIDに割り当てるファームコードを設定します。

プロダクトコード:

ライセンスIDに割り当てるプロダクトコードを設定します。

フィーチャーコード:

ライセンスIDに割り当てるフィーチャコードを設定します。

サブシステム:

サブシステムを割り当てます。

「ローカル」

ローカルのみアクセス

「ネットワーク」

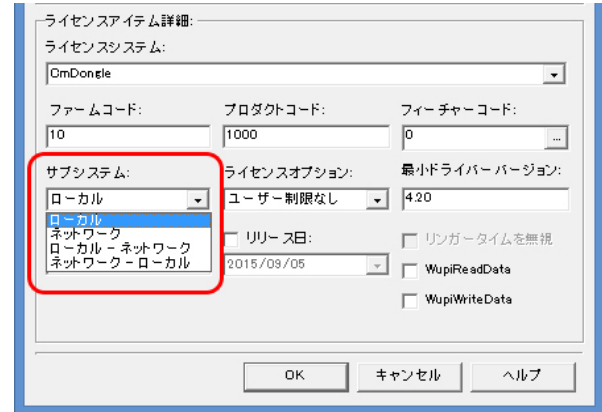
ネットワークのみアクセス

「ローカル-ネットワーク」

最初にローカルアクセス、次にネットワークアクセスを行う

「ネットワーク-ローカル」

最初にネットワークアクセス、次にローカルアクセスを行う



ライセンスオプション:

ライセンス形態を割り当てます。

「標準ユーザー制限」

アプリケーション単位で制限

「ステーションシェア」

クライアント単位で制限

「WK 互換モード」

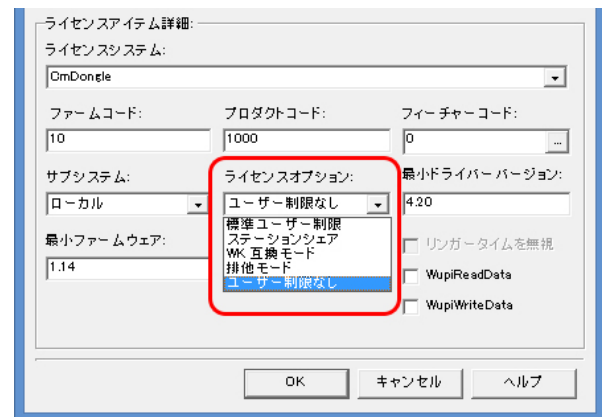
ネットワークでは標準制限、ローカルは制限なし

「排他モード」

クライアントごとに1つのアプリケーションのみ許可

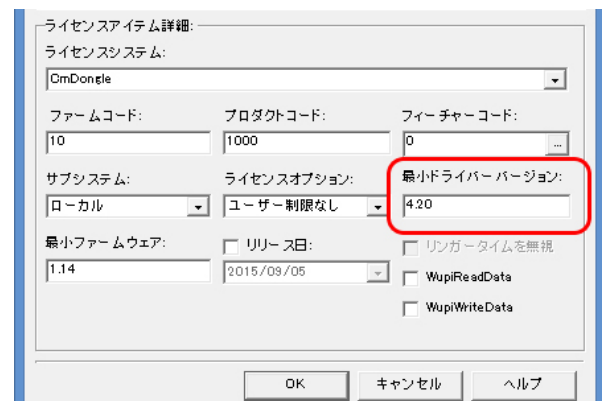
「ユーザー制限なし」

ユーザー無制限で使用可能



最小ドライバーバージョン:

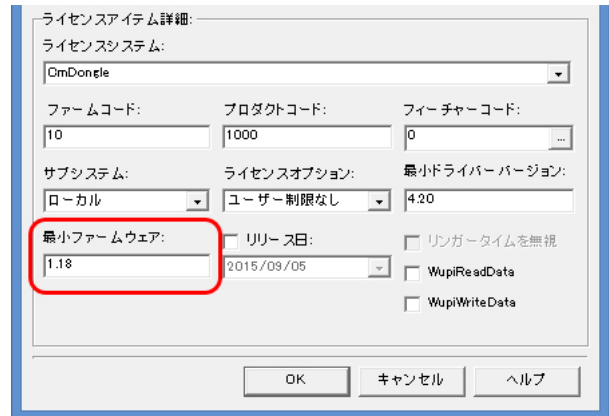
コードメータランタイムキットの最小バージョンを指定します。ここで指定したバージョンより古いバージョンがインストールされていると、暗号化されたプログラムは起動しません。コードメータランタイムキットは、OS環境に応じて定期的にバージョンアップしており、貴社プログラムが古いバージョンで動作し、誤動作するのを防ぐことができます。



最小ファームウェア:

CmDongleのファームウェアの最小バージョンを指定します。リリース日/メンテナンス期間機能など、古いファームウェアではサポートしていない機能を利用する場合は最新のファームウェアバージョンを指定します。「リリース日」機能を利用する場合は、最小ファームウェアバージョンを1.18以上にする必要があります。

CmDongleファームウェアのバージョンアップは、コードメータコントロールセンター上でユーザーが更新することができます。



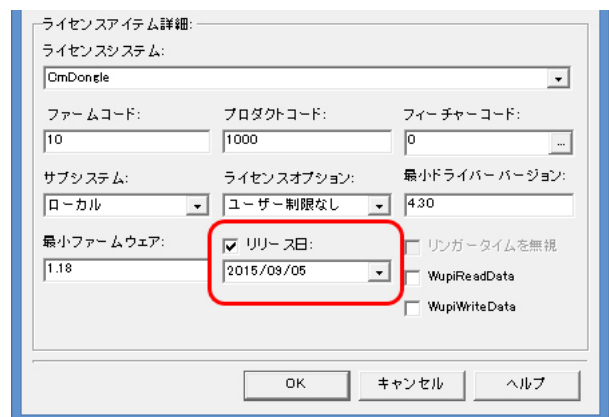
リリース日:

プログラムのリリース日を設定します。リリース日は、CmContainer内に登録されたメンテナンス期間と連動し、リリース日がメンテナンス期間内であればプログラムが起動し、メンテナンス期間以外の場合はプログラムが起動しません。

この機能は、ユーザーとのプログラムのメンテナンス契約をコントロールするために役立つ機能です。

[NOTE]

リリース日機能を利用するには、最小ファームウェアは1.18以上が必要になります。



リンガータイムを無視

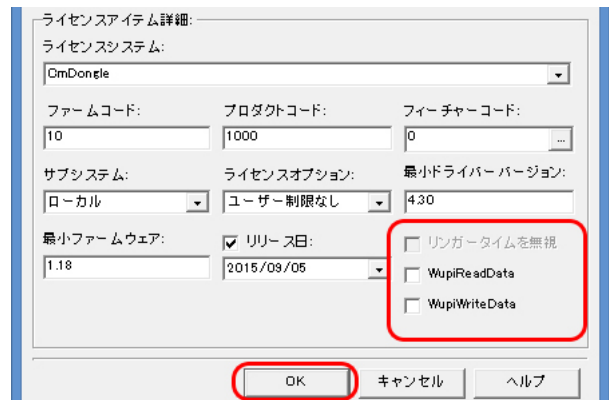
リンガータイムを無視する場合にチェックを入れます。(。NETアセンブリの場合、選択不可)

WupiReadData

CmContainerからデータをリード(Read)する場合はチェックを入れます。

WupiWriteData

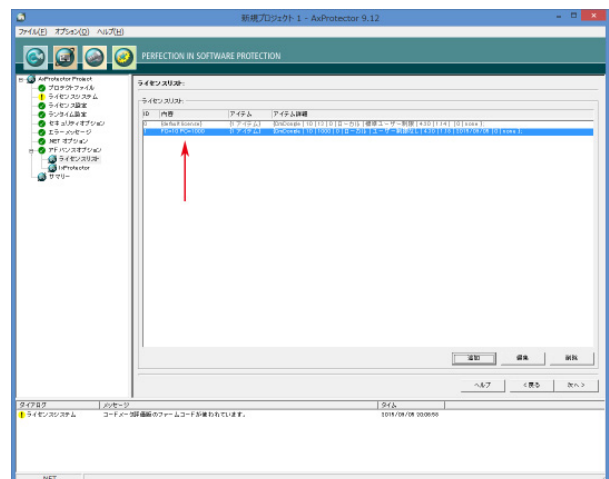
CmContainerからデータをライト(Write)する場合はチェックを入れます。



「OK」ボタンをクリックするとライセンス内容がライセンスリストに追加登録されます。

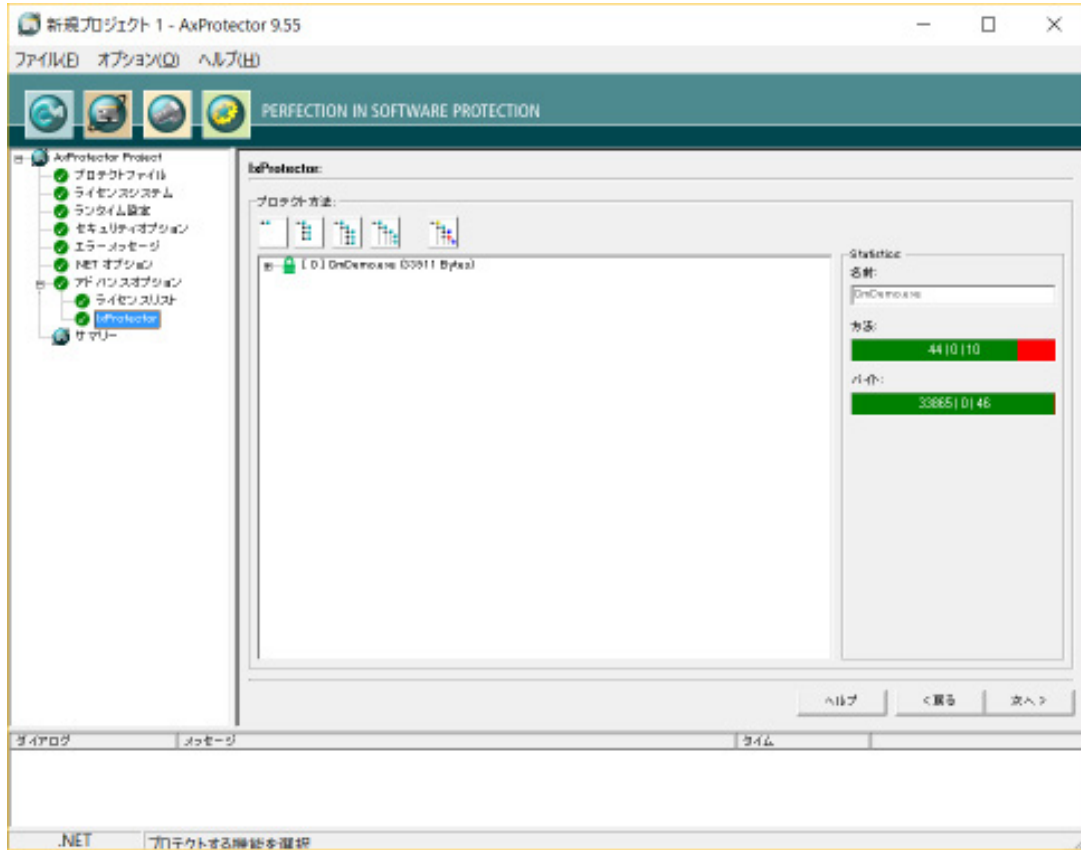
[NOTE]

異なるライセンスリストを複数登録することが可能です。

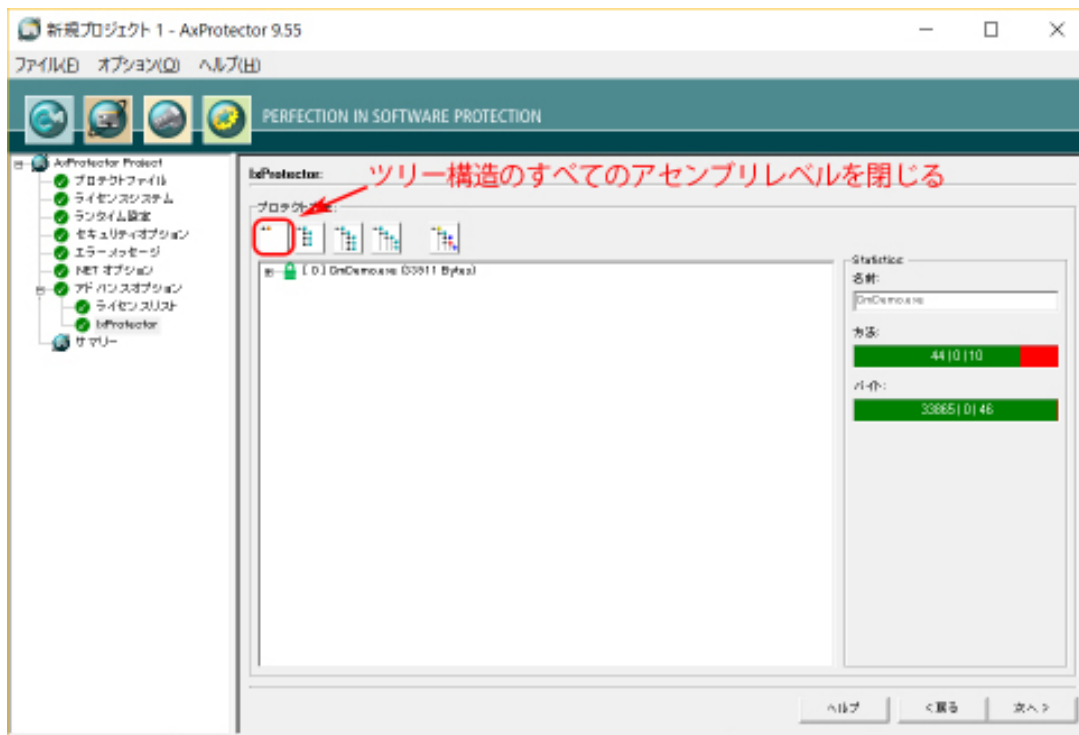


7-2. IxProtector

ソースコードの中の、それぞれ個別のネームスペース、クラス、メソッド単位で、それぞれ個別のライセンスリストを割り当てて暗号化することができます。アセンブリファイル全体の暗号化とは別に、モジュール単位で個別のライセンスを割り当てる場合に役立ちます。

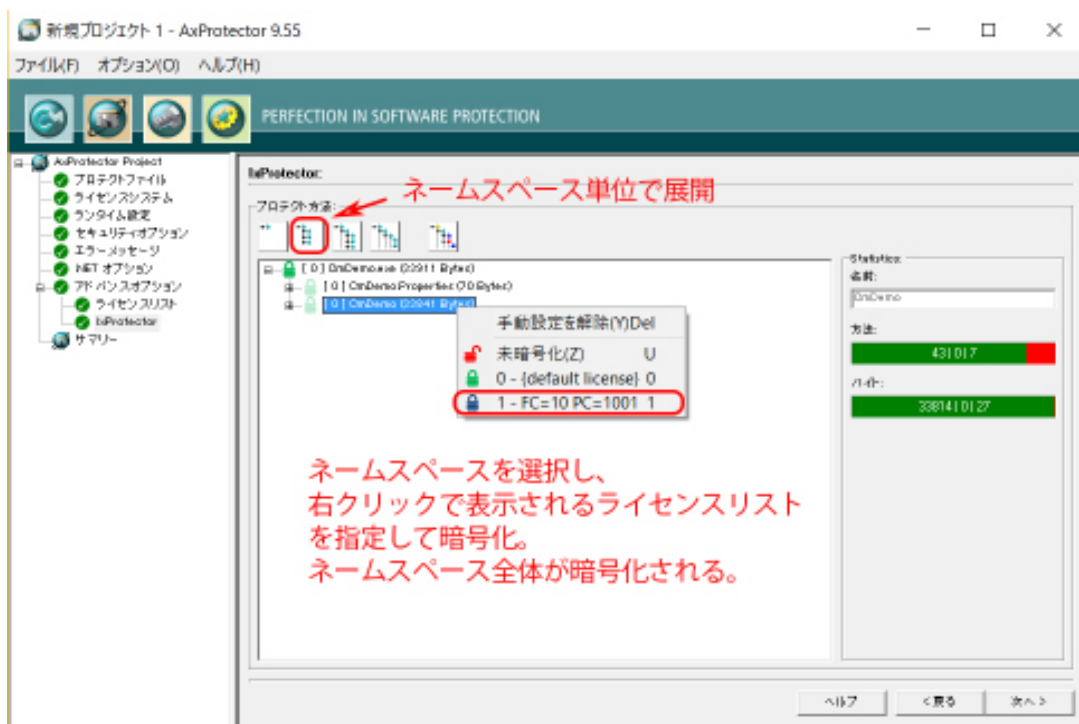


① ツリー構造のすべてのアセンブリレベルを閉じる



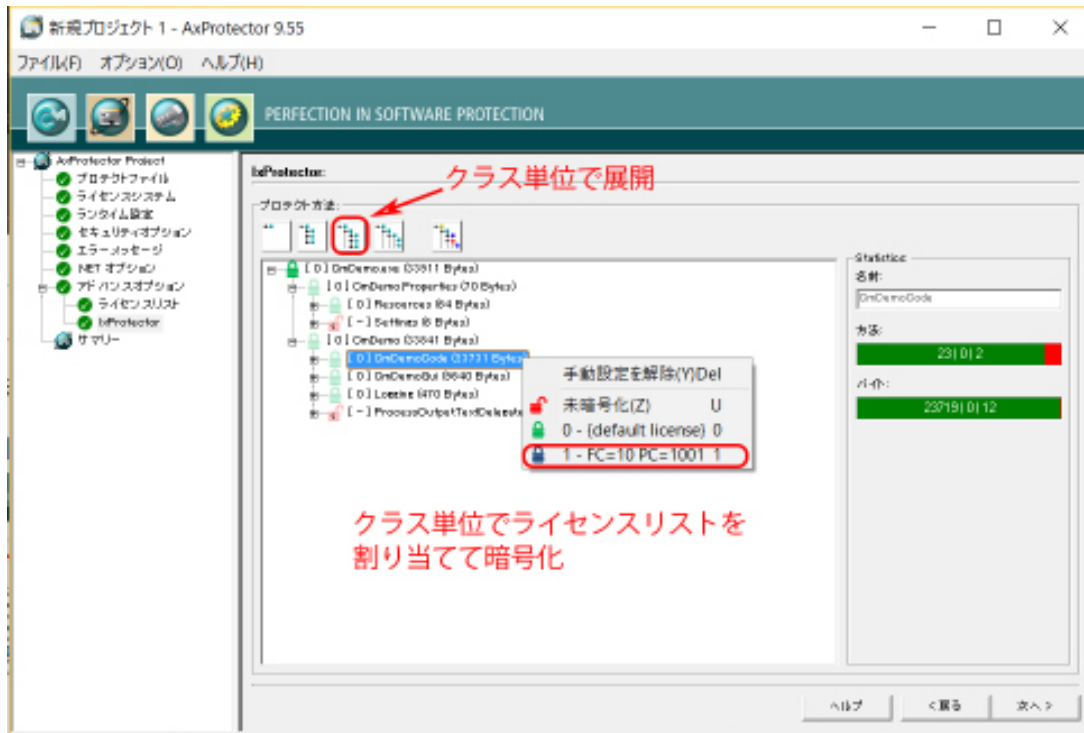
② ネームスペース単位で暗号化する

ネームスペース単位で展開されます。各ネームスペース単位で、指定のライセンスリストを割り当てて暗号化します。



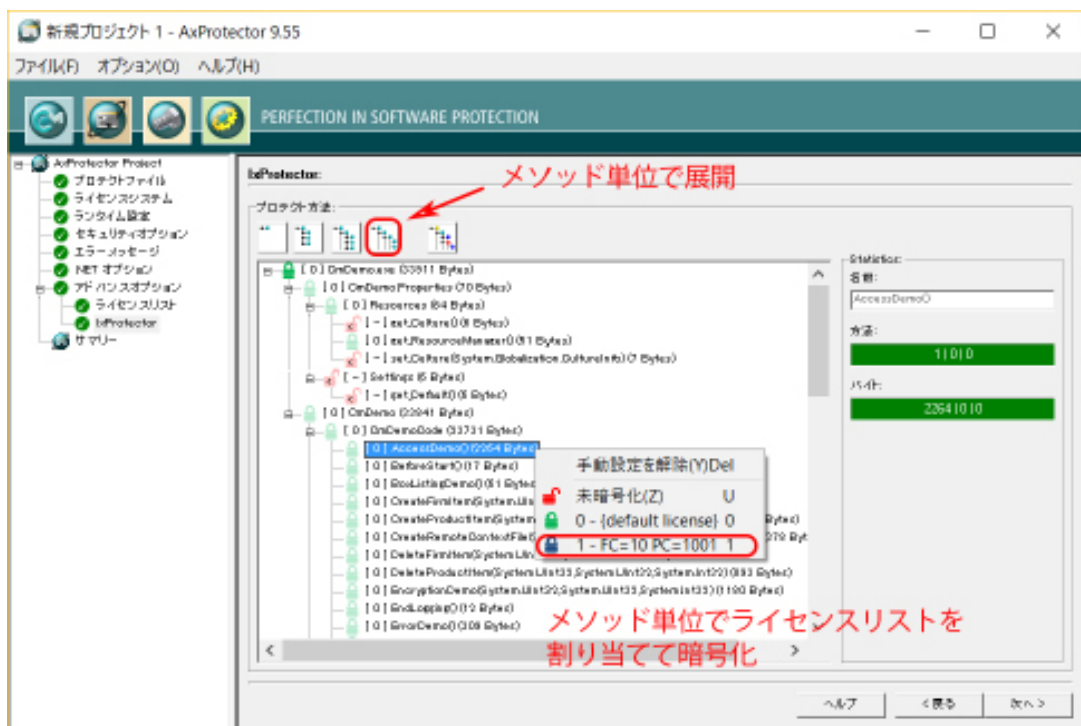
③ クラス単位で暗号化する

クラス単位で展開されます。各クラス単位で、指定のライセンスリストを割り当てて暗号化します。



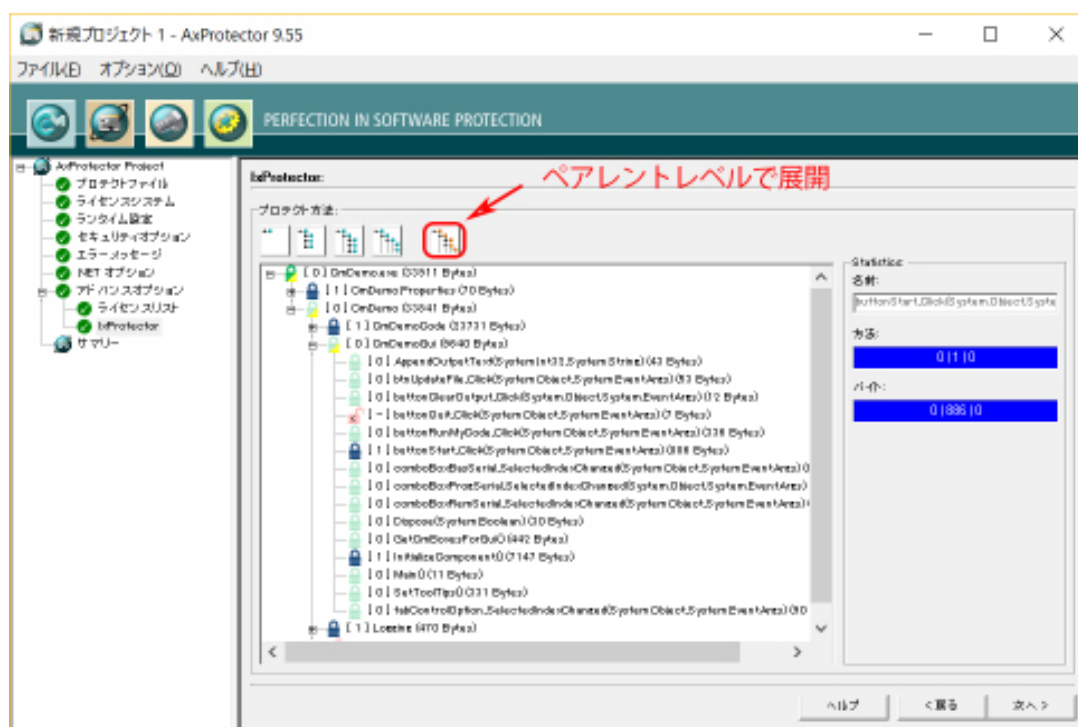
④ メソッド単位で暗号化する

メソッド単位で展開されます。各メソッド単位で、指定のライセンスリストを割り当てて暗号化します。



⑤ ペアレントレベルで展開し、ライセンスリストを割り当てたレベルを表示する

ペアレントレベルで展開し、ライセンスリストを割り当てたレベルを表示します。



[NOTE]

各ネームスペース、クラス、メソッドに割り当てたライセンスリストは、実行されるたびにライセンス数が1つ消費されます。同じライセンスリストを複数割り当てる場合は、CmDongle側に登録するライセンス数を十分考慮してください。なお、各ネームスペース、クラス、メソッドが終了するとライセンス数は元に戻ります。

8. サマリー：

設定したセキュリティ内容の概要を表示します。設定内容は、コマンドラインによるオプションパラメータで表示されます。

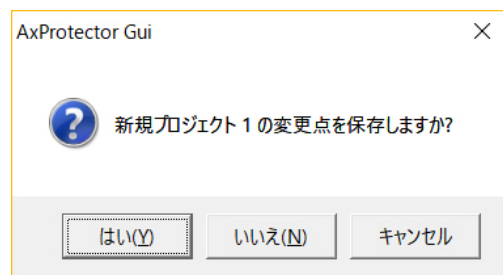
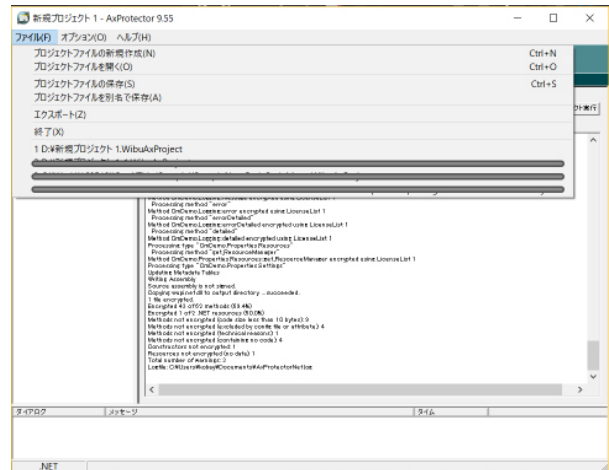
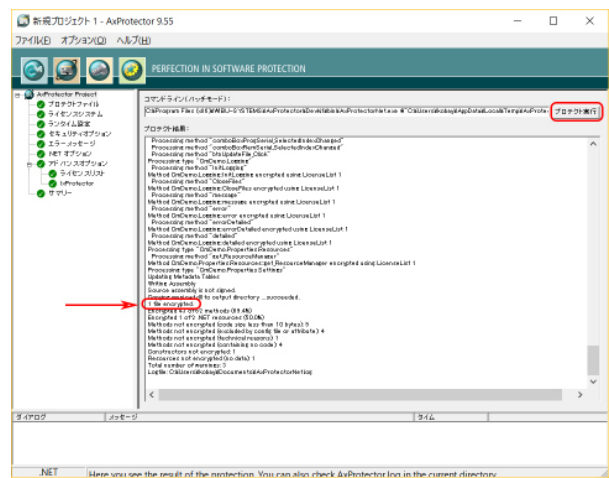
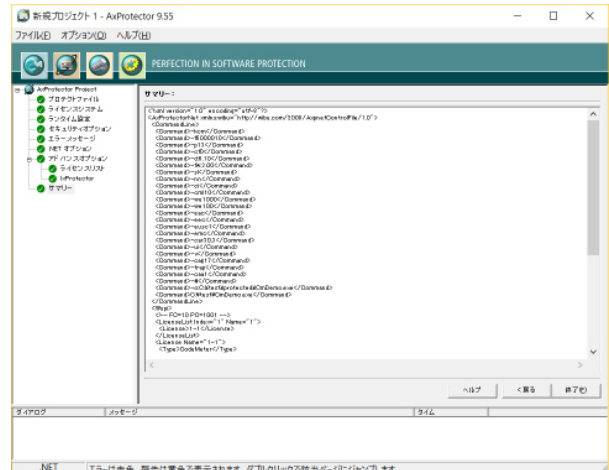
設定に問題がなければ、右下の「終了」ボタンがアクティブになります。もし、設定に問題がある場合は、「終了」ボタンが非アクティブの状態になります。

「終了」ボタンをクリックすると、暗号化処理が開始されます。暗号化処理に成功すると、「プロテクト結果」画面に"1 file encrypted!"が表示されます。

また、プロテクトオプションのコマンドラインからは、設定したオプションパラメータが表示されます。このコマンドラインを直接入力して、オプションパラメータを変更および追加をして暗号化処理を行うことも可能です。右部の「プロテクト実行」ボタンをクリックすると暗号化処理が開始されます。

[ファイル]メニューから[終了]を選択してAxProtectorを閉じます。

閉じる際、プロジェクトファイルを保存するか画面が表示されます。プロジェクトファイルとは、各項目で設定したセキュリティ内容を保存しておくものです。どのファイルをどのようなセキュリティ内容で暗号化したかの管理ができます。また、プロジェクトファイルを直接開くことで、最初から各項目を入力する手間が省けます。



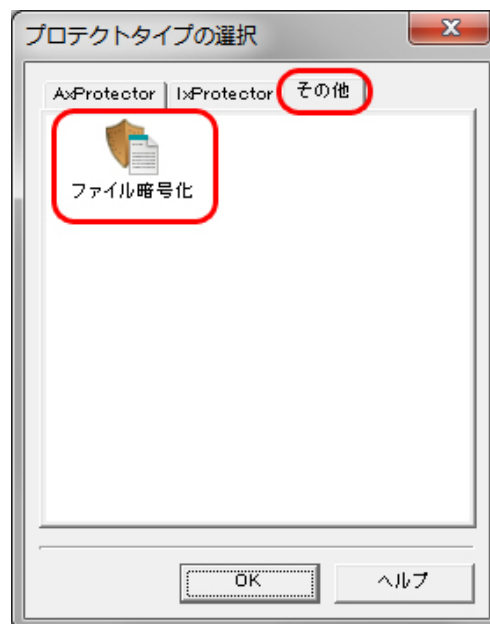
4-6. データファイルを暗号化する

データファイルを暗号化する場合は、AxProtectorの"ファイル暗号化"機能を使用します。AxProtector起動後、「プロテクトタイプの選択」で「その他」タブを選択し、「ファイル暗号化」を選択して"OK"ボタンをクリックするか、「ファイル暗号化」のアイコンをダブルクリックします。

この「ファイル暗号化」機能で暗号化可能なファイルは、txtなどのテキストファイル、Flash, WMV, MPEGなどの動画ファイル、その他のドキュメントファイルです。

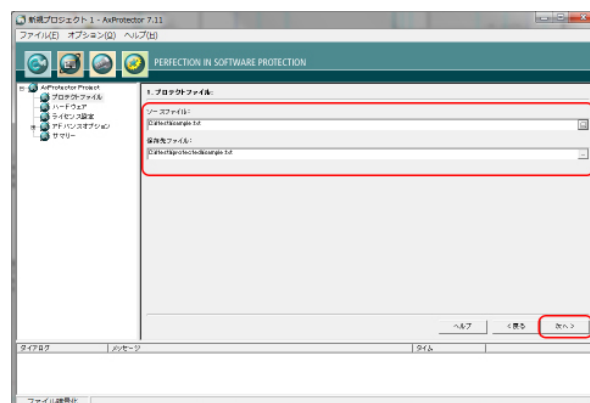
ただし、暗号化されたファイルを開くプログラム(exe等)にあらかじめAxProtectorで暗号化処理する必要がありますので、Excel.exeやMedia Player等、ライセンス上暗号化処理ができない場合は例外になります。

あくまで、貴社のプログラムから利用するデータファイルが対象になります。



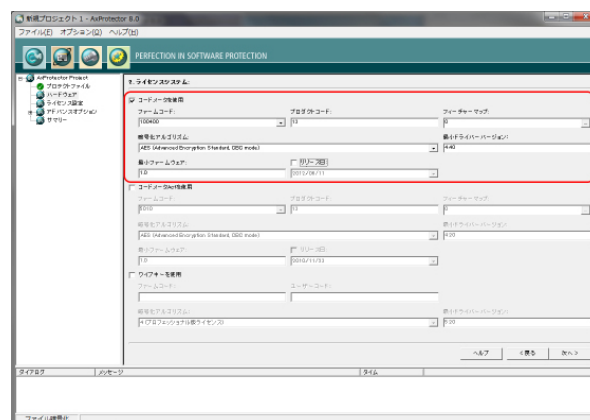
① ファイル名を入力する

「プロテクトファイル」画面で暗号化するファイル(ソースファイル)と暗号化後に作成されるファイル(保存先ファイル)を指定します。ソースファイルと保存先ファイルがフォルダも含めて同じ場合は上書き保存されますのでご注意ください。



② ファームコード、プロダクトコードを入力する

「コードメータを使用」にチェックを入れ、ファームコード、プロダクトコード、フィーチャーコード、暗号化アルゴリズム、最小ドライババージョン(ランタイムバージョン)等を設定します。



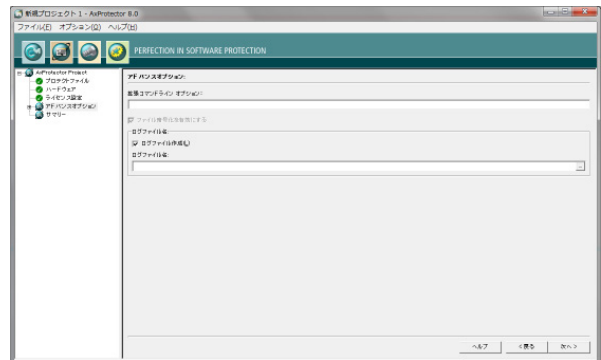
③ ライセンス設定を行う

データファイルの場合は、ネットワーク上のコードメータキーをサーチしないため、必ず"ローカル"で使
用します。また、ライセンスオプションは、"標準的な
ユーザー制限(アプリケーション単位で制限)"を選
択してください。



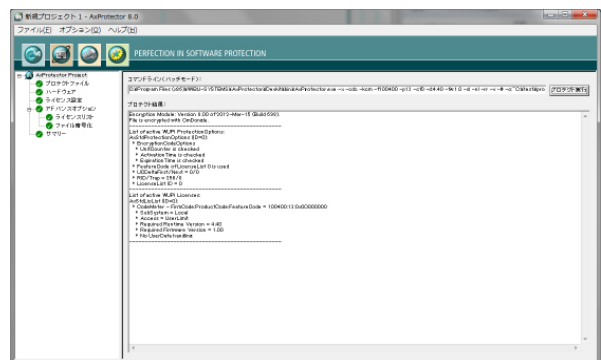
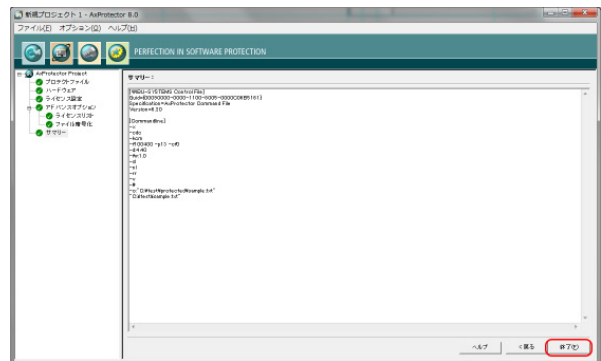
④ アドバンスオプション

アドバンスオプション画面では特に設定する項目は
ありません。また、次のライセンスリストおよびファ
イル暗号化オプションでも設定する項目はありませ
ん。「次へ」ボタンで進めます。



⑤ 暗号化処理を開始する

サマリー画面で「終了(F)」ボタンをクリックすると暗
号化処理が開始されます。保存先のフォルダに暗号
化されたファイルが存在するか確認してください。



4-7. コマンドラインでの使用方法

AxProtectorをコマンドラインで使用し、ファイルを自動暗号化することができます。ファイルを大量に暗号化したり、一連の暗号化作業をバッチ処理するのに役に立ちます。コマンドラインで使用する場合は、AxProtector.exeを使用します。なお、パラメータは大文字・小文字どちらも使用可能です。

axprotector.exe options filespec[/s]

以下は、オプションパラメータ設定の内容です。

/R([F:s][R][V])

暗号化に使用するランダム値の生成方法

F: 固定値を使用

R: ランダム値を使用 (デフォルト)

V: ファイルバージョン情報を使用 (Win32/64)

/X

スタティックライブラリを暗号化するプログラムにリンクする。ダイナミックライブラリ (デフォルト) を使用するよりも強固なセキュリティを実現できます。

/A

暗号化設定:

/K([CM][WK])

コードメータによる暗号化[CM]か、ワイプキーによる暗号化[WK]かを指定します。

CM: コードメータによる暗号化 (デフォルト)

WK: ワイプキーを使用

-1 暗号アルゴリズム 1

-2 暗号アルゴリズム 2

-3 暗号アルゴリズム 3

-4 暗号アルゴリズム 4

-5 暗号アルゴリズム 5

/Fx

ファームコード(Firm Code)を指定

/Px

プロダクトコード(Product Code)を指定

/CFx

フィーチャーコード(Feature Code)を指定

/D:v

最低ドライババージョン (デフォルト CM:3.20, WK:5.20)

/N[C[A]]L[A][N]S[X[A]]

ネットワークアクセスの指定

C: コンビニエントモード(簡易モード)

L: 標準的なユーザー制限(ノーマルユーザー制限)

N: ユーザー制限なし(No User Limit)

S: ステーションシェアモード

X: 排他的モード(Exclusive Mode)

/S[(L],[N])C]

サブシステムのサーチ順序

L: ローカルサブシステム(Local)

N: ネットワークサブシステム(LAN)

C: 最初にローカルアクセス、次にLANアクセスする。起動アプリケーションがネットワークドライブにある場合は、最初にLANアクセスする。

/CA[A[I],[G[I,1]],L],[M],[R[t],[m]],S[p],[T[t]],V]

A: アドバンスプロテクトスキームを有効にする(APS)

<l> レベル[0,15]を指定

レベル 0 (Level 0): APSを使用しない。

レベル 1(Level 1): リソースセクションを暗号(APS 1)

レベル 2(Level 2): スタティックコードの変更(APS 2)

レベル 4(Level 4): ダイナミックコードの変更(APS 3)

レベル 8(Level 8): 拡張スタティックコードの変更(APS 4)

D: -cdで暗号化されたファイルの自動復号を有効にする

E: プラグアウト検知を有効にする(CodeMeterのみ)

G: アンチデバッグチェック(ADC)を有効にする

<l> レベル(Level)[0,15]を指定

Level 0: デバッガーチェックをしない

Level 1: シンプルデバッガーチェック(ADC 1)

Level 2: カーネルデバッガーチェック(ADC 2)

Level 4: アドバンスデバッガーチェック(ADC 3)

Level 8: デバッガーの禁止。ただし、IDEは起動する。(ADC 4)

Level 16: デバッガーが見つかった場合はCmStickエントリを無効にする(ADC 5)

Level 32: デバッグを行うとプロテクトされたアプリケーションが無効になる(ADC 6)

Level 64: バーチャルマシン上でのアプリケーション起動を禁止する(ADC 7)

L: 暗号化する範囲を制限する

M: 'Control'と'About'メニューをシステムメニューに追加する

R: ランタイムチェックを設定する

<t> 秒を指定。デフォルトは300秒(5分)

<m> Max Ignore回数(無視の回数)を指定。デフォルトは3回。

S: 暗号化するサイズを変更する

<p>=[0..100]% デフォルト値は75%。

T: ボックスタイム(Box Time)と認証タイム(Certified Time)に<t>時間以上差がある場合は、認証タイム(Certified Time)をセットする。

<t>=時間

V: 暗号化されるプログラムにウイルスチェッカーを付加する。

/CC[M]

互換性を設定する。

M: 暗号化されたプログラムは、wibucrt32/64.dllをロードする。(msvcr*.dllのロード問題を解決する)

/CD(K([CM])[WK])F[x][P[y]])

ファイルを暗号化し、暗号パラメータのついたヘッダーを付加する。ファイルは、自動暗号化されたアプリケーションから自動的に復号化される。

K: コピープロテクションシステムの選択(コードメータまたはワイブキー)

F: アプリケーションがプロテクトされたファームコード(Firm Code)を指定。(アプリケーションから暗号化されたファイルを開く場合)

P: アプリケーションがプロテクトされたプロダクトコード(Product Code)を指定。(アプリケーションから暗号化されたファイルを開く場合)(ファームコードはすでに指定されている必要があります。)

/CI[D|N]

IxProtectorを使用するために、アプリケーション内の定義したコード範囲を暗号化する。

D: WupiEngineモジュールモードの使用を指定する。この場合、WupiEngineモジュールはプロテクトされたアプリケーションと一緒に配布する必要があります。

N: エラーが起こった場合、エラーメッセージを表示しないように設定する。

/CP[L]

アプリケーションをCmStickから起動した場合、アプリケーション終了後に生成されたすべてのファイルとレジストリエントリを削除するクリーンアップメカニズムをインストールする。

L: 削除されたエントリをすべてログファイルに残します。

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

暗号化/復号化プロセスにおける追加的チェック

A: アクティベーションタイムチェックを有効にする

F: ファームアクセスカウンタ(FirmAccessCounter)のデクリメント(減少)を有効にする

E: 有効期限(Expiration Time)チェックを有効にする

T: CmStickのPowerOn時に、認証タイム(Certified Time)更新を行う。このフラグは、有効期限(Expiration Time)チェックが有効になっている場合のみ有効。

U: ユニットカウンタ(Unit Counter)のチェックと減少を有効にする

<n>デクリメント数(デフォルトは1)

S: アプリケーション起動時のみチェック&減少

R: ランタイムチェックごとにチェック&減少(RはSを含む)

C: AT/ET/UCをチェックする(利用可能な場合)

I: AT/ET/UCを無視する

R: AT/ET/UCを要求する

/RIDx

RID変数を指定。RID=0の場合、デフォルト値(256)が設定される。

/G[0,l]:"Maker",l]

暗号化しない範囲を指定します。

<o>範囲の始めにファイルオフセットを指定する。

<l>暗号化しない範囲の長さを指定する。(Win32/64のみ)

"Marker"は、コード内に暗号化しない範囲の開始をロケートするテキストマーカーを指定します。

/W[E[t]][U[c]]

警告の開始点

E: 有効期限 (Expiration Time)の開始点を指定<t>

U: ユニットカウンタ(Unit Counter)の開始点を指定<c>

Chapter 5

IxProtector/WUPI について

- 5-1. IxProtector とは
- 5-2. WUPI ファンクションについて
- 5-3. WUPI ファンクション一覧
- 5-4. WUPI ファンクションの使い方
- 5-5. WUPI ファンクション詳細

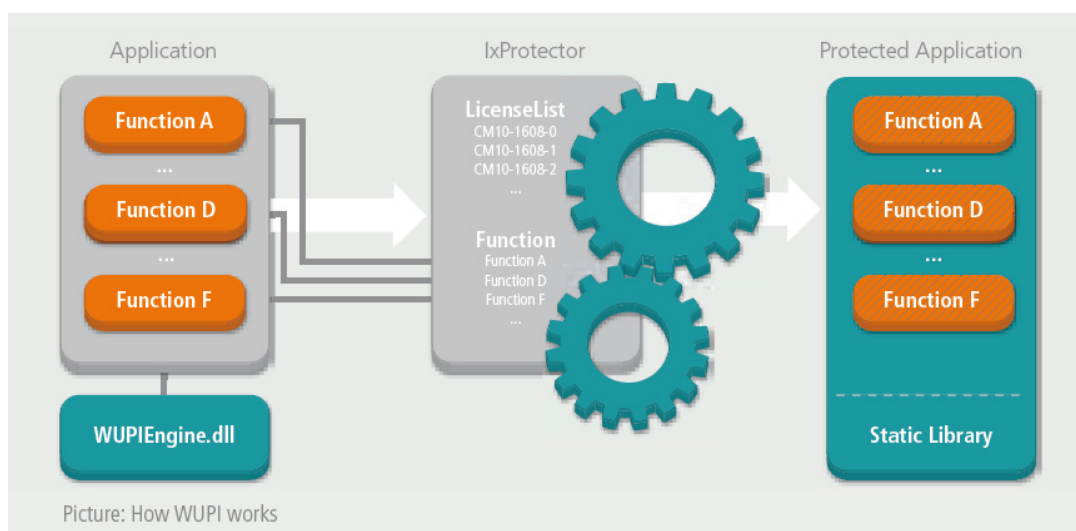
5-1. IxProtector とは

一般に市販されている暗号化ツールでプログラムファイルを暗号化した場合、暗号化されたファイルはディスク上では暗号化された状態を維持します。ただし、プログラムファイルが動作するために一度PCメモリーにロードされ、プロテクトチェックが通過した後は、暗号化されていたプログラムファイル（コード）はメモリー上で復号された状態になります。この状態で、クラッキング解析を行うことでセキュリティポイントをはずし、復号されたコードをそのまま複製することが可能になります。

コードメータの場合は、デバッガー解析などの攻撃に耐える強力なセキュリティ機能を備えておりますが、メモリー上でプログラムコードが復号されていること自体がセキュリティホールの一つになります。

コードメータのIxProtectorは、この問題を解決しました。メモリー上で展開されるプログラムコードを常に暗号化しておき、必要な時に必要なモジュールだけを復号して実行するという、メモリー上での「オンデマンド復号」を実現する機能です。AxProtectorで暗号化されたプログラムコードが、メモリー上でも常に暗号化されているため、クラッキングに対して非常に強力なセキュリティを実現することが可能になります。

IxProtectorによる「オンデマンド復号」セキュリティを実現するには、WUPI(Wibu Universal Protection Interface)ファンクションをソースコードに組み組み、ファンクションモジュール単位で暗号化・復号を行います。

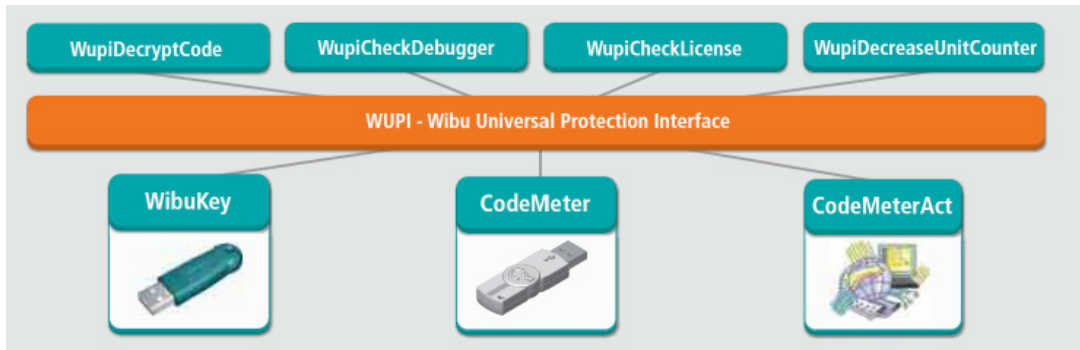


また、.NETアセンブリおよびJavaアプリケーションの場合、AxProtectorで自動暗号化することにより、このIxProtector/「オンデマンド復号」機能が自動的に組み込まれます。従い、WUPIファンクションを使用するためにソースコードを修正する必要がありません。

5-2. WUPI ファンクションについて

WUPI (ウーピー)とは、Wibu Universal Protection Interfaceの略で、ワイブキー(Wibukey)、コードメータ(CodeMeter)、コードメータAct(CodeMeterAct)に共通に使用できるユニバーサルなAPIファンクションです。WUPIファンクションで取得したハンドルを、ネイティブの各APIファンクションに渡してWUPIファンクションと従来のAPIファンクションを連携して使用することも可能です。

さらに、WUPIファンクションによって暗号化されるファンクションモジュールは、最新バージョンのAxProtectorツールで暗号化し直すことにより、ソースコードを変更せずに、常に最新の暗号化セキュリティ技術でモジュールをプロテクトすることが可能になります。



5-3. WUPI ファンクション一覧

WUPIファンクションの一覧です。各WUPIファンクションの詳細につきましては、後述の「WUPIファンクション詳細」をご参照ください。

WUPI Accessing API

WupiAllocateLicense

指定したライセンスリストのライセンスを割り当てます。

WupiFreeLicense

割り当てたライセンスを解放します。

WupiGetHandle

エントリのネイティブハンドルを返します。

CodeMeterの場合はHCMSysEntryに、WIBU-KEYの場合はHWKBENTRYに指定されます。

WUPI Encryption / Decryption API

WupiEncryptionCode

WupiEncryptionCodeはファンクションを暗号化します。暗号化するファンクションはIxProtector設定で指定する必要があります。

WupiDecryptCode

WupiDecryptCodeはファンクションを復号します。復号するファンクションはIxProtector設定で指定する必要があります。

WUPI Management API

WupiDecreaseUnitCounter

CodeMeterの場合はUnit Counter (ユニットカウンター)、WIBU-KEYの場合はLimit Counter (リミットカウンター)を指定された数値分減らします。

WupiQueryInfo

使用中のライセンス情報を返します。

WupiGetLicenseType

ライセンスタイプを返します。

WupiReadData

CodeMeterの場合はHidden Dataに格納されているデータをリード (Read) します。WIBU-KEYの場合はプロテクト拡張メモリに格納されているデータをリード (Read) します。

WupiReadDataInteger

CodeMeterの場合はHidden Dataに格納されているデータをリード (Read) します。WIBU-KEYの場合はプロテクト拡張メモリに格納されているデータをリード (Read) します。2バイト単位でデータをリード (Read) します。ポインタが使用できる言語の場合、WupiReadDataを利用することを推奨します。

WupiWriteData

CodeMeterの場合はHidden Dataにデータ(raw data)をライト(Write)します。WIBU-KEYの場合はプロテクト拡張メモリにデータ(raw data)をライト(Write)します。

[NOTE]

このファンクションはCmActLicense1には未対応です。

WupiWriteDataInteger

CodeMeterの場合はHidden Dataにデータ(raw data)をライト(Write)します。WIBU-KEYの場合はプロテクト拡張メモリにデータ(raw data)をライト(Write)します。ポインタが使用できる言語の場合、WupiWriteDataを利用することを推奨します。

[NOTE]

このファンクションはCmActLicense1には未対応です。

WUPI Security API

WupiCheckDebugger

プロテクトされたアプリケーションに対して、デバッグ処理が施されたかどうかをチェックします。

WupiCheckLicense

与えられたライセンスリストからライセンスをチェックします。使用されるセキュリティチェックは実行中に変化します。このファンクションは、ライセンスを自動的に割り当てます。

WUPI Error API

WupiGetLastError

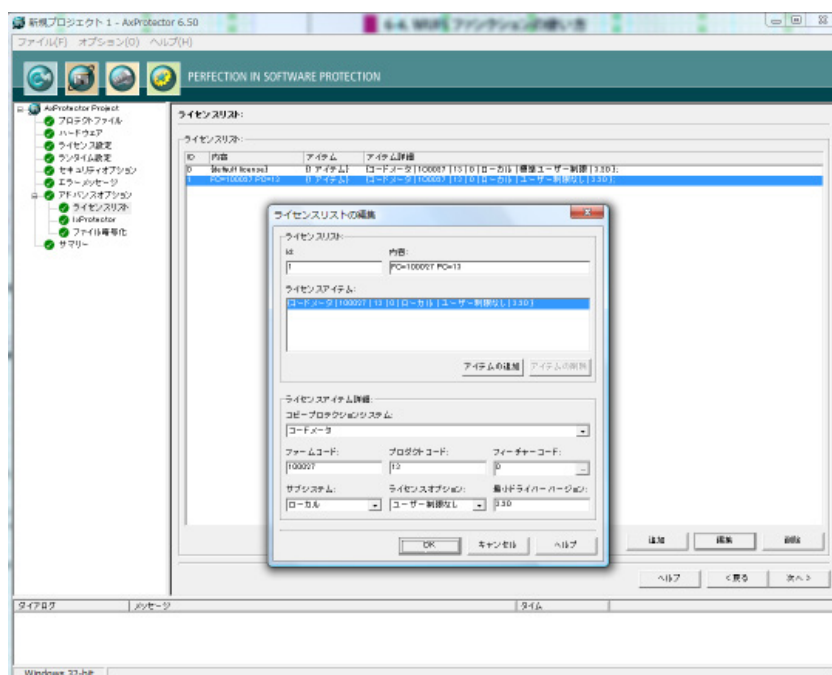
直前に実行したWUPIファンクションのエラーコードを返します。WUPIファンクションの一覧です。各WUPIファンクションの詳細につきましては、後述の「WUPIファンクション詳細」をご参照ください。

5-4. WUPI ファンクションの使い方

WUPIには、インデックスベースWUPI(Index Based WUPI)とポインタベースWUPI(Pointer Based WUPI)の2種類があります。インデックスベースWUPIは、ほとんどすべての言語から呼び出すことが可能で、プログラム開発とライセンスモデルを切り離して管理することが可能です。ポインタベースWUPIは、C/C++のようにポインタを指定しながら開発します。将来性を考え、できる限りインデックスベースWUPIのご使用を推奨致します。以下は、インデックスベースのWUPIの使い方です。

1. ライセンスリスト (License List) を作成する

WUPIファンクションはライセンスリストを参照しながら動作します。このライセンスリストは、自動暗号化ツールAxProtectorの「アドバンスオプション」の「ライセンスリスト」画面で作成します。

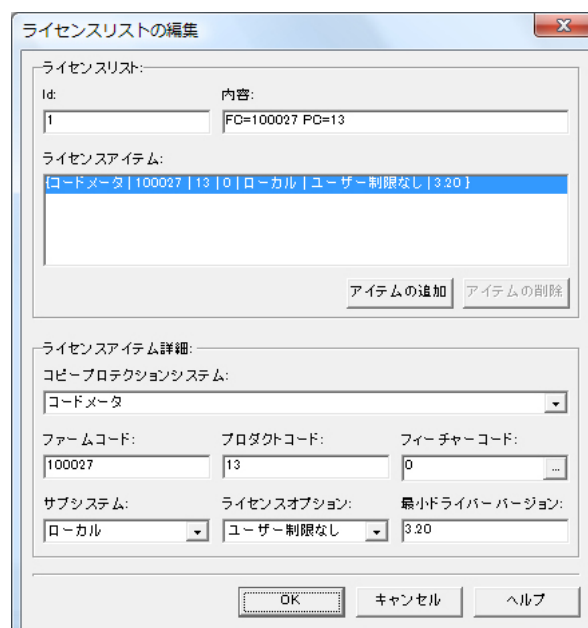


① ライセンスリストを作成する

「ライセンスリストの編集」画面で、インデックスベースのライセンスを作成します。"Id=1"は"Index=1"を意味します。"内容"には、ライセンス名称を明記します。

"ライセンスアイテム"でアイテムを追加する場合は「アイテムの追加」ボタンをクリックしてライセンスアイテムを追加します。

"ライセンスアイテム詳細"には、コピープロテクションシステム(ハードウェアキー)の選択、ファームコード、プロダクトコードなどのセキュリティ内容を設定します。



② ライセンス内容を追加する

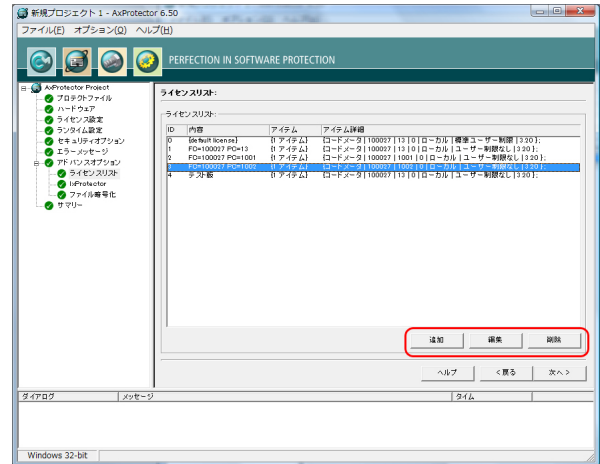
必要に応じて「ライセンスリスト」画面でライセンス内容を追加します。また、すでに登録されたライセンスは「編集」ボタンをクリックして編集したり、「削除」ボタンで削除することができます。

ここで作成されたライセンスリストが、WUPIファンクションから参照されます。

このライセンスリストにアクセスする最初のWUPIファンクションが "WupiCheckLicense()"です。

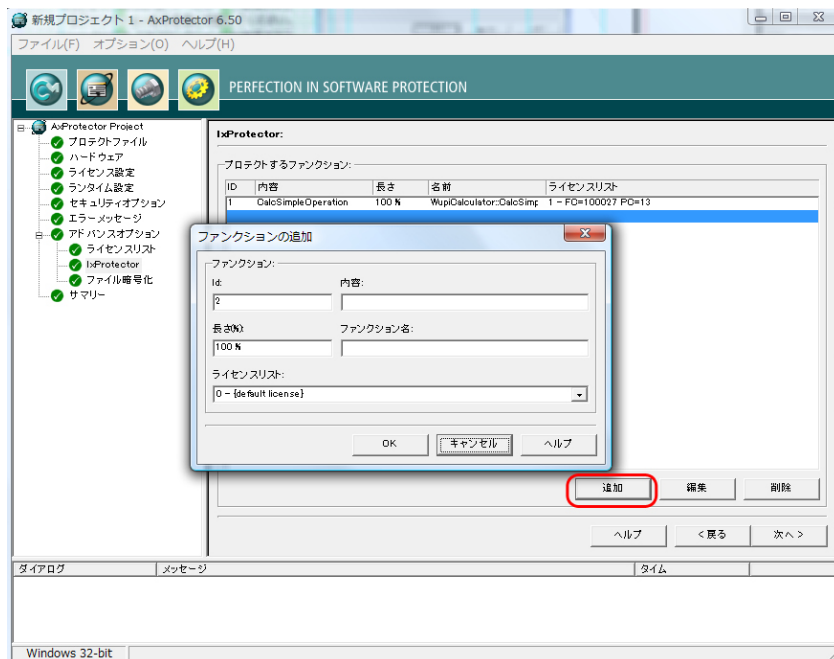
例えば、プログラムから、
WupiCheckLicense(1)

を実行すると、ライセンスリスト上のId=1 (Index=1)にアクセスし、ライセンス内容を取得し、その内容に応じたセキュリティチェックを行います。チェックエラーの場合はFALSE(0)を返し、それ以外はTRUE(1)を返します。



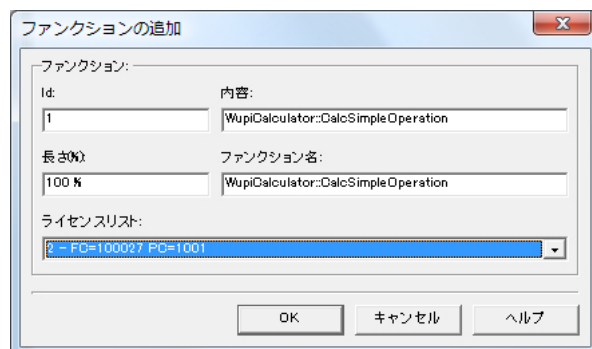
2. ファンクションとライセンスリストを割り当てる

次に、IxProtector画面でファンクションとライセンスリストの割り当てを行います。「追加」ボタンをクリックすると「ファンクションの追加」画面が表示されますので、この画面上でファンクションとライセンスリストの割り当てを行います。「ファンクション名」には、実際にコード内で使用されているファンクション名を正確に入力してください。



① ファンクション定義を行う

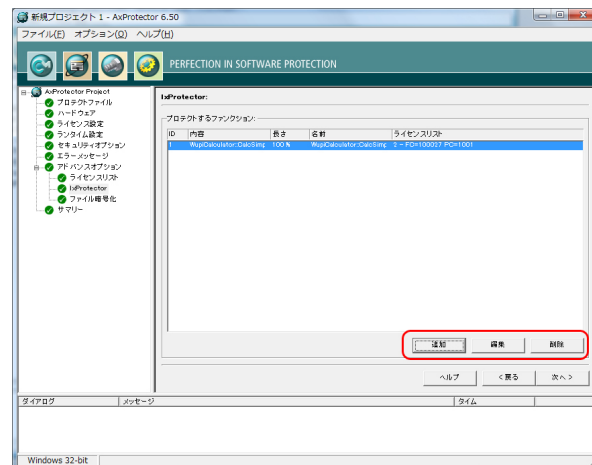
「ファンクションの追加」画面で、暗号化するファンクションを定義します。Idは、ファンクションIdになり、自動的に連番が割り当てられます。「内容」には、ファンクション内容を明記し、「長さ(%)」には暗号化する範囲を%またはバイト数で設定します。%を明記しないで整数値を指定すると、指定されたバイト数を暗号化します。ファンクション名には、ソースコードで使用されているファンクション名を忠実に明記してください。LicenseListでは、このファンクションIdが使用するライセンスを設定します。



② ファンクション定義を追加する

暗号化するファンクションを追加します。すでに定義されたファンクションは「編集」ボタンをクリックして、「ファンクションの編集」画面で編集できます。

ここで定義されたファンクションモジュールは、IxProtectorで暗号化され、メモリー上でも暗号化された状態になります。実際に、ファンクションを実行するためには、そのファンクションが呼び出される直前にWupiDecryptCode()ファンクションを使って復号する必要があります。



例えば、上記①で、CalcSimpleOperationファンクションをファンクションId=1(ライセンスリスト=2/FC=100027,PC=1001)で定義しましたが、実際に使用するには、WupiDecryptCode()ファンクションのパラメータにファンクションID=1を入れて実行します。

WupiDecryptCode(1)

上記1行を実行することで、ライセンスリスト=2に登録されたFC=100027/UC=1001を持つコードメータキーを検索し、見つかった場合、暗号化されているCalcSimpleOperationファンクションモジュールをメモリー上で復号します。(該当するコードメータキーが見つからなかった場合は復号しない。)

WupiDecryptCode(1)の後に、CalcSimpleOperationファンクションを実行します。

実行後は、CalcSimpleOperationファンクションモジュールを暗号化しておくために、暗号化ファンクションWupiEncryptCode()を実行します。復号時と同様、ファンクションId=1をパラメータに入れ、

WupiEncryptCode(1)

を実行します。これで、CalcSimpleOperationファンクションモジュールは暗号化されます。以降、WupiDecryptCodeとWupiEncryptCodeを繰り返すことで「オンデマンド復号」を実現することが可能になります。

WUPIファンクションの詳しい使用方法は、サンプルプログラムが用意されていますので、そちらをご参照ください。

¥Users¥Public¥Documents¥WIBU-SYSTEMS¥Software Protectionの中にあります。
(C#: WupiCalculator/C++; WupiCalculatorIndex)

3. 作成方法

C/C++における作成方法についてご説明します。

- ① ヘッダファイル<wibuixap.h>をプロジェクトにインクルードします。(#include "wibuixap.h")
<wibuixap.h>は、¥Program Files¥WIBU-SYSTEMS¥AxProtector¥devKit¥includeの中にあります。
- ② ソースファイルをコンパイル+リンクする。WupiEngine(32/64).libをリンクします。
WupiEngine(32/64).libは、¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥libの中にあります。
- ③ 上記で生成されたプログラムをWupiEngine(32/64).dllを使って動作検証します。
- ④ 動作検証後、問題なければプログラムをAxProtectorで暗号化します。
暗号化後は、WupiEngine(32/64).dllは不要です。(必要なモジュールが静的にリンクされます。)
- ⑤ これで完成です。

[NOTE]

WUPIファンクションの具体的な組み込み方法は、サンプルプログラムが用意されていますので、そちらをご参照ください。

[NOTE]

IxProtectorの対象になるファイルは、以下の3つの条件を満たす必要があります。

- Windows EXE実行形式プログラムまたはDLLプログラム
- 自動暗号化ツールAxProtectorで暗号化し、問題なく動作すること。
(IxProtectorオプションは指定しないで暗号化すること)
- 一般的なDLLファイルを呼び出せるプログラムであること。

5-5. WUPI ファンクション詳細

WupiAllocateLicense

[内容]

指定したライセンスリストのライセンスを割り当てます。

ライセンスの割り当てには、WupiCheckLicenseファンクションを使用する方が一般的です。
WibuCheckLicenseはライセンスを自動的に割り当てます。

[Syntax]

```
int WupiAllocateLicense (int iLicenseList);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

[リターン値]

エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiFreeLicense

[内容]

割り当てたライセンスを解放します。

[Syntax]

```
int WupiFreeLicense (int iLicenseList);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

[リターン値]

エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiGetHandle

[内容]

エントリのネイティブハンドルを返します。

CodeMeterの場合はHCMSysEntryに、WIBU-KEYの場合はHWKBENTRYに指定されます。

[Syntax]

```
WupiGetHandle (int iLicenseList);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

[リターン値]

エラーの場合は0を返します。

WupiEncryptionCode**[内容]**

WupiEncryptionCodeはファンクションを暗号化します。暗号化するファンクションはIxProtector設定で指定する必要があります。

[Syntax]

```
int WupiEncryptCode (int iFunction);
```

[パラメータ]**iFunction**

ファンクションインデックスを参照します。

[リターン値]

エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiDecryptCode**[内容]**

WupiDecryptCodeはファンクションを復号します。復号するファンクションはIxProtector設定で指定する必要があります。

[Syntax]

```
int WupiDecryptCode (int iFunction);
```

[パラメータ]**iFunction**

ファンクションインデックスを参照します。

[リターン値]

エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiDecreaseUnitCounter**[内容]**

CodeMeterの場合はUnit Counter (ユニットカウンタ)、WIBU-KEYの場合はLimit Counter (リミットカウンタ)を指定された数値分減らします。

[Syntax]

```
int WupiDecreaseUnitCounter (int iLicenseList, int cUnits);
```

[パラメータ]**iLicenseList**

ライセンスリストインデックスを参照します。

cUnits

削減する数値。

[リターン値]

エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiQueryInfo

[内容]

使用中のライセンス情報を返します。

[Syntax]

```
int WupiQueryInfo (int iLicenseList, int iCmd);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

iCmd (フラグ)

WupiQIUnitCounter

WIBU-KEYの場合はLimit Counter (リミットカウンタ) の値、CodeMeterの場合はUnit Counter (ユニットカウンタ) の値を返します。

WupiQIExpTime

Expiration Date (有効期限) を返します。

Expiration Dateは、2000年1月1日からの秒数になります。

WupiQIActTime

Activation Time (使用開始期日) を返します。(CodeMeterのみ)

Activation Timeは、2000年1月1日からの秒数になります。

WupiQIUsagePeriod

使用期間の最終日時からの有効な秒数を返します。使用期間が過ぎている場合は0を返します。(CodeMeterのみ)

WupiQIFirmCode

使用中のライセンスのファームコード(Firm Code)を返します。

WupiQIProductCode

使用中のライセンスのプロダクトコード(Product Code)を返します。WIBU-KEYの場合は、ユーザーコード (User Code)を返します。

WupiQIFeatureMap

使用中のライセンスのフィーチャーマップ (Feature Map)を返します。(CodeMeterのみ)

WupiQIBoxMask

使用中のライセンスを持つデバイスキーのボックスマスク(Box Mask)を返します。

WupiQIBoxSerial

使用中のライセンスを持つデバイスキーのシリアル番号(Serial Number)を返します。

[リターン値]

エラーの場合、または取得する情報が無い場合は、-1を返します。

ただし、WupiQIFeatureMapフラグを使って取得した値が-1の場合は、エラーではありません。

FeatureMap=0xFFFFFFFF(=-1)になります。WupiQIFeatureMapのエラーコードは0になります。

WupiGetLastErrorファンクションを使って正確なエラーコードを取得することができます。

WupiGetLicenseType

[内容]

ライセンスタイプを返します。

[Syntax]

```
int WupiGetLicenseType (int iLicenseList);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

[リターン値]

UpiLicenseNone

ライセンスタイプは内部で設定されている。

UpiLicenseWibuKey

ライセンスタイプはWIBU-KEY。

UpiLicenseCodeMeter

ライセンスタイプはCodeMeter。

UpiLicenseCodeMeterAct

ライセンスタイプはCodeMeterAct。

WupiCheckDebugger

[内容]

プロテクトされたアプリケーションに対して、デバッグ処理が施されたかどうかをチェックします。

[Syntax]

```
int WupiCheckDebugger (int iLicenseList, int nLevel);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

nLevel

アンチデバッグレベルを指定します。

Level 0: デバッガーチェックをしない。

Level 1: 簡易デバッガーチェック。

Level 2: カーネルデバッガーチェック。

Level 4: アドバンスデバッガーチェック。

Level 8: IDEデバッガーチェック。デバッガーが完全に使用できなくなる。

Level 16: デバッガーを検知するとCodeMeterまたはWIBU-KEYのエントリーを無効化(アクセス不可)する。

Level 64: 仮想マシン上での動作をチェックする。プロテクトされたアプリケーションが仮想マシン上で動作することを禁止する。

[リターン値]

デバッガーが検知されるとTRUE(1)を返し、それ以外はFALSE(0)を返します。Level16が含まれると、何も返さずにエントリーを無効化しアプリケーションを速やかに終了させます。デバッガーが検知された場合、WupiGetLastErrorファンクションは、error wibu::UpiErrorDebuggerDetected(-5)を返します。

WupiCheckLicense

[内容]

与えられたライセンスリストからライセンスをチェックします。使用されるセキュリティチェックは実行中に変化します。このファンクションは、ライセンスを自動的に割り当てます。

[Syntax]

```
int WupiCheckLicense (int iLicenseList);
```

[パラメータ]

iLicenseList

ライセンスリストインデックスを参照します。

[リターン値]

エラーの場合FALSE(0)を返し、それ以外はTRUE(1)を返します。

WupiGetLastError

[内容]

直前に実行したWUPIファンクションのエラーコードを返します。

[Syntax]

```
int WupiGetLastError();
```

[リターン値]

エラーコードを返します。

Error Codes

Errorcode: UpiErrorDebuggerDetected

デバッガーによる解析行為が検知されました。(-5)

Errorcode: UpiErrorFunctionNotFound

指定されたファンクションポイントが見つかりませんでした。エラー(-3)

Errorcode: UpiErrorInfoNotAvailable

WupiQueryInfoファンクションで取得した情報は無効です。(-7)

Errorcode: UpiErrorLicenseModuleNotLoaded

ライセンスモジュールがロードできません。(-6)

このエラーは、WIBU-KEYの場合WkWin32/64.dll、CodeMeter/CodeMeterActの場合WibuCm32/64.dllが見つからないか、またはインストールされていない場合に発生します。

Errorcode: UpiErrorLicenseNotFound

指定されたライセンスが見つかりませんでした。エラー(-2)

Errorcode: UpiErrorNoDefaultLicense

デフォルトライセンスが見つかりません。アプリケーションはAxProtectorでプロテクトされていません。エラー(-1)

Errorcode: UpiErrorNoError

エラーは発生しませんでした。(0)

Errorcode: UpiErrorRuntimeTooOld

インストールされているランタイムは古いバージョンです。エラー(-4)

Chapter 6

CodeMeter Core API について

- 6-1. CodeMeter Core API
- 6-2. サンプルプログラムについて
- 6-3. CodeMeter API ガイドの使い方
- 6-4. CodeMeter API ガイドの使用例
- 6-5. CodeMeter Core API 一覧

6-1. CodeMeter Core API

コードメータには、多数のAPIファンクションが用意されています。これらのAPIファンクションを、貴社のソースコードに直接組み込むことにより、きめの細かいセキュリティを実現することができます。APIファンクションを組み込んだ後は、自動暗号化ツールAxProtectorを使って、ファイル全体を暗号化することが可能です。逆コンパイラやデバッガによる解析からファイルを守るためにも、自動暗号化ツールAxProtectorとの併用をお勧めいたします。また、CodeMeter Core APIファンクションは、Windows/Linux/Macに共通に使用できるクロスプラットフォームAPIです。

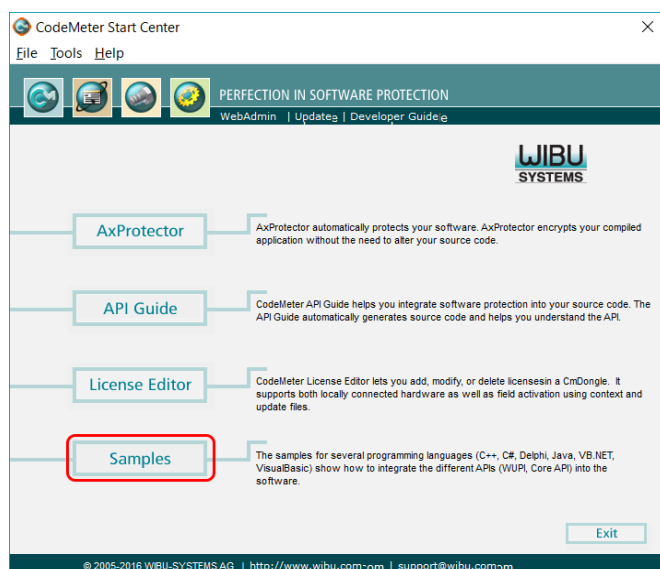
6-2. サンプルプログラムについて

コードメータ開発キットをインストールすると、インストール先の下記フォルダに開発言語ごとのサンプルプログラムがインストールされます。APIファンクションの使い方を参考にしてください。

[すべてのプログラム]/[CodeMeter]/[Samples]/[Software Protection]

¥Users¥Public¥Documents¥WIBU-SYSTEMS¥Software Protection

デスクトップ上のCodeMeter Start Centerのアイコンをクリックし、CodeMeter Start Center/Sampleをクリックしてもアクセスできます。



サンプルプログラムがある言語は以下のとおりです。それ以外のサンプルプログラムについては、弊社までお問い合わせください。

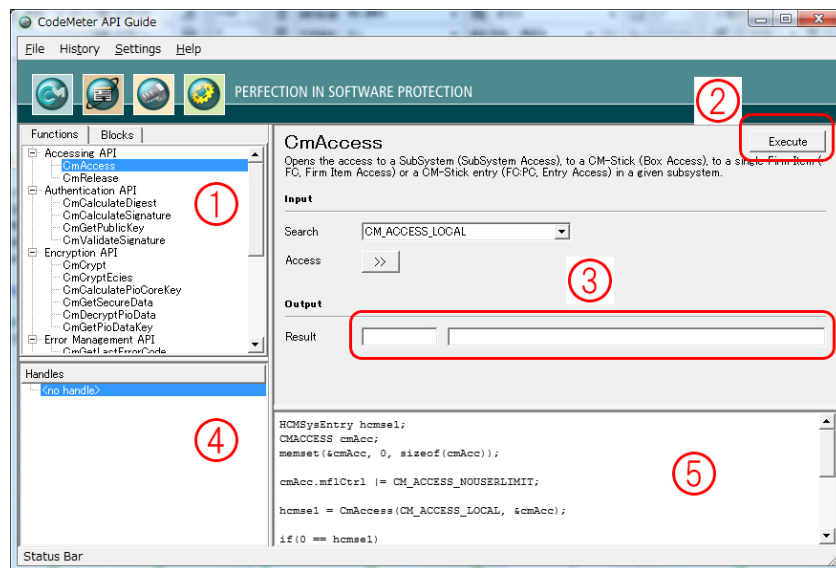
C#, C++, Delphi, Java, VBasic.NET

6-3. CodeMeter API ガイドの使い方

コードメータAPIファンクションを利用する場合の支援ツールとして CodeMeter API ガイドが用意されています。CodeMeter API ガイドを使ってAPIファンクションを実際に行わせながら、APIファンクション呼び出しのサンプルソースコードを生成することができます。なお、CodeMeter API ガイドを利用する場合は、APIファンクション呼び出しで使用するCmContainer(CmDongleまたはCmActLicense)をPCに装着またはインポートする必要があります。以下に、CodeMeter API ガイドの使い方について説明します。

CodeMeter API ガイドの起動方法

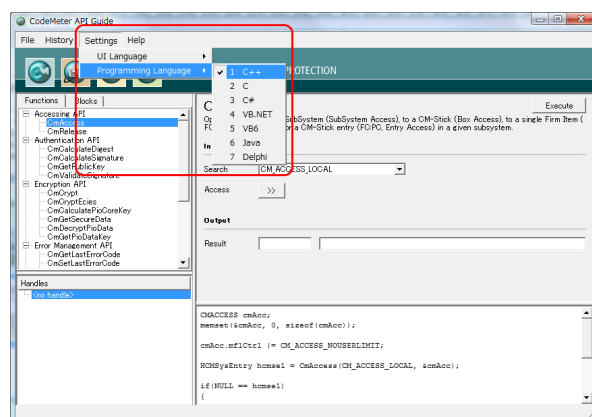
【スタート】→【プログラム(P)】→【CodeMeter】→【Tools】→【CodeMeterApiGuide】をクリックして起動します。



- ① APIファンクションの一覧が表示されます。APIファンクションを選択します。
- ② APIファンクションを実行します。「Execute」ボタンをクリックするとファンクションが実行されます。ファンクションによっては、パラメータ入力が必要になる場合があります。
- ③ APIファンクションの実行結果が表示されます。
- ④ 実行したAPIファンクションのハンドルが表示されます。
- ⑤ 実行したAPIファンクションのソースコードが表示されます。作成されたソースコードは、[File]/[Export generated code]でファイル保存できます。

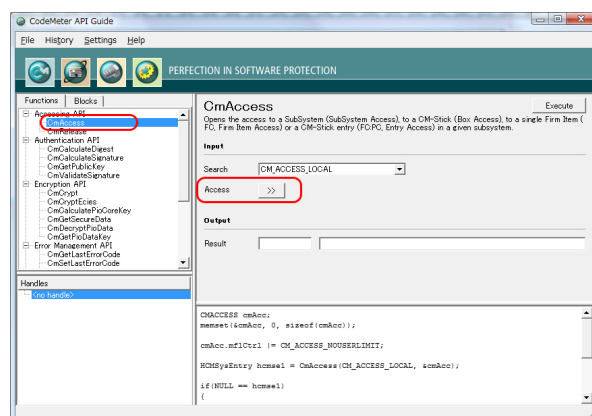
6-4. CodeMeter API ガイドの使用例

実際に ファームコード=10、プロダクトコード=13 のCmStickがローカルPCに装着されているかどうかをチェックする例を示します。CodeMeter API ガイドを起動して、メニューの【Settings】→【Programming Language】を開いて使用する開発言語を選択します。ここで指定した開発言語でサンプルソースが作成されます。

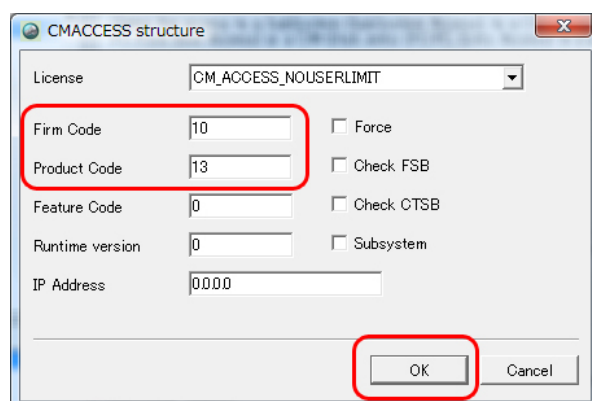


特定のファームコードとプロダクトコードが登録されたCmContainerをチェックするだけであれば、CmAccess() → CmGetLastErrorCode() → CmRelease() を行うだけで実現できます。CodeMeter API ガイドで実際に実行してみましょう。

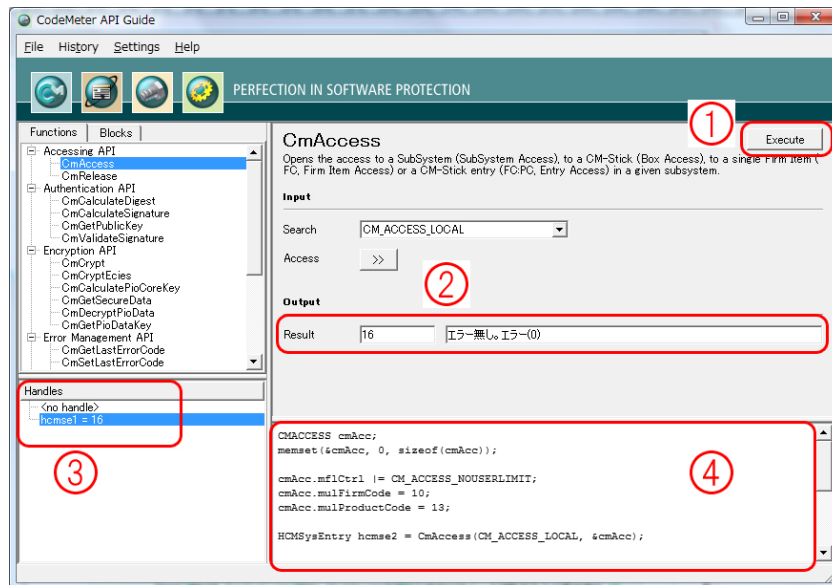
① FirmCode=10、ProductCode=13 が登録された CmStickをPCに装着し、CmAccess を選択します。Access項目の [>>] をクリックし、CMACCESS ストラクチャ画面を開きます。



② CMACCESSストラクチャ画面で、Firm Codeに10、Product Codeに13を入力し、OKボタンをクリックします。

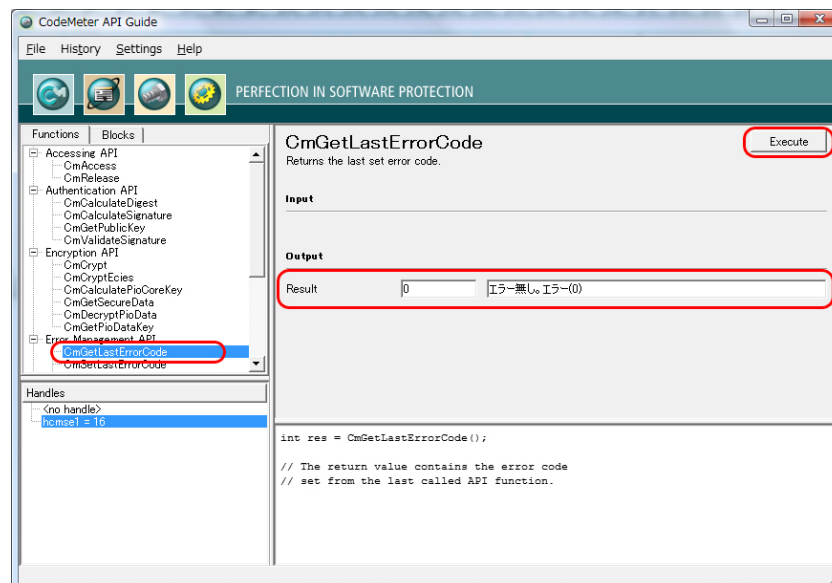


③ CodeMeter APIガイド画面に戻り、「Execute」ボタンをクリックすると、CMACCESSストラクチャで設定した内容でCmAccessファンクションが実行され、その結果がResult欄に表示されます。(②) CmAccessファンクションが返したハンドルは、「Handles」欄に表示されます。(③) 実行されたソースコードは(④)に表示されます。

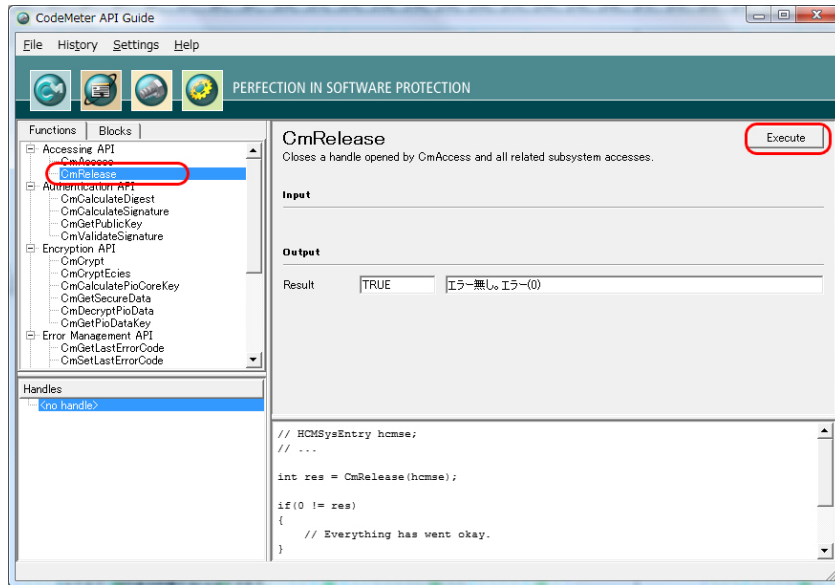


実行に成功すると、リターン値に0以外の整数値が返り、実行エラーの場合は、リターン値に0が返ります。(②) リターン値が0であるとエラーが発生したことはわかりますが、どのようなエラーが発生したかまではわかりません。どのようなエラーが発生したかを知るために、CmAccessファンクションに続けてCmGetLastErrorCodeファンクションを実行します。このCmGetLastErrorCodeファンクションは、直前に実行したファンクションのエラーコードを取得するファンクションです。

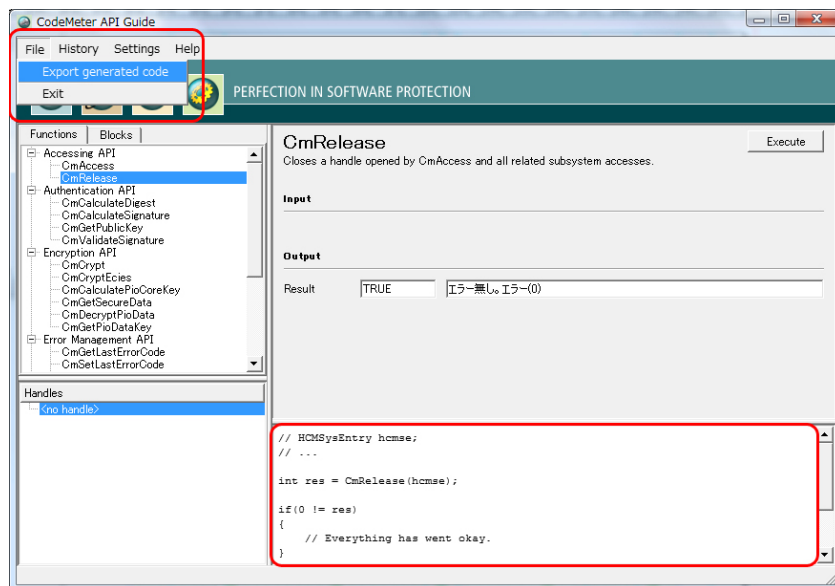
④ Functions一覧から、CmGetLastErrorCodeを選択し、「Execute」ボタンをクリックします。エラーが無い場合、リターン値に0が返ります。エラーの場合は、そのエラーコードが返ります。エラーコードを調べることでエラーの内容を把握することができます。



⑤ 最後に、CmReleaseファンクションを実行して、CmAccessファンクションで使用したハンドルを閉じます。



⑥ CodeMeter API ガイドで行ったCmAccess、CmGetLastErrorCode、CmReleaseの操作のソースコードは、メニューの【File】→【Export Generated Code】でファイル保存することができます。



6-5. CodeMeter Core API 一覧

Accessing API

CmAccess

サブシステム、CmStick、ファームアイテム、CmStickエントリへアクセスします。

CmAccess2

CmAccessおよび拡張機能の実行 (CodeMeter Version 3.30よりサポート)
拡張機能を利用する場合は、CmAccess2を利用してください。

CmRelease

CmAccessおよびCmAccess2でオープンされたハンドルをクローズする。

[注意]

WUPIファンクションWupiGetHandleで割り当てたエントリはCmReleaseでクローズしないでください。

Authentication API

CmCalculateDigest

認証で使用する32バイトのハッシュ値を計算する。SHA-256アルゴリズムが使用されます。

CmCalculateSignature

CmStick内の指定されたハッシュ値でECDSA (Elliptic Curve Digital Signature Algorithm)シグニチャーを計算する。

CmGetPublicKey

CmStickからパブリックキーを読む。

CmValidateSignature

指定されたパブリックキーでECDSA (Elliptic Curve Digital Signature Algorithm)シグニチャーを検証する。

Encryption API

CmCrypt

CmStickを使って暗号化および復号化を行う。直接(ダイレクト)または間接(インダイレクト)による暗号化・復号化が可能。

CmCrypt2

CmStickを使って暗号化および復号化を行う。直接(ダイレクト)または間接(インダイレクト)による暗号化・復号化が可能。CodeMeter Ver4.30よりサポート。

[NOTE]

CmCryptに以下の2つのコマンドが追加されました。

CM_CRYPT_AES_ENC_CBC_DIRECT

CM_CRYPT_AES_DEC_CBC_DIRECT

CmCryptEcies

ECIES (Elliptic Curve Integrated Encryption Scheme) アルゴリズムで指定したバイトを暗号化する。

CmCryptSim

指定したファームコードを持つFSB(Firm Security Box) エントリを使って、直接および間接の暗号化・復号化を行う。

CmCryptSim2

指定したファームコードを持つFSB(Firm Security Box) エントリを使って、直接および間接の暗号化・復号化を行う。CodeMeter Ver4.30よりサポート。

[NOTE]

CmCryptSimに以下の2つのコマンドが追加されました。

CM_CRYPT_AES_ENC_CBC_DIRECT

CM_CRYPT_AES_DEC_CBC_DIRECT

CmCalculatePioCoreKey

PIO Hidden Dataを暗号化するためのコアキーを計算する。この作業を行うにはFSB(Firm Security Box)が必要です。

CmGetSecureData

Production Item Option Encryption Key (PIOEK)を使って、CmStickから暗号化されたHidden Dataを読む。

CmDecryptPioData

Production Item Option Data Key (PIODK)を使って、暗号されたHidden Dataを復号化する。

CmGetPioDataKey

HIDDEN DATAを復号化するためのキーを計算する。

Error Management API

CmGetLastErrorCode

最後のエラーコードを取得する。

CmSetLastErrorCode

エラーコードを内部グローバルエラーコードに設定する。

CmGetLastErrorText

最後のエラーテキストを取得する。

Managing API

CmBorrow

ライセンスの貸出し、検証、返却を行う。

CmCheckEvents

選択したイベントのどれか 1 つが発生するまでウェイト (待つ) し、その結果を返す。

CmGetBoxes

接続されている全てのCmStickを調べる。

CmGetBoxContents

CmStickの全てのエントリを読む。

CmGetBoxContents2

CmStickの全てのエントリを読む。さらに、拡張機能をサポートする。

CmGetInfo

オープンされたエントリに関連する情報が返る。エントリのデータ自身は返らない。

CmGetServers

ローカルネットワークを検索し、CmStickが接続されているコードメータライセンスサーバーを探す。

CmGetVersion

コードメータモジュールのバージョンを返す。

Remote Programming API

CmGetRemoteContext

CmStickの内容をリモートコンテキストファイル (ライセンス要求ファイル) (WibuCmRaC) に保存する。リモートコンテキストファイルは暗号化され圧縮されます。

CmGetRemoteContext2

CmGetRemoteContextと同じくリモートコンテキストファイルを保存します。さらに、拡張機能をサポートします。

CmSetRemoteUpdate

指定したリモートアクティベーションアップデートファイル (ライセンス更新ファイル; WibuCmRaU) でCmStickをプログラムします。

CmSetRemoteUpdate2

指定したリモートアクティベーションアップデートファイル (ライセンス更新ファイル; WibuCmRaU) でCmStickをプログラムします。さらに、拡張機能をサポートします。

CmListRemoteUpdate

リモートアクティベーションアップデートファイル (ライセンス更新ファイル; WibuCmRaU) を解析し、ファイルに登録されている全てのCmStickのシリアル番号を調べる。

CmListRemoteUpdate2

リモートアクティベーションアップデートファイル(ライセンス更新ファイル; WibuCmRaU)を解析し、ファイルに登録されている全てのCmStickのシリアル番号を調べる。さらに、拡張機能をサポートします。

CmExecuteRemoteUpdate

WibuCmRaUファイルを解析し、必要な更新を実行します。

Time Update API

CmSetCertifiedTimeUpdate

タイムサーバー(認証時間サーバー、CTCS)から現在の認証時間とデータスタンプを取得し、CmStickに保存する。

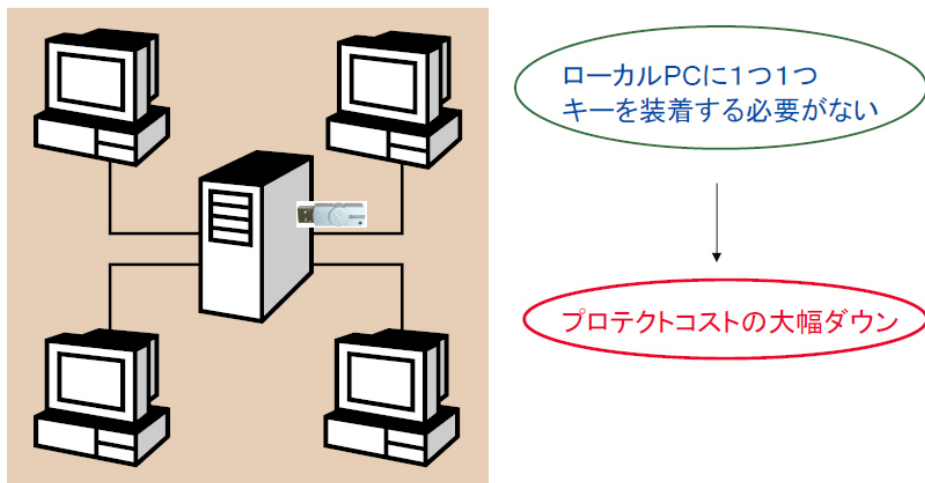
Chapter 7

ネットワーク機能について

- 7-1. ネットワークライセンス管理とは
- 7-2. ネットワークカウンタの登録方法
- 7-3. コードメータサーバーの起動方法
- 7-4. ネットワーク対応型プロテクトの作成方法

7-1. ネットワークライセンス管理とは

コードメータのネットワークライセンス管理とは、ネットワーク上のコードメータサーバーにコードメータキー(CmDongle)を1つ装着し、アプリケーションのクライアントライセンス数(フローティングライセンス数)を制御することです。CmDongleのネットワークカウンタ(Network Counter)に数値を登録することで、ユーザーに提供するライセンス数を設定することができます。ローカルPCに1つ1つ装着する必要がないため、プロテクトコストを大幅に節約することができます。また、ライセンスモードも5通りの方法があり、ニーズに応じて使い分けることができます。



コードメータキー(CmDongle)を装着するサーバーは、必ずしも実際のネットワークを管理するサーバーである必要がなく、クライアントPCの1台をコードメータサーバーにすることが可能です。実際には、コードメータのWebAdmin上でネットワークサーバーの実行をONにするだけでコードメータサーバーに切り替わります。

1つのCmDongleに登録できるネットワークカウンタ(Network Counter)値は最大値65,536までです。理論的には、1つのCmDongleで最大クライアント数65,536台までライセンス制御が可能です。運用的には、サーバー負荷などを考慮の上、ライセンス数を決定してください。なお、CmDongleを複数使用することで、ライセンス数を加算させながらサーバー負荷を減らすことも可能です。また、万が一のためのバックアップサーバーに、同じ内容のCmDongleを置くことも可能です。(Hot Standby/Cold Standby)

7-2. ネットワークカウンタの登録方法

コードメータキー (CmDongle) にネットワークカウンタを登録するには、コードメータライセンスエディタまたはCmBoxPgm.exeを使ったコマンドラインから行います。また、登録作業を行うには、必ず貴社のコードメータFSB(CmFSB)が必要になります。

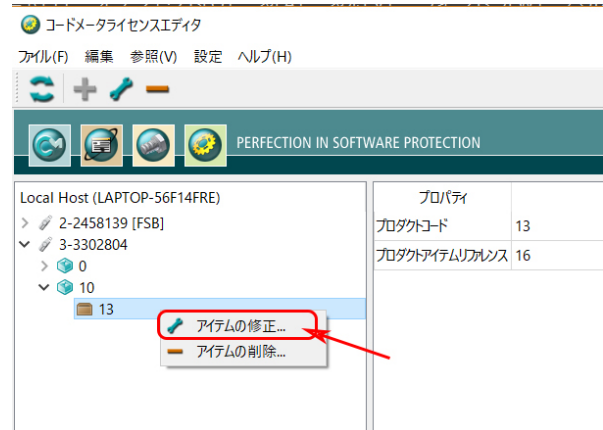
コードメータライセンスエディタを使用する場合

① CmDongle と CmFSB を PC に装着する

ネットワークカウンタを登録するCmDongleと、貴社のコードメータFSB(CmFSB)をPCに装着します。作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

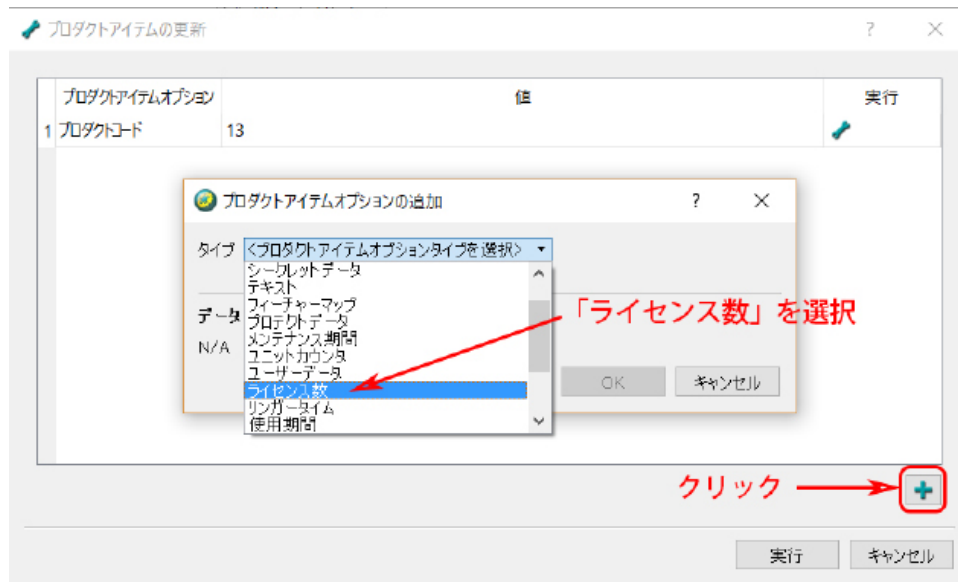
② コードメータライセンスエディタを起動

【スタート】→【すべてのプログラム】→【CodeMeter】→【Tools】→【CodeMeter License Editor】をクリックし、コードメータライセンスエディタを起動します。起動後、プロダクトコード上で右クリックをし、「アイテムの修正...」メニューを選択します。

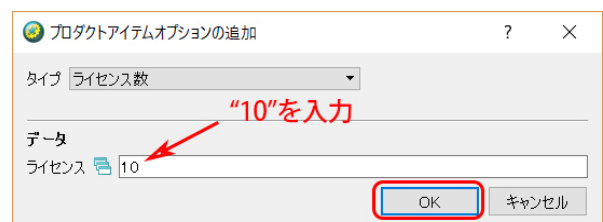


③ 「ライセンス数」を設定

「プロダクトアイテムの更新」画面の「+」ボタンをクリックし、「プロダクトアイテムオプションの追加」画面で「ライセンス数」を選択します。

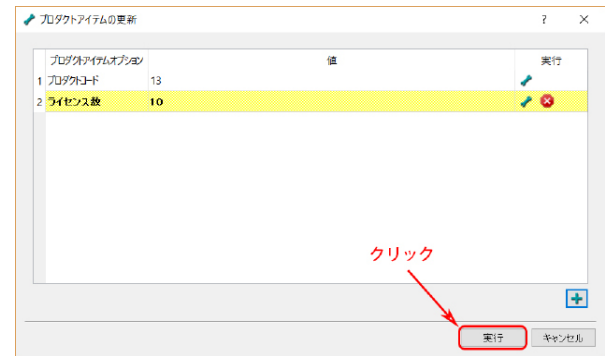


「ライセンス」項目にネットワークライセンス数を入力します。ここでは、「10」を入力し、OKボタンをクリックします。



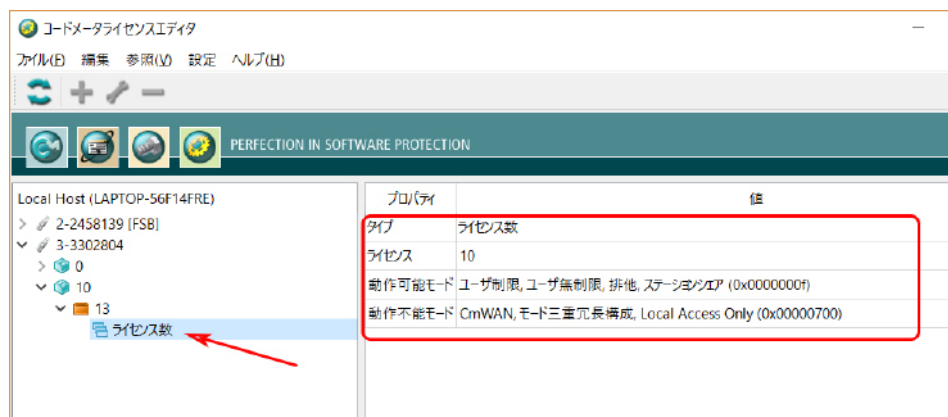
④ 「実行」 ボタンをクリックする

「プロダクトアイテムの更新」画面で「実行」ボタンをクリックすると、CmDongleにネットワークライセンス数が登録されます。



⑤ 登録内容を確認する

「プロダクトアイテムの更新」画面で「実行」ボタンをクリックすると、CmDongleにネットワークライセンス数が登録されます。



Webアドミンからも確認できます。



CmBoxPgm.exe を使用する場合（コマンドライン環境）

① CmDongle と CmFSB を PC に装着する

ネットワークカウンタを登録するCmDongleと、貴社のコードメータFSB(CmFSB)をPCに装着します。作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

② コマンドプロンプトを開きます。

【スタート】→【すべてのプログラム】→【アクセサリ】で【コマンドプロンプト】を起動します。以後、コマンドライン上での操作になります。

③ CD(Change Directory) コマンドで以下のフォルダをカレントにする。

¥Program Files(x86)¥CodeMeter¥DevKit¥bin

[例] コマンドプロンプトを開いて、
>CD ¥Program Files(x86)¥CodeMeter¥DevKit¥bin ↓

④ ネットワークカウンタを登録する

ファームコード=10、プロダクトコード= 13、ネットワークカウンタ= 50 を新規登録します。ネットワークカウンタを登録するパラメータは"/PNWC"です。コマンドラインから、下記のようにタイプしEnterキーを押します。

CmBoxPgm /F10 /P13 /PNWC50 /CA ↓ （ ↓はEnterキー）



[NOTE]

"/CA"オプションを実行すると、指定されたプロダクトコードを持つプロダクトアイテムが新規追加されます。すでに同一のプロダクトコードを持つプロダクトアイテムが存在している場合でも、新規で追加作成されます。あらかじめプロダクトアイテムを削除しておくか、変更オプション"CU"のご使用をお勧めします。既存のプロダクトアイテムのネットワークカウンタを変更する場合は、

CmBoxPgm /F10 /P13 /PNWC100 /CU ↓ （ ↓はEnterキー）

既存のファームコード=10、プロダクトコード=13のプロダクトアイテムに対して、ネットワークカウンタが"100"に変更されます。(変更の場合は、"/CA"でなく"/CU"を使用します)

CmBoxPgm.exeおよびパラメータの使い方は、「Chapter 10 CmBoxPgmの使い方」をご参照ください。

7-3. コードメータサーバーの起動方法

コードメータサーバーは、Webアドミンのネットワーク設定から行います。コードメータランタイムキットをインストールするとこの機能も自動的にインストールされます。コードメータサーバーは、必ずしもネットワークを実際に管理するサーバーを指定する必要がなく、クライアントPCの1台をコードメータサーバーにすることが可能です。どれをサーバーにするかは、アクセス負荷を考慮の上、指定してください。

① コードメータサーバーに CmStick を装着します。

コードメータサーバーには、すでにコードメータランタイムキットがインストールされている必要があります。

② コードメータ Web アドミンを起動する

タスクバーにあるコードメータアイコンをクリックし、コードメータコントロールセンターを開き、右下の「Webアドミン」ボタンをクリックし、CodeMeter Webアドミンを起動します。

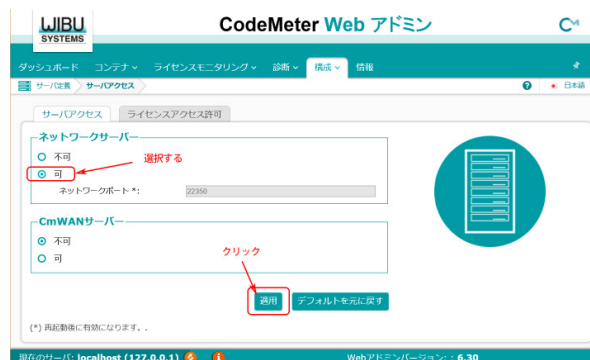


「構成」メニューをクリックし、「サーバー」/「サーバアクセス」メニューをクリックします。

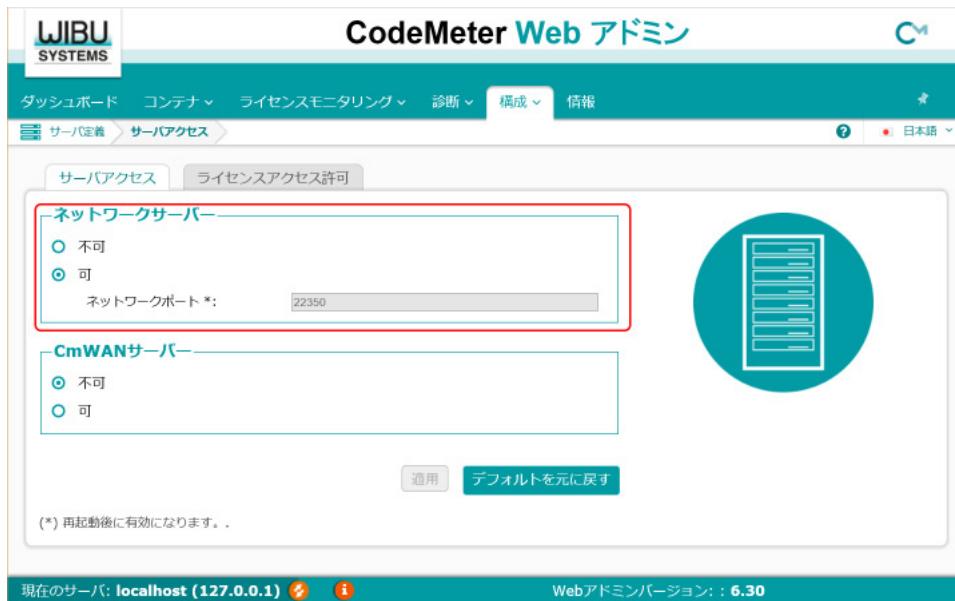


③ ネットワークサーバーを設定する。

「サーバアクセス」画面で、「ネットワークサーバー」の「可」にチェックを入れ、画面下の「適用」ボタンをクリックします。



「適用」ボタンをクリックすると、ネットワークサーバーが設定されます。



④ ライセンス数を確認する。

Webアドミンの「ライセンスモニタリング」をクリックすると許可されたライセンス数を確認することができます。また、「詳細」をクリックすると、ライセンスの使用状況を確認することができます。



さらに、<ボタンをクリックすると、詳細が展開されます。



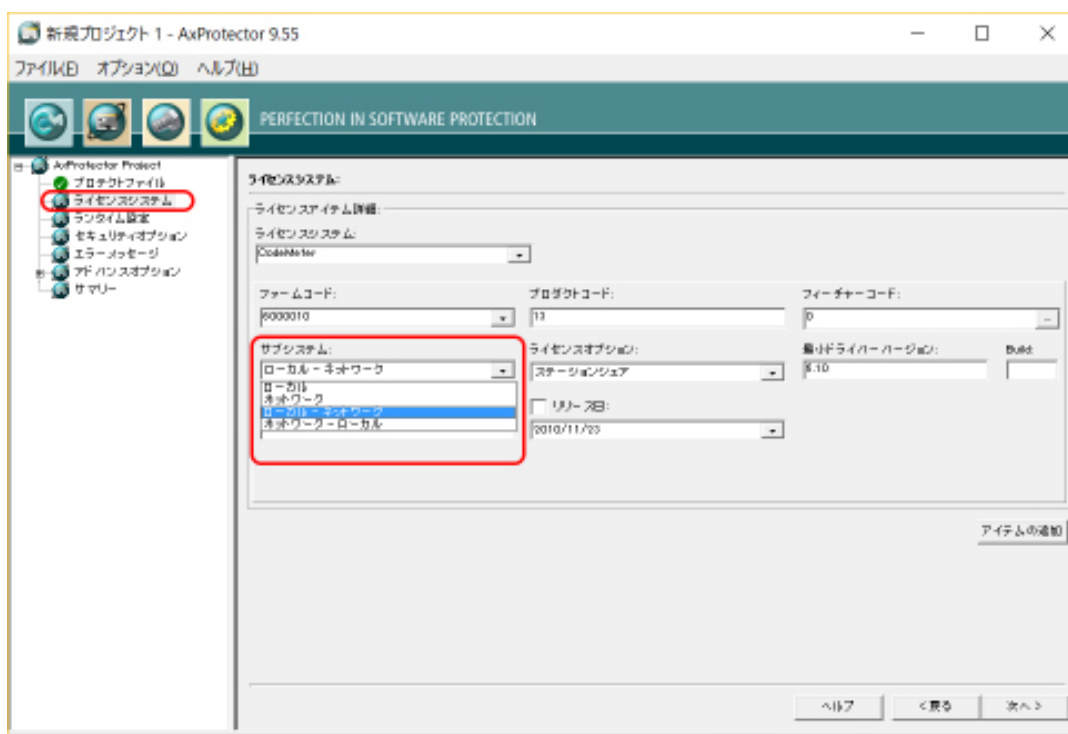
7-4. ネットワーク対応型プロテクトの作成方法

自動暗号化ツール「AxProtector」を使って、ネットワーク対応型のプログラムを作成するには、下記の2点が必要になります。

- ① AxProtector「ライセンスシステム」の「サブシステム」で「ネットワーク」、「ローカルネットワーク」、「ネットワークローカル」のいずれかを選択する。（「ローカル」は選択しない）
- ② AxProtector「ランタイム設定」で「ランタイムチェックを有効」にチェックを入れる。

①「サブシステム」を指定する

自動暗号化ツール「AxProtector」の「ライセンスシステム」の「サブシステム」で「ローカル」以外を選択します。



「ネットワーク」

ネットワークのみアクセスする

「ローカルネットワーク」

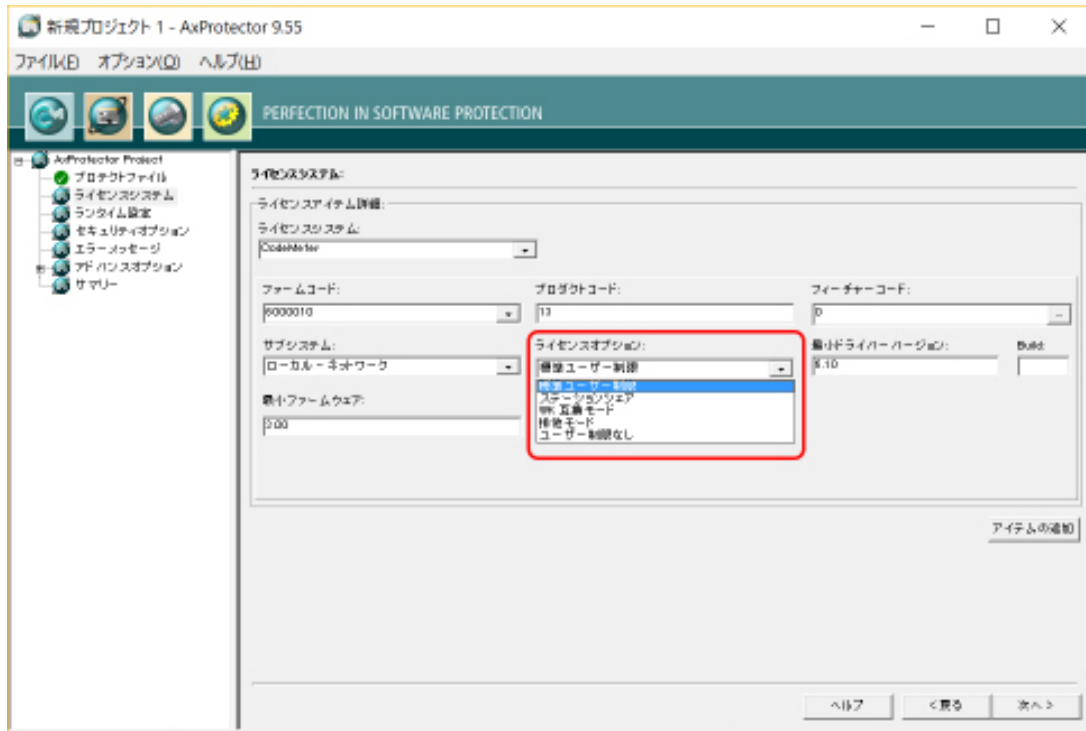
最初にローカルアクセスを行い、次にネットワークアクセスを行う

「ネットワークローカル」

最初にネットワークアクセスを行い、次にローカルアクセスを行う

ライセンスオプション:

ライセンスオプションはライセンスニーズに応じて設定します。



○ 標準ユーザー制限(アプリケーション単位で制限)

実行するアプリケーションごとに1つのライセンスを割り当てます。例えば、同じアプリケーションを同時に2回起動する場合は2つのライセンスが必要になります。この原則はコードメータキーがローカルにある場合もネットワーク上にある場合も同じように適用されます。

(1アプリケーション=1ライセンスの考え方)

○ ステーションシェア(クライアント単位で制限)

1台のPCで同一のアプリケーションを複数回起動した場合でも1ライセンスとして扱われます。

(1PC=1ライセンスの考え方)

○ WK互換モード(ネットワークでは標準的なユーザー制限、ローカルは制限なし)

ネットワーク上のコードメータキーに対しては「標準ユーザー制限」として動作しますが、ローカルマシン上のコードメータキーに対しては制限がありません。

* CmDongleをローカルPCに装着すると、そのPC上では同一のアプリケーションが無限に起動できるが、そのCmDongleをコードメータサーバーに装着すると、「標準ユーザー制限(1アプリケーション=1ライセンス)」として機能する。

○ 排他モード(クライアントごとに1つのアプリケーションのみ許可)

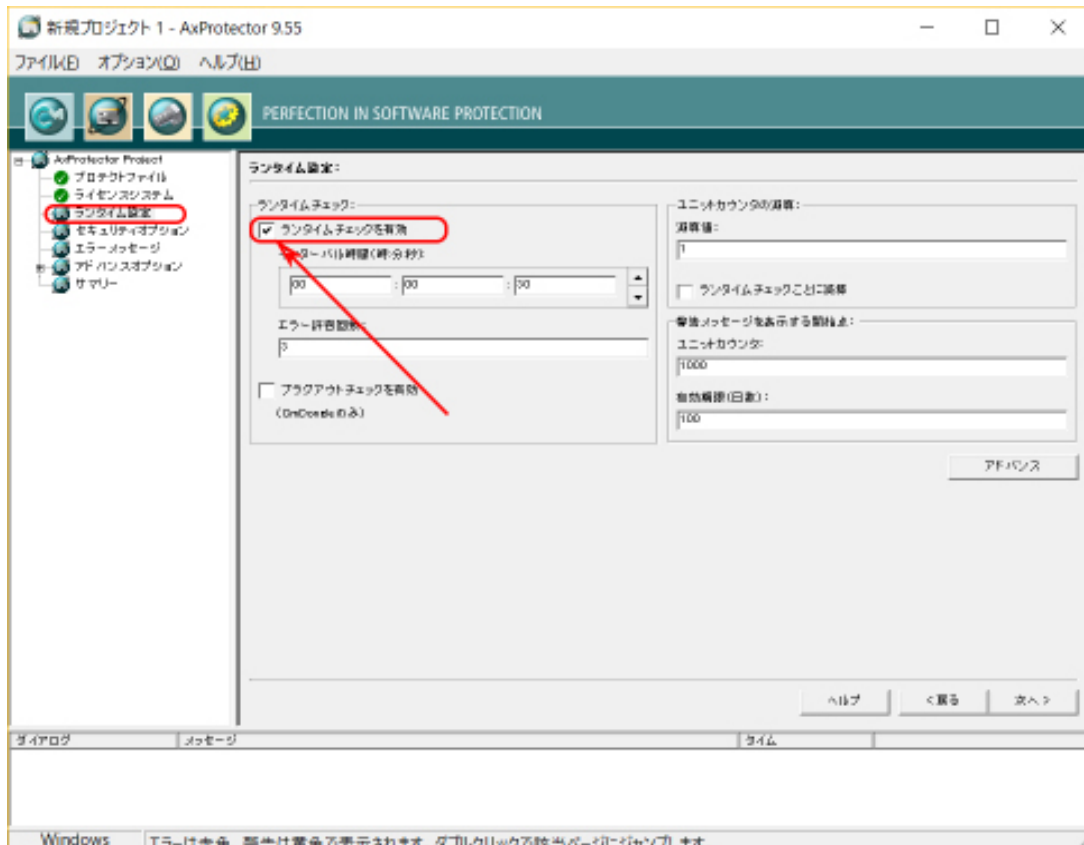
同一クライアント上でのアプリケーションの重複起動を防止します。

○ ユーザー制限なし(ユーザー数無制限)

起動に必要なコードメータキーがネットワーク上で見つければ、ライセンス数の制限にかかわらずアプリケーションが起動します。ライセンス数の制約を受けません。

②「ランタイムチェックを有効」を指定する

自動暗号化ツール「AxProtector」の「ランタイム設定」で「ランタイムチェックを有効」にチェックを入れます。この項目を指定しないと、アプリケーションが起動したあとにサーバークラスタが解放され、ネットワークカウンタが戻り、ライセンス制御ができなくなります。ランタイムチェックを有効にすると、アプリケーションが終了するまでネットワークカウンタが消費された状態になります。ネットワーク機能を利用する場合は、必ず「ランタイムチェックを有効」を指定してください。インターバル時間は自由に設定できます。



[注意!!!]

「ランタイムチェックを有効」にチェックを入れない場合、クライアント数は無制限になりますのでご注意ください。

[NOTE]

AxProtectorを使用せずに、CodeMeter Core APIまたはWUPIにてプロテクトする場合は、CmReleaseまたはWupiFreeLicenseが実行されるまでネットワークライセンスが保持されます。

Chapter 8

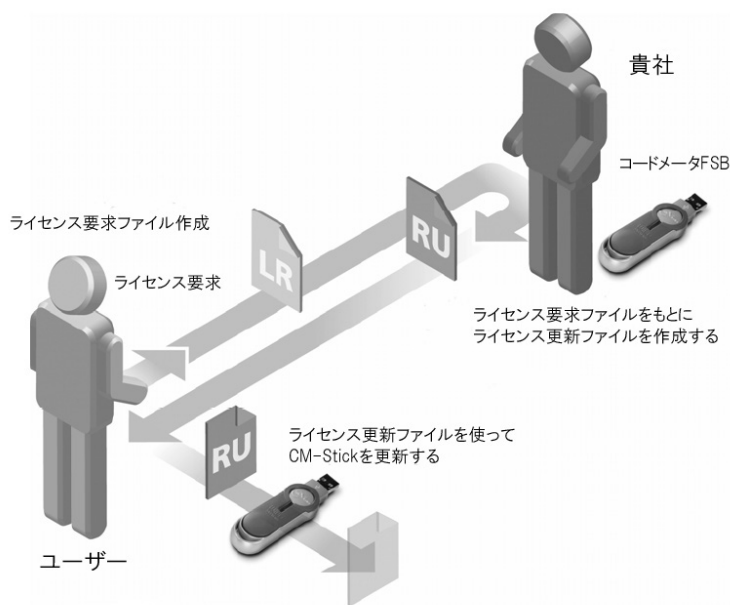
リモートアップデート機能について

- 8-1. リモートアップデート機能とは
- 8-2. リモートアップデート作業の流れ
- 8-3. ライセンス要求ファイルの作成（ユーザー側）
- 8-4. ライセンス更新ファイルの作成（貴社側）
- 8-5. CmDongle を更新する（ユーザー側）

8-1. リモートアップデート機能とは

リモートアップデート機能とは、ユーザー側にあるコードメータキー(CmDongle)の内容をファイル操作で更新する機能のことです。基本的に、コードメータキー(CmDongle)の内容を変更するには、コードメータFSB(CmFSB)を使ってローカルPC上で作業する必要がありますが、一度ユーザーに配布したコードメータキー(CmDongle)を変更するたびに送り返してもらうことは時間とコストの面からあまり得策ではありません。

コードメータのリモートアップデート機能を使うことで、更新ファイルをメールなどでやりとりすることにより、ユーザー先のコードメータキー(CmDongle)の内容を更新することが可能になります。プロダクトコードの追加、使用期限の更新、ユニットカウンタの更新または削除など、必要な時にスピーディに対応が可能になります。

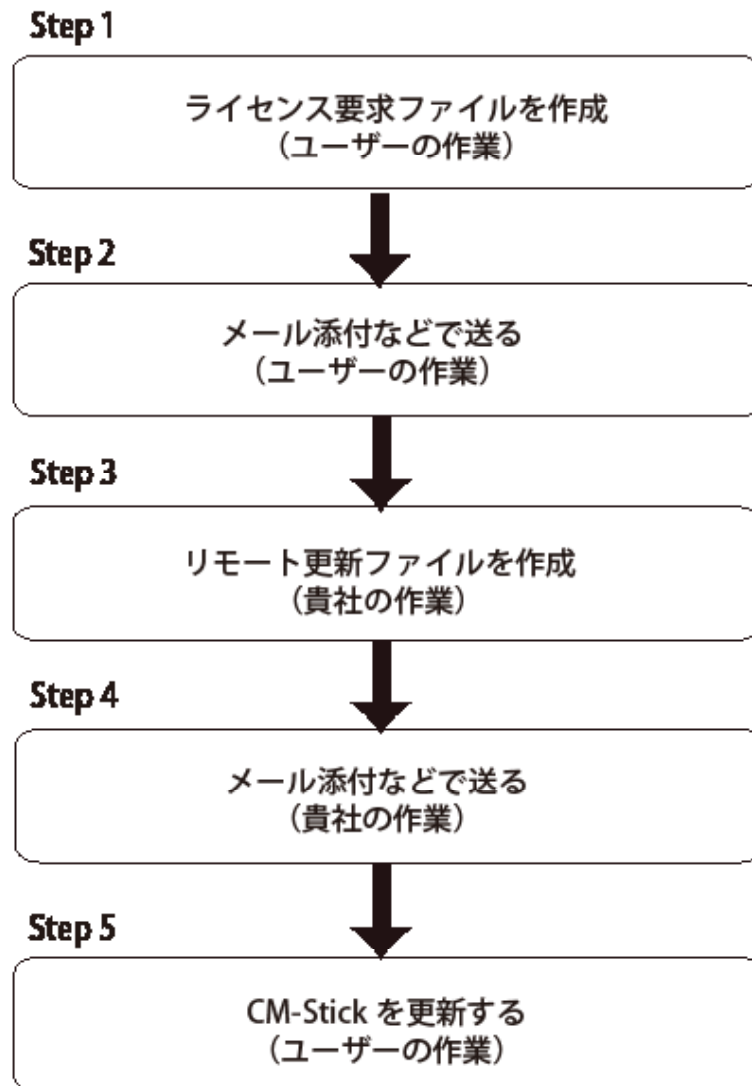


ユーザーに配布したライセンス更新ファイルは指定したコードメータキー(CmDongle)に対して1回だけ使用可能です。同じコードメータキー(CmDongle)に対して2回以上使用したり、別のコードメータキー(CmDongle)に使用することができないため、確実にセキュリティを確保できます。

また、ライセンス更新ファイルを作成するには、貴社のコードメータFSB(CmFSB)が必ず必要になるため、第三者が勝手にライセンス更新ファイルを作成することができません。

8-2. リモートアップデート作業の流れ

まず、CmDongleのライセンス要求ファイルをユーザー自身が作成します。それを、メール添付などで、貴社に送ってもらいます。貴社は、ユーザーのライセンス要求ファイルをベースにライセンス更新ファイルを作成します。ライセンス更新ファイル作成には貴社のコードメータFSB(CmFSB)が必要です。そのライセンス更新ファイルをメール添付などでユーザーに送ります。ユーザーは、ライセンス更新ファイルを使ってCmDongleの内容を更新します。



8-3. ライセンス要求ファイル (.WibuCmRaC) の作成 (ユーザー側)

ライセンス要求ファイル(xxxxx.WibuCmRaC)はコードメータコントロールセンターから作成します。

① 更新する CmDongle を PC に装着します。

更新するCmDongleをPCに装着します。



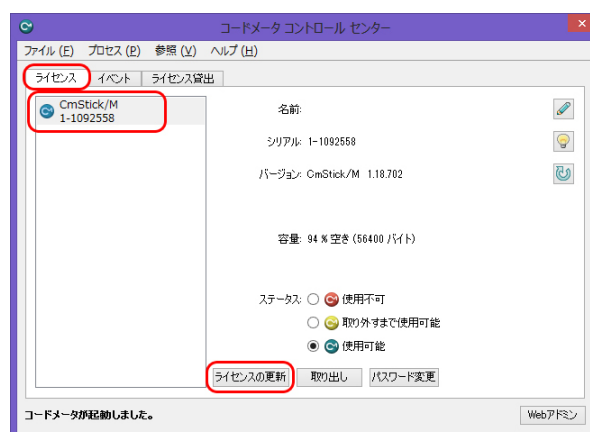
② コードメータコントロールセンターを開く

コードメータコントロールセンターを開きます。画面右下のコードメータコントロールセンターのアイコンをクリックするか、【スタート】→【すべてのプログラム】→【CodeMeter】→【CodeMeter Control Center】から起動します。



③ 「ライセンスの更新」をクリックする

「ライセンス」のタブを選択し、更新するCmDongleをマウスで選択し、「ライセンスの更新」ボタンをクリックします。



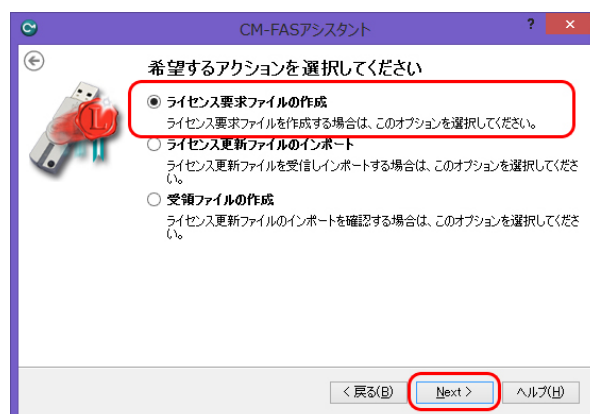
④ CM-FAS アシスタント画面が表示される

「CM-FASアシスタント」画面が表示されますので、「Next」ボタンをクリックして次に進めます。



⑤ ライセンス要求ファイルの作成を選択

希望するアクションを選択する画面で、「ライセンス要求ファイルの作成」を選択します。



⑥ オプションを選択する

ライセンス要求には2つのオプションがあります。

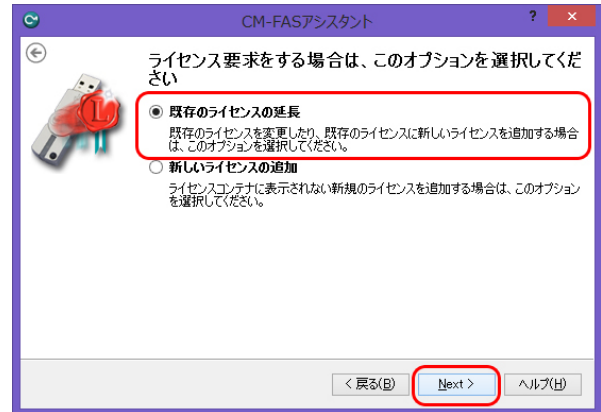
- 既存のライセンスの延長
- 新しいライセンスの追加

「既存のライセンスの延長」とは、すでに登録されているファームコードに対してプロダクトの追加や有効期限の更新などを行うオプションです。

「新しいライセンスの追加」とは、まだ登録されていない新規のファームコードを追加するオプションです。

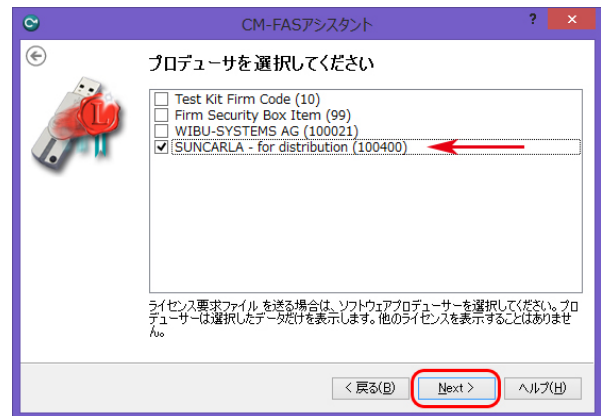
[NOTE]

すでに登録されているファームコードに対して、新規のプロダクトコードを追加する場合は、「既存のライセンスの延長」を選択します。ここでは、「既存のライセンスの延長」を選択後、「Next」ボタンをクリックします。



⑦ ファームコードを選択する

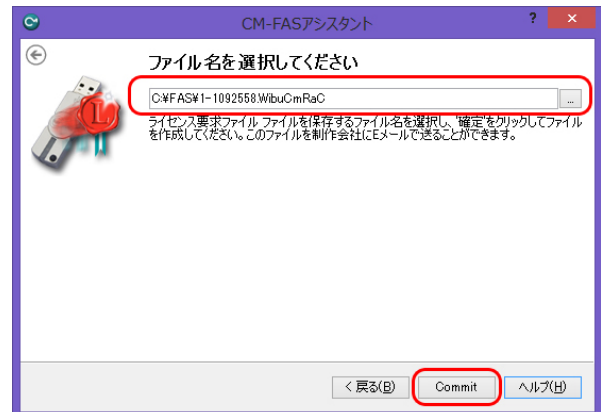
プロデューサー選択画面で、更新を要求するファームコードを選択し、「Next」ボタンをクリックします。異なるファームコードが複数表示されている場合は、更新を要求するファームコードを選択してください。右の例では、ファームコード = 100400に対するライセンス要求ファイルを指定しています。



⑧ 保存先を指定する

ライセンス要求ファイルの名前と保存先を指定します。ライセンス要求ファイルの拡張子は".WibuCmRaC"です。拡張子は変更しないでください。また、ファイル名の左部には、CmDongleのシリアル番号が表示されます。ファイル名は、デフォルトのままご使用されることをお勧め致します。(拡張子以外のファイル名を変更することは可能です。)

設定後、「Commit」ボタンをクリックします。



⑨ ライセンス要求ファイルが作成される

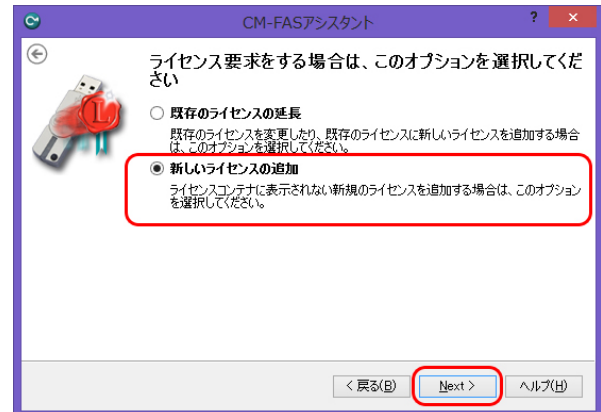
"⑧保存先を指定する"で指定したフォルダにライセンス要求ファイルが作成されます。エクスプローラ等を使って確認してください。

「完了」ボタンをクリックすると、コードメータコントロールセンター画面に戻ります。

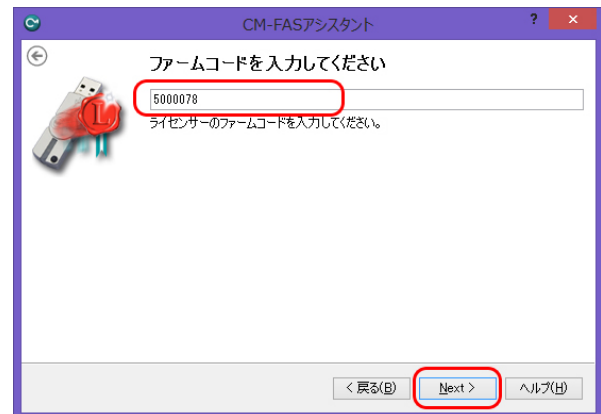


▶ ⑥オプションで「新しいライセンスの追加」を選択した場合

CmStickに新規のファームコードを登録する場合は、「新しいライセンスの追加」を選択し、「Next」ボタンをクリックします。



ファームコード入力画面で、要求するファームコードを半角数字で入力し、「Next」ボタンをクリックします。(ここでは、ファームコード= 5000078を要求)



ライセンス要求ファイルの名前と保存先を指定します。ライセンス要求ファイルの拡張子は ".WibuCmRaC"です。拡張子を変更しないでください。また、ファイル名の左部には、CmDongleのシリアル番号が表示されます。ファイル名は、デフォルトのままご使用されることをお勧め致します。(拡張子以外のファイル名を変更することは可能です。)



設定後、「Commit」ボタンをクリックします。指定したフォルダにライセンス要求ファイルが作成されます。エクスプローラ等を使って確認してください。

「完了」ボタンをクリックすると、コードメータコントロールセンター画面に戻ります。



8-4. ライセンス更新ファイルの作成（貴社側）

ユーザーから送られたライセンス要求ファイル（コンテキストファイル）をもとに、ライセンス更新ファイルを作成します。ライセンス更新ファイルの作成は2つの方法があります。

- コードメータライセンスエディタを使用する方法
- CmBoxPgm.exeを使用する方法（コマンドライン環境）

コードメータライセンスエディタを使用する方法

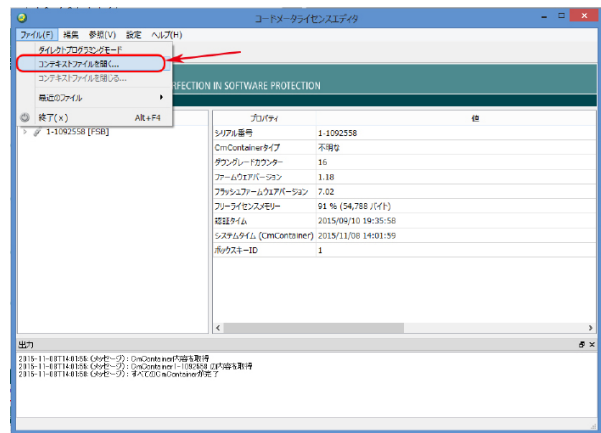
コードメータライセンスエディタを使って、ライセンス更新ファイルを作成します。ライセンス更新ファイルを作成するには、貴社のコードメータFSB(CmFSB)が必要になります。

① 貴社のコードメータ FSB を PC に装着する

貴社のコードメータFSB(CmFSB)をPCに装着します。作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

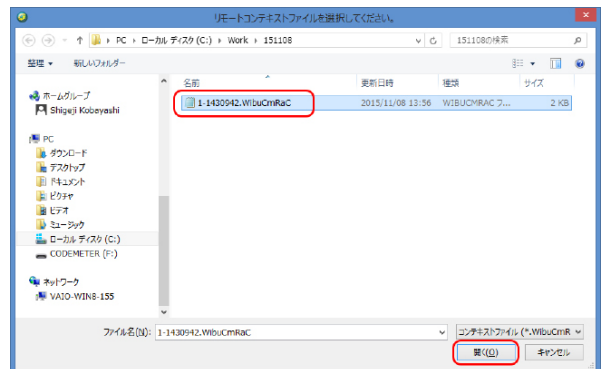
② コードメータライセンスエディタを起動

【スタート】→【すべてのプログラム】→【CodeMeter】→【Tools】→【CodeMeter License Editor】をクリックし、コードメータライセンスエディタを起動します。コードメータスタートセンターからも起動できます。起動後、「ファイル」メニューから「コンテキストファイルを開く...」を選択しクリックします。



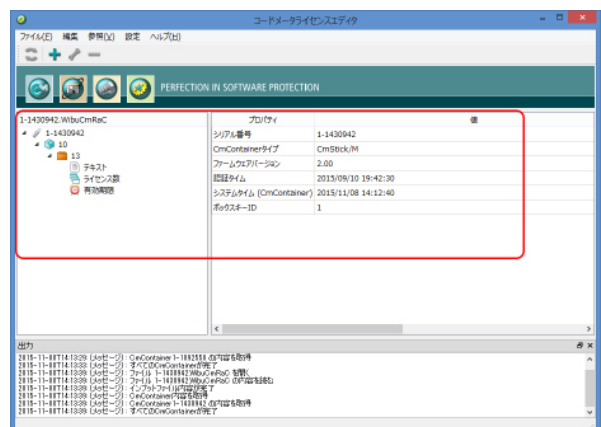
③ ライセンス要求ファイルを選択する

「リモートテキストファイルを選択してください」画面で、ユーザーから送られてきたライセンス要求ファイルを選択し、「開く」ボタンをクリックします。



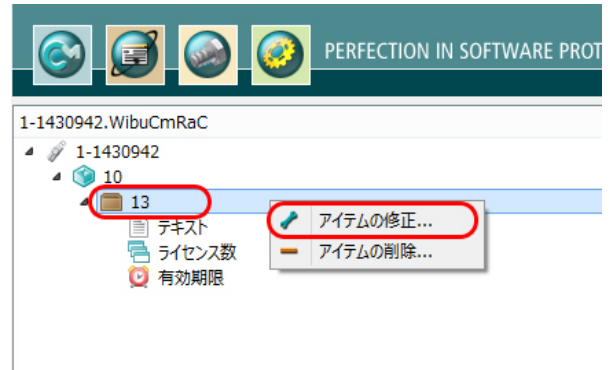
④ ライセンス要求ファイルの内容を確認

左ペインにユーザーから送られてきたライセンス要求ファイルの内容が表示されます。



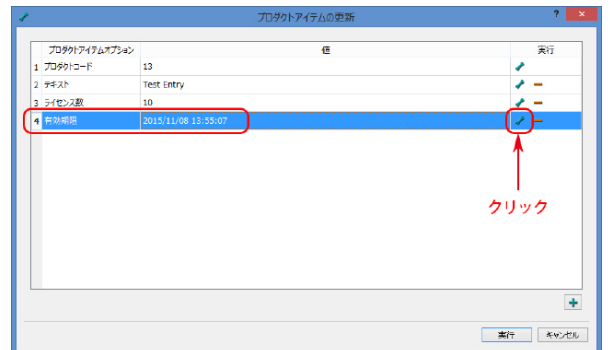
⑤ ライセンス更新ファイルを作成する

プロダクトコード(=13)にマウスを合わせ、右クリックで「アイテムの修正...」を選択します。

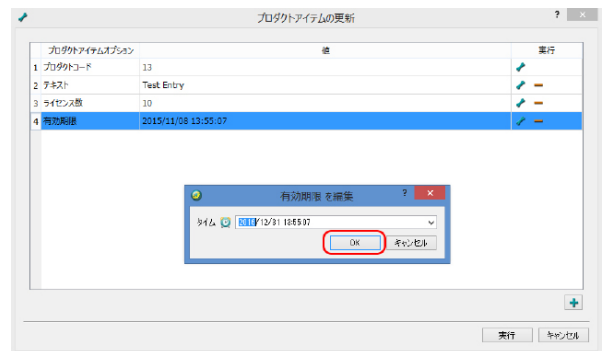


「プロダクトアイテムの更新」画面で更新する内容を編集します。右端の ボタンをクリックします。

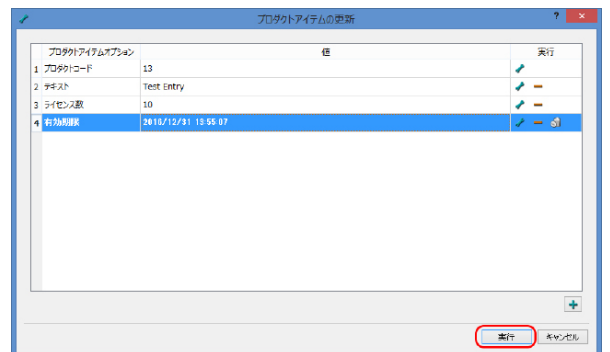
ここでは、有効期限を2016年12月31日に変更します。



「有効期限を編集」画面でOKボタンをクリックします。



「プロダクトアイテムの更新」画面で「実行」ボタンをクリックすると、ライセンス更新ファイル(.WibuCmRaU)がライセンス要求ファイルと同じフォルダに作成されます。



[NOTE]

ライセンス更新ファイル(.WibuCmRaU)と同時に、ライセンス修正ファイル(.WibuCmRaM)も作成されます。(詳細は後述)

名前	更新日時	種類	サイズ
1-1430942.WibuCmRaC	2015/11/08 14:12	WIBUCMRAC フ...	2 KB
1-1430942.WibuCmRaM	2015/11/08 14:56	WIBU Control File	2 KB
1-1430942.WibuCmRaU	2015/11/08 14:56	WIBU Control File	1 KB

⑥ ライセンス更新ファイルをユーザーに送る

作成されたライセンス更新ファイルをユーザーにメール添付などで送ります。ライセンス更新ファイルの拡張子は、".WibuCmRaU"で、ファイル名の左部は変わりません。

X-XXXXXXXX.**WibuCmRaU**

CmBoxPgm.exe を使用する方法（コマンドライン環境）

次に、CmBoxPgm.exeを使って、コマンドライン環境にてライセンス更新ファイルを作成する方法をご説明いたします。ライセンス要求ファイルからライセンス更新ファイルを自動的に作成するバッチ処理が可能になります。

① 貴社のコードメータ FSB を PC に装着します。

作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

② コマンドプロンプトを開きます。

【スタート】→【すべてのプログラム】→【アクセサリ】→【コマンドプロンプト】を起動します。以後、コマンドライン上での操作になります。

③ CD コマンドでライセンス要求ファイルが存在するフォルダをカレントにする。

[例] コマンドプロンプトを開いて、
>CD ¥FAS ↓

* CD = Change Directory

④ ライセンス更新ファイルを作成する

ファームコード=100027の中に、プロダクトコード= 13 を新規登録するライセンス更新ファイルを作成します。ライセンス更新ファイルを作成するパラメータは"/RA:"です。"/RA:"に続いてライセンス要求ファイルを指定します。コマンドラインから、下記のようにタイプしEnterキーを押します。2-1264652と拡張子WibuCmRaCとの間にピリオド"."があることにご注意ください。ライセンス要求ファイルの左部は、CmDongleのシリアル番号になります。

CmBoxPgm /RA:2-1264652.WibuCmRaC /F100027 /P13 /CA ↓（↓はEnterキー）

2-1264652.WibuCmRaCファイルと同じフォルダに、ライセンス更新ファイル2-1264652.WibuCmRaUが作成されます。ライセンス更新ファイルの拡張子は"WibuCmRaU"になります。

また、既存のプロダクトコードに設定された使用有効期限を更新する場合は、

CmBoxPgm /RA:2-1264652.WibuCmRaC /F100027 /P300 /PETA2015Dec31 /CU ↓

ファームコード=100027、プロダクトコード=300のプロダクトアイテムに対して、有効期限2015年12月31日が設定（更新）されます。更新の場合は、"/CA"でなく"/CU"を使用します。CmBoxPgm.exeおよびパラメータの使い方は、「Chapter 10 CmBoxPgmの使い方」を参照してください。

⑤ ライセンス更新ファイルをユーザーに送る

作成したライセンス更新ファイルをメール添付などでユーザーに送ります。

[参考]

ライセンス貸出用サーバー CmDongle を作成する更新ファイル

コードメータのリモートアップデート機能を使って、ライセンス要求ファイルからコードメータライセンス貸出用サーバーCmDongleを作成する更新ファイルを作成することが可能です。作業はコマンドライン環境で行います。パラメータは、大文字・小文字両方使用可能です。また、ライセンスエディタ (License Editor)上では作成できません。

コマンドライン上から下記のように実行します。

```
CmBoxPgm /RA:2-1264652.WibuCmRaC /F100027 /FT:"License Borrowing Server"
/CAU /P13 /PT:"License Borrowing with CmStick" /PLQ10
/BLS:cm,100027,13,0,5,28800,0x12345678 /CA
```

[パラメータの説明]

/RA

指定したライセンス要求ファイルからライセンス更新ファイルを作成する。

(例) /RA:2-1264652.WibuCmRaC

/RA:の次には、ユーザーから取得したライセンス要求ファイルを指定する。

/F

ファームコード(Firm Code)を指定する。

/FT

ファームアイテムテキストを指定する。

(例) /FT:"License Borrowing Server"

/CAU

CmDongleの既存のエントリを更新する。既存のエントリが存在しない場合は、新規で追加する。

/P

プロダクトコード(Product Code)を指定する。

/PT

プロダクトアイテムテキストを指定する。

(例) /PT:"License Borrowing with CmStick"

/PLQ

与えるライセンス数を指定する。

(例) /PLQ10 (10ライセンス数を設定する)

/BLS

ライセンス貸出用サーバーCmDongleを作成する。

(Syntax)

/BLS:[cm|ca],<fc>,<pc>,<fm>,<lqClient>,<duration> [, serverID]

[cm|ca]

cm=CodeMeter, ca=CodeMeterAct

<fc>

fc=Firm Code (ファームコード)

<pc>

pc=Product Code (プロダクトコード)

<fm>

fm=Feature Map (フィーチャーマップ/フィーチャーコード)

<lqClient>

貸出を許可するクライアント数

ここで設定したクライアント数が実際に貸出可能なクライアント数になります。この数字は当然のことながら/PLQで指定したライセンス数を超えて設定することはできません。

(例) /BLS:cm,100027,13,0,5,28800,0x12345678

ここでは、貸出を許可するクライアント数を"5"に設定しています。

<duration>

最大貸出期間を設定する。設定する数字は分(minutes)を使用します。

(例) /BLS:cm,100027,13,0,5,28800,0x12345678

ここでは、28800(minutes)=480(hours)=20(days)に設定しています。

実際のライセンス貸出期間は、Webアドミンの[構成]/[借用]ページの最大貸出期間で設定された数字が反映されます。Webアドミン上で何も設定されていない場合は、ここで指定した<duration>の数字が反映されます。

[, serverID]

サーバーIDを8バイトで0x12345678の形式で任意に割り当てます。

このサーバーIDは、ライセンス貸出用クライアントCmDongleで割り当てるサーバーIDを一致する必要があります。

(例) /BLS:cm,100027,13,0,5,28800,0x12345678

ここでは、サーバーIDを0x12345678に設定しています。

/CA

新しいエントリを追加します。

ライセンス貸出用クライアント CmDongle を作成する更新ファイル

コードメータのリモートアップデート機能を使って、ライセンス要求ファイルからコードメータライセンス貸出用クライアントCmDongleを作成することが可能です。

```
CmBoxPgm /RA:2-1264652.WibuCmRaC /F100027 /P13
/BLC:cm,100027,13,0x12345678 /PT:"Borrow License Client CodeMeter" /CA
```

[パラメータの説明]

/RA

指定したライセンス要求ファイルからライセンス更新ファイルを作成する。

(例) /RA:2-1264652.WibuCmRaC

/RA:の次には、ユーザーから取得したライセンス要求ファイルを指定する。

/F

ファームコード(Firm Code)を指定する。

/P

プロダクトコード(Product Code)を指定する。

/BLC

ライセンス貸出用クライアントCmDongleを作成する。

(Syntax)

```
/BLC:[cm|ca],<fc>,<pc> [ , serverID ]
```

[cm|ca]

cm=CodeMeter, ca=CodeMeterAct

<fc>

fc=Firm Code (ファームコード)

<pc>

pc=Product Code (プロダクトコード)

[, serverID]

ライセンス貸出用サーバーCmDongleに割り当てられているサーバーIDを指定します。

8バイトで0x12345678の形式で指定します。

/PT

プロダクトアイテムテキストを指定する。

(例) /PT:"Borrow License Client CodeMeter"

/CA

新しいエントリを追加します。

8-5. CmDongle を更新する（ユーザー側）

ライセンス更新ファイル"2-1264652.WibuCmRaU"を使って、CmDongleの内容を更新します。
CmDongleの更新は、コードメータコントロールセンターを使用します。

① CmDongle を PC に装着する

装着するCmDongleは、ライセンス要求ファイルを作成したときのCmDongleである必要があります。
別のCmDongleを装着してもシリアル番号が一致しないため更新作業は行われません。

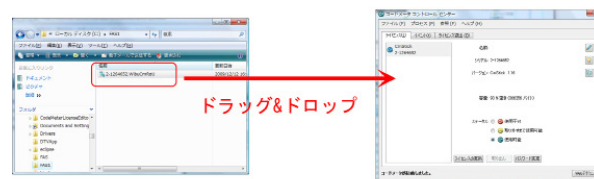
② コードメータコントロールセンターを開く

PC画面右下のコードメータコントロールセンターのアイコンをクリックして、コードメータコントロールセンターを開きます。アイコンが見つからない場合は、【スタート】→【すべてのプログラム】→【CodeMeter】→【CodeMeter Control Center】から起動します。



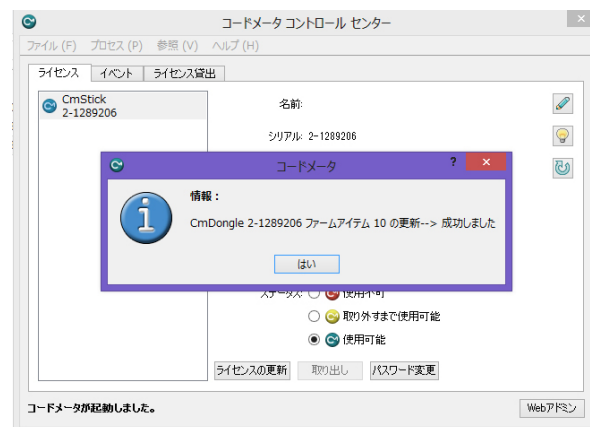
③ ライセンス更新ファイルをドラッグする

エクスプローラ上からライセンス更新ファイルを選択し、マウスでコードメータコントロールセンターの画面にドラッグ&ドロップします。CmDongleの更新作業が開始されます。



④ CmDongle が更新された

右の画面が表示され、CmDongleが更新されます。WebAdmin上でCmDongleの内容を確認してください。



使用したライセンス更新ファイル2-1264652.WibuCmRaUは、一度しか使用できません。同じCmDongleに対して2回使用することはできません。また、別のCmDongleに対しては最初から使用できません。これにより、確実なセキュリティを維持しています。

[NOTE]

一度使用したライセンス更新ファイル(2-1264652.WibuCmRaU)は2回使用できませんが、再度、ライセンス要求ファイルを作り、それに対してライセンス更新ファイルを作成することで無限に更新作業が可能です。また、同じファイル名を使用することも可能です。

また、ライセンスの更新は、コードメータコントロールセンターの「ライセンスの更新」ボタンから行うこともできます。

① CmDongle を PC に装着する

装着するCmDongleは、ライセンス要求ファイルを作成したときのCmDongleである必要があります。異なるCmDongleを装着してもシリアル番号が一致しないため更新作業は行われません。

② コードメータコントロールセンターを開く

画面右下のコードメータコントロールセンターのアイコンをクリックして、コードメータコントロールセンターを開きます。アイコンが見つからない場合は、【スタート】→【すべてのプログラム】→【CodeMeter】→【CodeMeter Control Center】から起動します

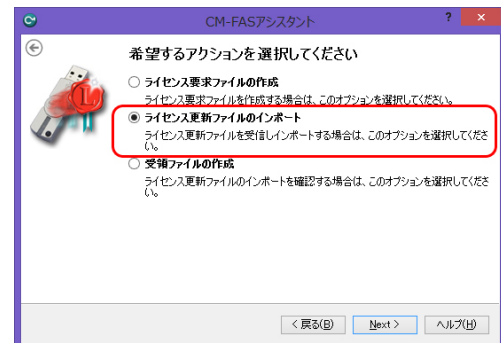
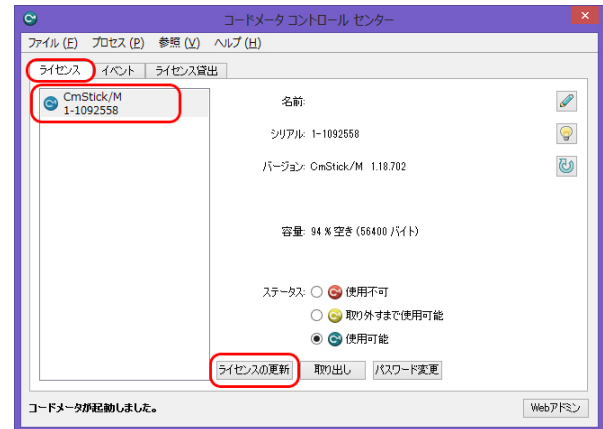
③ 「ライセンスの更新」 ボタンをクリックする

「ライセンスの更新」ボタンをクリックします。CM-FASアシスタントが起動しますので、メッセージに従って進めます。

「ライセンス更新ファイルのインポート」を選択し、「Next」ボタンをクリックします。

ライセンス更新ファイル(xxxxx.WibuCmRaU)を選択し、「Commit」ボタンをクリックすると、CmDongleの更新が開始されます。

ライセンス更新に成功すると右のメッセージが表示されます。



これで、CmDongleは更新されました。
念のため、WebAdminで確認してください。



Chapter 9

CodeMeter License Editor の使い方

- 9-1. CodeMeter License Editor について
- 9-2. ライセンスエディタの画面構成
- 9-3. ライセンス要求ファイルからライセンス更新ファイルを作成
- 9-4. ライセンス修正ファイルからライセンス更新ファイルを作成
- 9-5. ファームアイテムの追加と修正
- 9-6. プロダクトアイテムの追加と修正

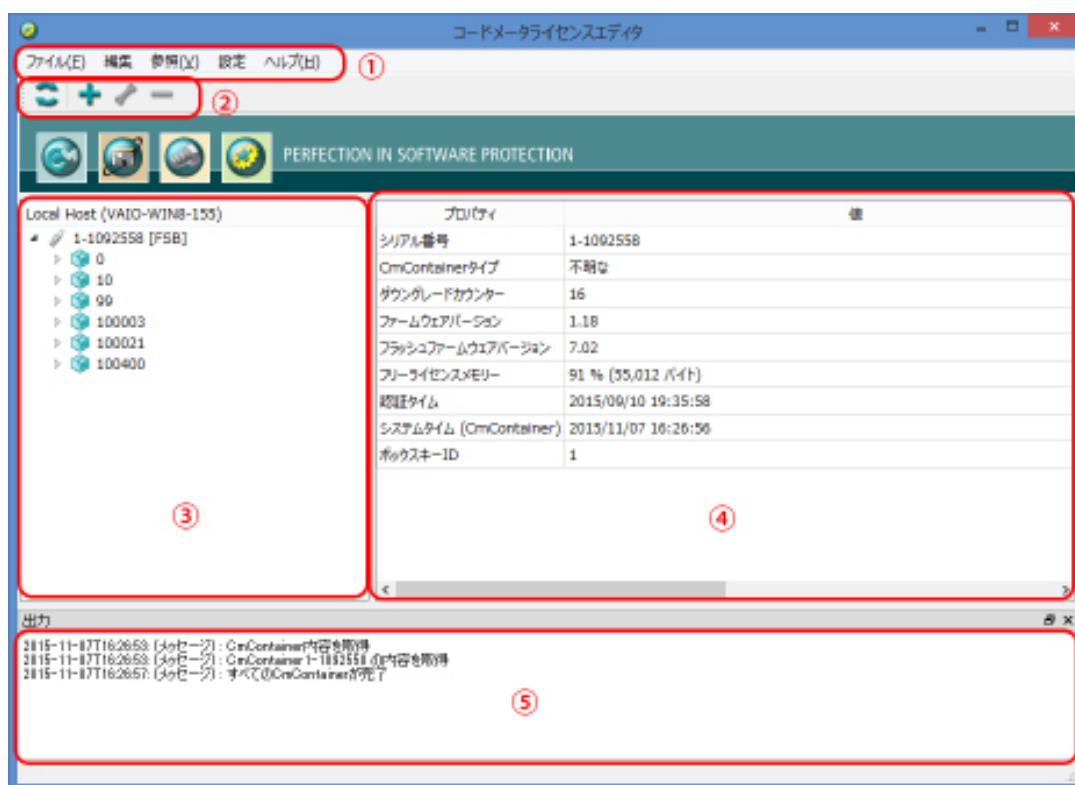
9-1. CodeMeter License Editor について

コードメータライセンスエディタ (CodeMeter License Editor) は、CmDongle内のライセンスやライセンスコンポーネント (ファームアイテム、プロダクトアイテム、プロダクトアイテムオプション等) を作成・編集・削除するツールです。リモートプログラミング (CmFAS: CodeMeter Field Activation Service) にも対応しています。

コードメータライセンスエディタ (CodeMeter License Editor) は、少量のCmDongleを扱うときに便利です。「スタート」/「すべてのプログラム」/「CodeMeter」/「Tools」/「CodeMeter License Editor」から起動できます。

9-2. ライセンスエディタの画面構成

コードメータライセンスエディタのGUI画面は5つの部分から構成されます。



- ①メニューバー
- ②シンボルバー
- ③ツリービューウィンドウ
- ④表示ウィンドウ
- ⑤出カウィンドウ

メニューバー

「ファイル(F)」メニュー

ダイレクトプログラミングモード

CmDongleを直接編集します。ライセンスエディタの起動直後はダイレクトプログラミングモードの状態になります。

コンテキストファイルを開く...

コンテキストファイル (ライセンス要求ファイル) *.WibuCmRaC または *.WibuCmRaM ファイルを開きます。ツリービューウィンドウにライセンス要求ファイルの内容が表示されますので、必要に応じて内容を編集し、ライセンス更新ファイルを作成します。

コンテキストファイルを閉じる...

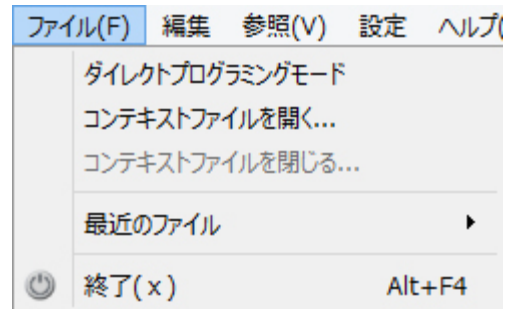
コンテキストファイル (ライセンス要求ファイル) を閉じます。

最近のファイル

最近使用したコンテキストファイル (ライセンス要求ファイル) がリスト表示されます。

終了

コードメータライセンスエディタを終了します。
<Alt+F4>で終了することも可能です。



「編集」メニュー

アイテムの追加

新規のファームアイテムを追加します。<Ctrl+A>

アイテムの修正

既存のファームアイテムの修正を行います。
<Ctrl+M>

アイテムの削除

既存のファームアイテムまたはプロダクトアイテムを削除します。<Ctrl+D>

編集	参照(V)	設定	ヘルプ(H)
+	アイテムの追加		Ctrl+A
↶	アイテムの修正		Ctrl+M
-	アイテムの削除		Ctrl+D
↻	リフレッシュ		Ctrl+R

[NOTE]

正規のファームコードを削除することはできません。(ファームコード=10は削除できます)

リフレッシュ

ローカル接続しているCmDongle情報をリフレッシュします。<Ctrl+R>

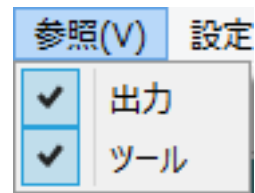
「参照」メニュー

出力

出力ウィンドウを表示します。

ツール

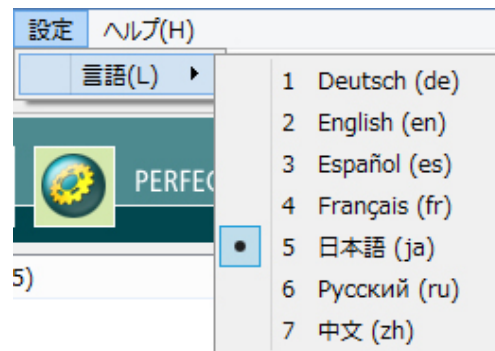
シンボルバーを表示します。



「設定」メニュー

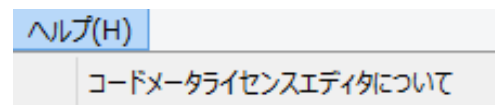
言語(L)

表示する言語を設定します。



「ヘルプ(H)」メニュー

コードメータライセンスエディタについて
コードメータライセンスエディタのバージョン情報
を表示します。



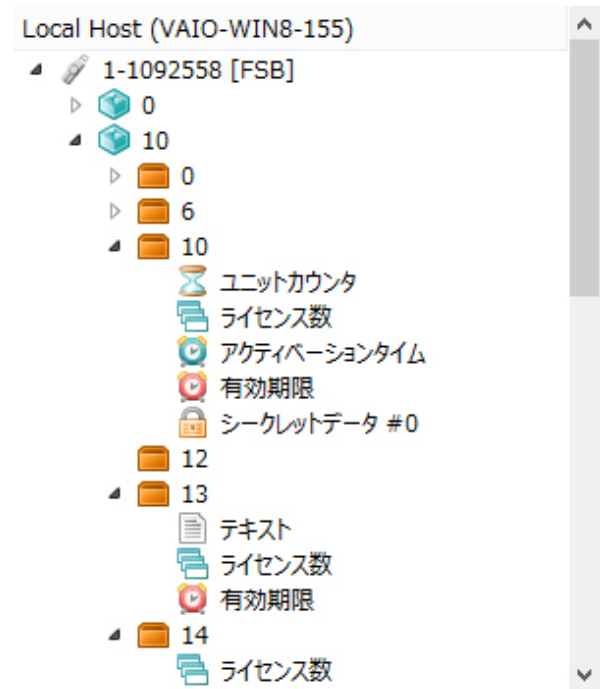
シンボルバー

標準機能をシンボルバーから実行することができます。



ツリービューウィンドウ

PCに接続されているCmDongleの内容をウィンドウ内に表示します。📁 ▶ をクリックすることで、CmDongleの内容が展開されたり閉じたりします。



表示ウィンドウ

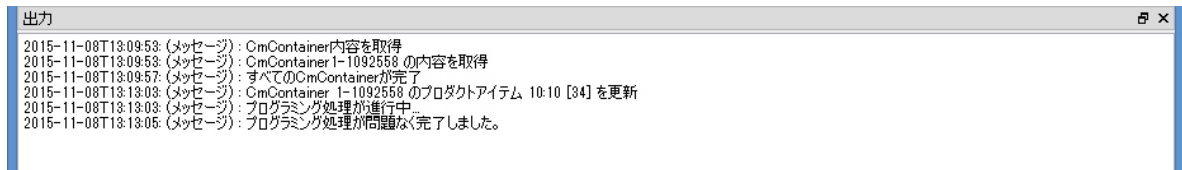
CmDongleの詳細が表示されます。

プロパティ	値
プロダクトコード	10
プロダクトアイテムリファレンス	34
フィーチャーマップ	0x0000000f
ユニットカウンタ	10
アクティベーションタイム	2015/12/01 12:59:08
有効期限	2016/12/31 13:00:30
ライセンス数	純粋ローカルライセンス
ライセンス情報	〇〇商事



出力ウィンドウ

実行結果やエラーメッセージが表示されます。



The screenshot shows a window titled "出力" (Output) with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains the following log messages:

```
2015-11-08T13:09:53: (メッセージ) : CmContainer内容を取得  
2015-11-08T13:09:53: (メッセージ) : CmContainer 1-1092558 の内容を取得  
2015-11-08T13:09:57: (メッセージ) : すべてのCmContainerが完了  
2015-11-08T13:13:03: (メッセージ) : CmContainer 1-1092558 のプロダクトアイテム 10:10 [34] を更新  
2015-11-08T13:13:03: (メッセージ) : プログラミング処理が進行中  
2015-11-08T13:13:05: (メッセージ) : プログラミング処理が問題なく完了しました。
```


9-3. ライセンス要求ファイルからライセンス更新ファイルを作成する

ユーザーから送られてきたライセンス要求ファイル(.WibuCmRaC)をもとにライセンス更新ファイル(.WibuCmRaU)ファイルを作成します。(「コードメータのリモートアップデート機能」参照)

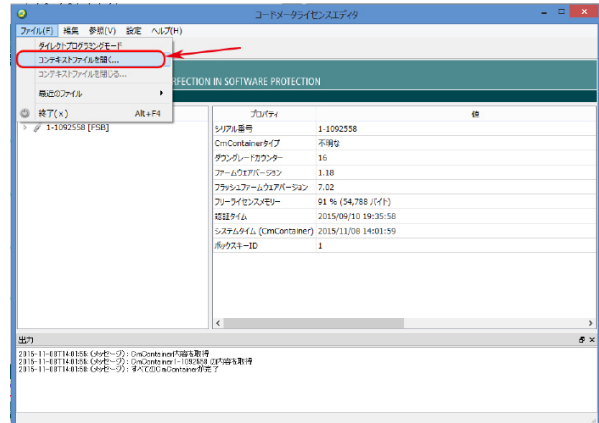
ライセンス更新ファイル(.WibuCmRaU)を作成するには、該当するCmFSBをPCに装着する必要があります。また、ライセンス更新ファイル(.WibuCmRaU)は該当するCmFSBがあれば、無制限に作成可能です。(CmDongleの場合、CmActLicenseの場合は異なります)

① 貴社のコードメータFSBをPCに装着する

貴社のコードメータFSB(CmFSB)をPCに装着します。作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

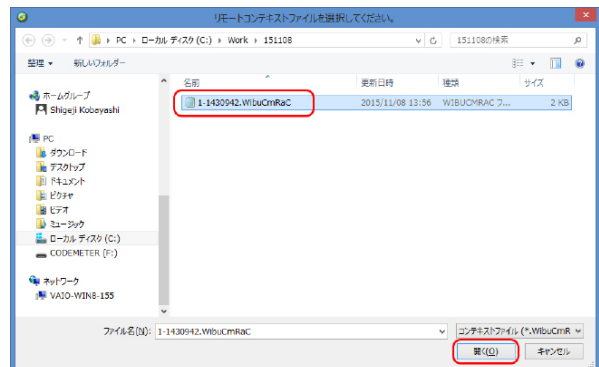
② コードメータライセンスエディタを起動

【スタート】→【すべてのプログラム】→【CodeMeter】→【Tools】→【CodeMeter License Editor】をクリックし、コードメータライセンスエディタを起動します。コードメータスタートセンターからも起動できます。起動後、「ファイル」メニューから「コンテキストファイルを開く...」を選択しクリックします。



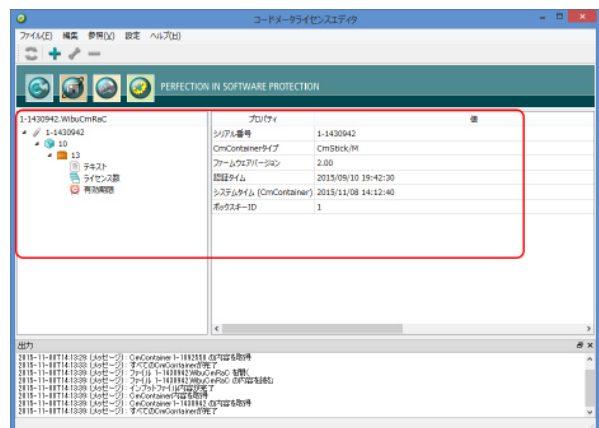
③ ライセンス要求ファイルを選択する

「リモートテキストファイルを選択してください」画面で、ユーザーから送られてきたライセンス要求ファイルを選択し、「開く」ボタンをクリックします。



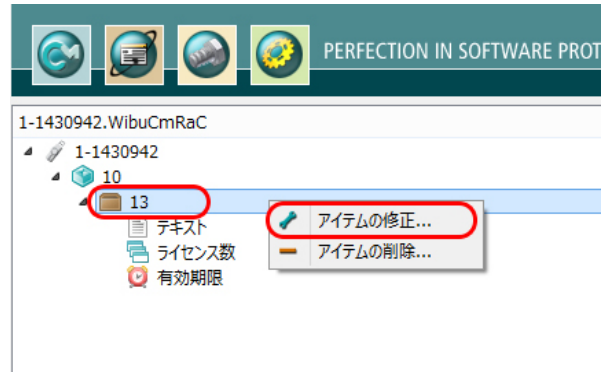
④ ライセンス要求ファイルの内容を確認


左ペインにユーザーから送られてきたライセンス要求ファイルの内容が表示されます。



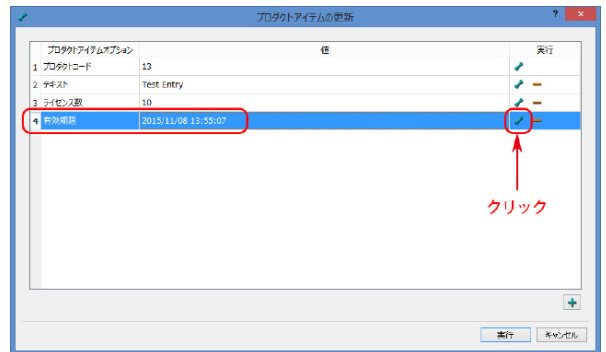
⑤ ライセンス更新ファイルを作成する

プロダクトコード(=13)にマウスを合わせ、右クリックで「アイテムの修正...」を選択します。

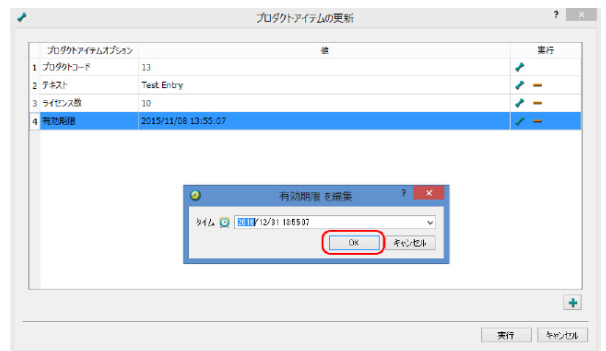


「プロダクトアイテムの更新」画面で更新する内容を編集します。右端の  ボタンをクリックします。

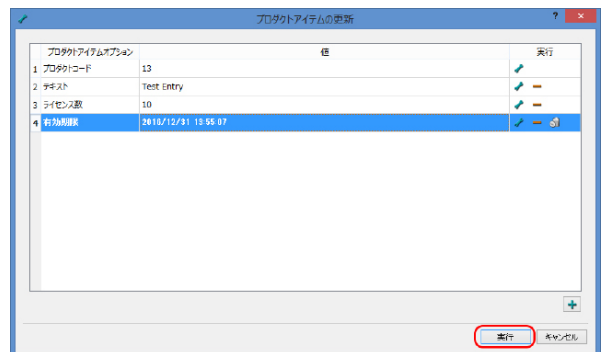
ここでは、有効期限を2016年12月31日に変更します。



「有効期限を編集」画面でOKボタンをクリックします。



「プロダクトアイテムの更新」画面で「実行」ボタンをクリックすると、ライセンス更新ファイル(.WibuCmRaU)がライセンス要求ファイルと同じフォルダに作成されます。



[NOTE]

ライセンス更新ファイル(.WibuCmRaU)と同時に、ライセンス修正ファイル(.WibuCmRaM)も作成されます。(詳細は後述)

名前	更新日時	種類	サイズ
1-1430942.WibuCmRaC	2015/11/08 14:12	WIBUCMRAC フ...	2 KB
1-1430942.WibuCmRaM	2015/11/08 14:56	WIBU Control File	2 KB
1-1430942.WibuCmRaU	2015/11/08 14:56	WIBU Control File	1 KB

⑥ ライセンス更新ファイルをユーザーに送る

作成されたライセンス更新ファイルをユーザーにメール添付などで送ります。ライセンス更新ファイルの拡張子は、".WibuCmRaU"で、ファイル名の左部は変わりません。

X-XXXXXXXXX.WibuCmRaU

[NOTE]

ライセンス更新ファイル(.WibuCmRaU)を作成する際に使用したライセンス要求ファイルは、つぎに作成するライセンス更新ファイル(.WibuCmRaU)には使用できません。その都度、ユーザーからライセンス要求ファイルを取得する必要があります。CmDongle内、ライセンス要求ファイル内、ライセンス更新ファイル内にそれぞれアップデートカウンタが組み込まれ、3者が一致しないとCmDongleの更新ができないからです。一度、ライセンス更新ファイルでCmDongle内を更新すると、CmDongle内のアップデートカウンタが繰り上がるために、その都度、ライセンス要求ファイルを取得する必要があります。ただし、後述するライセンス修正ファイル(.WibuCmRaM)を利用すると、ユーザーからライセンス要求ファイルを取得しなくても、ライセンス修正ファイル(.WibuCmRaM)から直接、つぎのライセンス更新ファイル(.WibuCmRaU)を作成することが可能になります。

[NOTE]

ライセンス更新ファイルを作成すると、同じフォルダにライセンス修正ファイル(.WibuCmRaM)が作成されます。つぎにライセンス更新ファイルを作る際、このライセンス修正ファイル(.WibuCmRaM)からライセンス更新ファイルを作成することが可能です。その都度、ユーザーからライセンス要求ファイルを送ってもらう必要が省けます。

9-4. ライセンス修正ファイルからライセンス更新ファイルを作成する

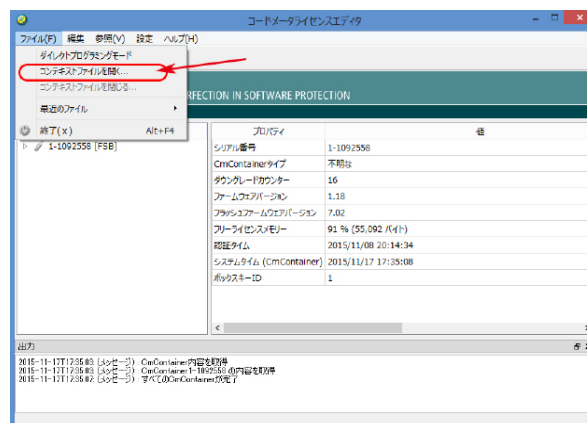
ライセンス修正ファイル(.WibuCmRaM)から、ライセンス更新ファイル(.WibuCmRaU)ファイルを作成します。ライセンス更新ファイル(.WibuCmRaU)を作成するには、該当するCmFSBをPCに装着する必要があります。

① 貴社のコードメータ FSB を PC に装着する

貴社のコードメータFSB(CmFSB)をPCに装着します。作業するPCには、すでにコードメータ開発キットがインストールされている必要があります。

② コードメータライセンスエディタを起動

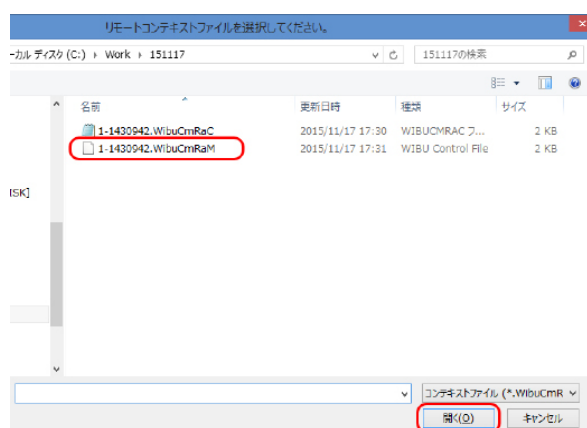
【スタート】→【すべてのプログラム】→【CodeMeter】→【Tools】→【CodeMeter License Editor】をクリックし、コードメータライセンスエディタを起動します。コードメータスタートセンターからも起動できます。起動後、「ファイル」メニューから「コンテキストファイルを開く...」を選択しクリックします。



③ ライセンス修正ファイルを選択する

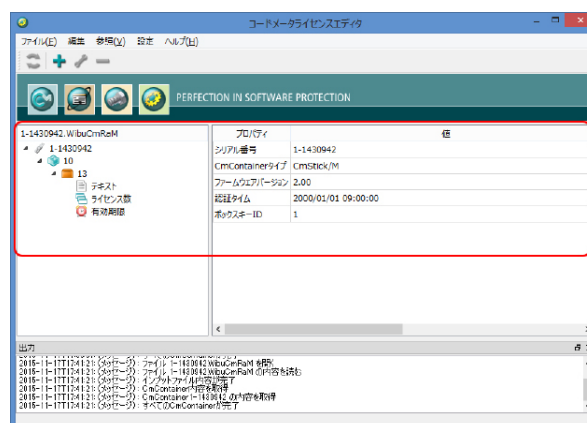
「リモートテキストファイルを選択してください」画面で、ライセンス修正ファイル(.WibuCmRaM)を選択し、「開く」ボタンをクリックします。ライセンス修正ファイルの拡張子の最後は"M"です。

X-XXXXXXX.WibuCmRaM



④ ライセンス修正ファイルの内容を確認

左ペインにライセンス修正ファイルの内容が表示されます。

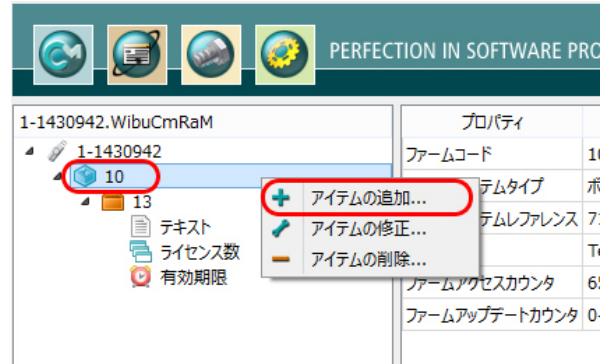


⑤ ライセンス更新ファイルを作成する

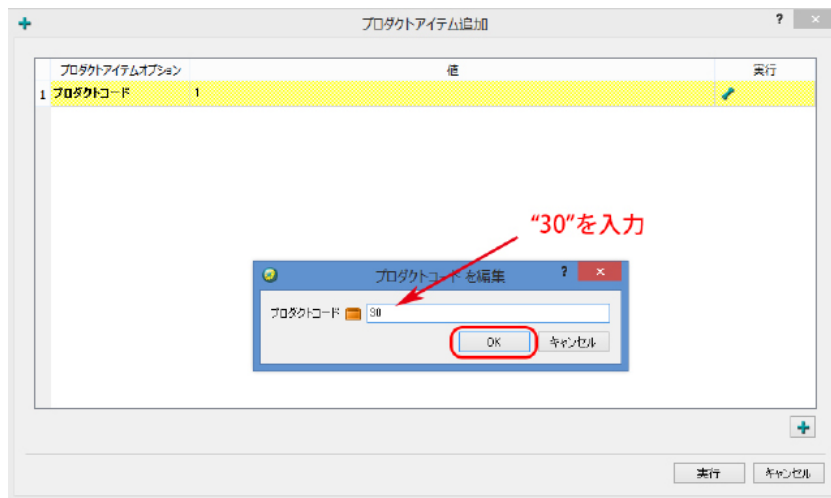
ファームコード(=10)にマウスを合わせ、右クリックで「アイテムの追加...」を選択します。

[NOTE]

ここでは、プロダクトコード=30を新規追加します。プロダクトアイテム=13の内容を修正する場合は、プロダクトコード=13にマウスを合わせ、右クリックで「アイテムの修正...」を選択し、作業を進めます。

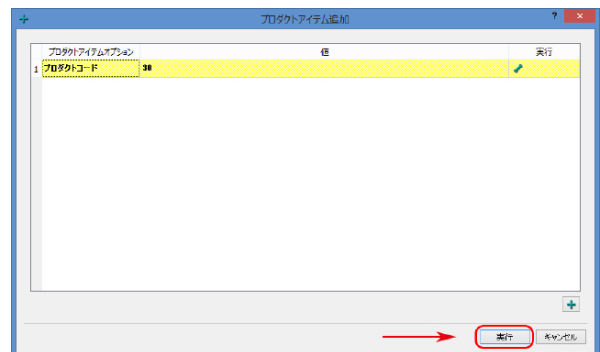


「プロダクトアイテム追加」/「プロダクトコードを編集」画面でプロダクトコード欄に"30"を入力し、「OK」ボタンをクリックします。



「プロダクトアイテム追加」画面で「実行」ボタンをクリックします。ライセンス更新ファイル(.WibuCmRaU)がライセンス修正ファイルと同じフォルダに作成されます。

ライセンス修正ファイルから作成すると、ファイル名左部に(1)が追加されます。ライセンス更新ファイルと同時にあらたなライセンス修正ファイルも作成されます。次にライセンス更新ファイルを作成する場合は、(1)が付いた新しいライセンス修正ファイルを使用します。



\\(C:) > Work > 151117

名前	更新日時	種類	サイズ
1-1430942(1).WibuCmRaM	2015/11/17 17:59	WIBU Control File	2 KB
1-1430942(1).WibuCmRaU	2015/11/17 17:59	WIBU Control File	1 KB
1-1430942.WibuCmRaC	2015/11/17 17:30	WIBUCMRAC フ...	2 KB
1-1430942.WibuCmRaM	2015/11/17 17:31	WIBU Control File	2 KB
1-1430942.WibuCmRaU	2015/11/17 17:31	WIBU Control File	1 KB










⑥ ライセンス更新ファイルをユーザーに送る

作成されたライセンス更新ファイルをユーザーにメール添付などで送ります。ライセンス更新ファイルの拡張子は、".WibuCmRaU"で、ファイル名の左部は変わりません。

x-xxxxxxx(1).WibuCmRaU

[NOTE]

ライセンス修正ファイルからライセンス更新ファイルを作成するたびに、(x)が追加されます。つねに、新しいファイルを使用してください。古いファイルが使用された場合、CmDongleの更新に失敗しますのでご注意ください。

名前	更新日時
 1-1430942(1)(2)(3).WibuCmRaM	2015/11/17 18:20
 1-1430942(1)(2)(3).WibuCmRaU	2015/11/17 18:20
 1-1430942(1)(2).WibuCmRaM	2015/11/17 18:17
 1-1430942(1)(2).WibuCmRaU	2015/11/17 18:17
 1-1430942(1).WibuCmRaM	2015/11/17 17:59
 1-1430942(1).WibuCmRaU	2015/11/17 17:59
 1-1430942.WibuCmRaC	2015/11/17 17:30
 1-1430942.WibuCmRaM	2015/11/17 17:31
 1-1430942.WibuCmRaU	2015/11/17 17:31

9-5. ファームアイテムの追加と修正

CmDongleに新規のファームアイテムを追加したり、既存のファームアイテムの内容を修正します。

新規ファームアイテムの追加

① CmFSB と CmDongle を装着する

まず、コードメータFSB(CmFSB)とコードを登録するCmDongleをPCにそれぞれ装着します。必ず、2つを同時に装着します。

②コードメータライセンスエディタを起動する

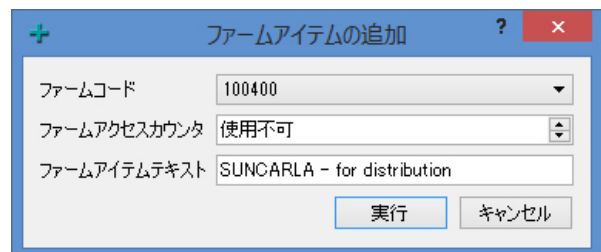
スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

③ 「アイテムの追加 ...」を選択する

新規ファームコードを追加したいCmDongleのシリアル番号を選択し、右マウスで「アイテムの追加...」を選択します。

④ファームアイテムを追加する

「ファームアイテムの追加」画面で、ファームコード、ファームアクセスカウンタ、ファームアイテムテキストを入力します。



ファームコード

新規追加するファームコードを指定します。利用可能なファームコードをリストから選択します。すでに同じファームコードが登録されている場合や、該当するCmFSBが装着されていない場合はファームコードの作成ができません。また、あらかじめ貴社のライセンスファイルCmFirm.wbcがコードメータコントロールセンターにインポートされている必要があります。(Chapter2/「2-3.貴社のライセンスファイルCmFimr.wbcをインポートする」参照)

[NOTE]

ファームコードを作成するためには、CmFSB側にユニットカウンタが1以上存在する必要があります。新規のCmDongleにファームコードを登録するごとにユニットカウンタが1つずつ減ります。ユニットカウンタが0の場合、新規のCmDongleにファームコードを登録することができません。CmFSBのユニットカウンタを増やす(購入する)場合は弊社までご連絡ください。

ファームアクセスカウンタ

ファームアイテムに対する数値を指定します。特別なオプションが設定されるとファームアクセスカウンタは1減ります。デフォルト値は「使用不可」(65535)に設定されており、通常ファームアクセスカウンタは非アクティブの状態になっています。(推奨)この値をプログラムすることができます。値が0になると、ファームアイテムがロックされ使用不可の状態になります。

ファームアイテムテキスト

ファームアイテムの内容をテキスト入力します。

既存のファームアイテムの修正

① CmFSB と CmDongle を装着する

まず、コードメータFSB(CmFSB)とコードを登録するCmDongleをPCにそれぞれ装着します。必ず、2つを同時に装着します。

②コードメータライセンスエディタを起動する

スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

③「アイテムの修正 ...」を選択する

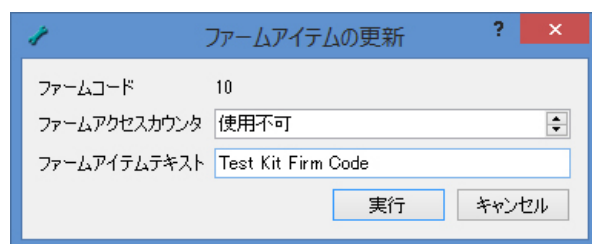
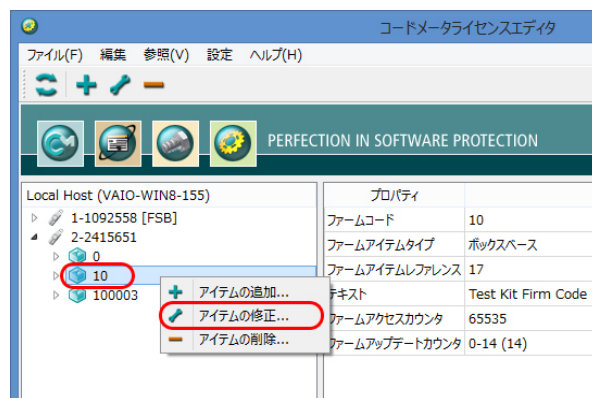
ファームコードを選択し、右マウスで「アイテムの修正...」を選択します。

④ファームアイテムを修正

「ファームアイテムの更新」画面で、ファームアクセスカウンタ、ファームアイテムテキストを修正します。

[NOTE]

「ファームアイテムの更新(修正)」画面では、ファームコード自身の変更はできません。



9-6. プロダクトアイテムの追加と修正

CmDongleに新規のプロダクトアイテムを追加したり、既存のファームアイテムの内容を修正します。

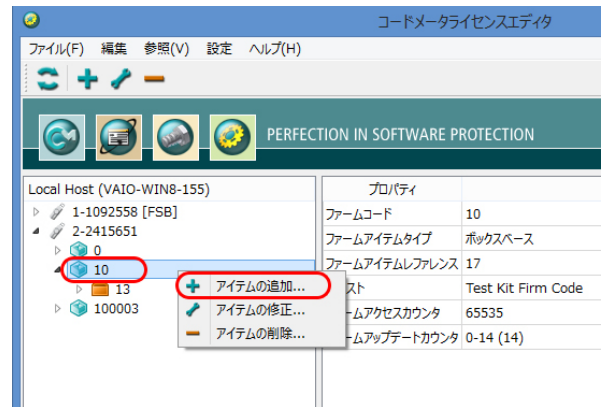
新規プロダクトアイテムの追加

① CmFSB と CmDongle を装着する

まず、コードメータFSB(CmFSB)とコードを登録するCmDongleをPCにそれぞれ装着します。必ず、2つを同時に装着します。

②コードメータライセンスエディタを起動する

スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

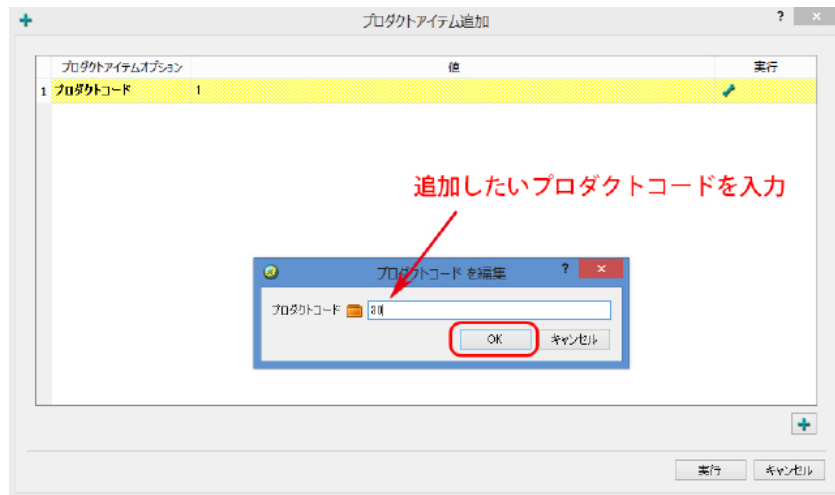


③ 「アイテムの追加 ...」を選択する

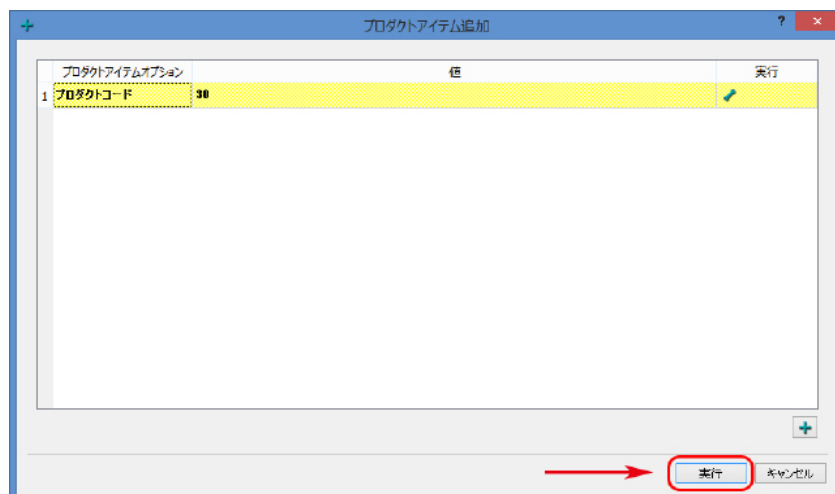
ファームコードを選択し、右マウスで「アイテムの追加...」を選択します。

④プロダクトコードを追加する

「プロダクトアイテム追加」画面で、プロダクトコードを入力し、「OK」ボタンをクリックします。

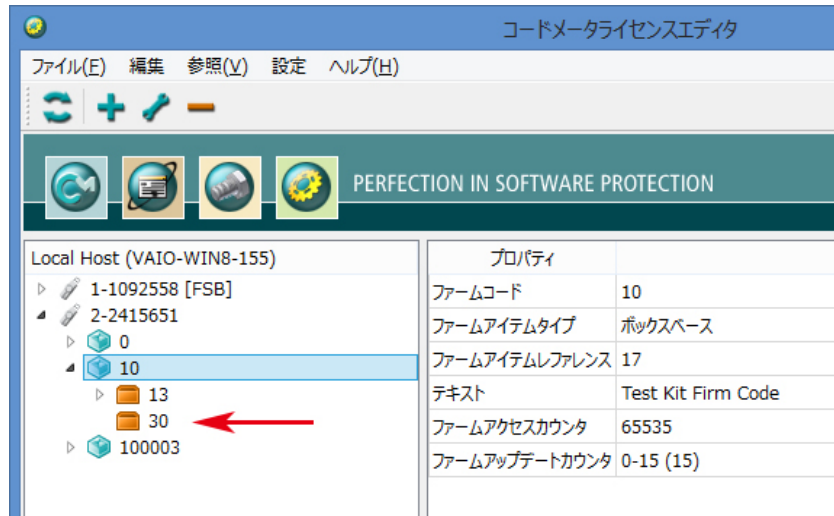


「実行」ボタンをクリックします。



⑤新規のプロダクトコードが追加される

新規のプロダクトコードが追加されます。



既存プロダクトアイテムの修正

① CmFSB と CmDongle を装着する

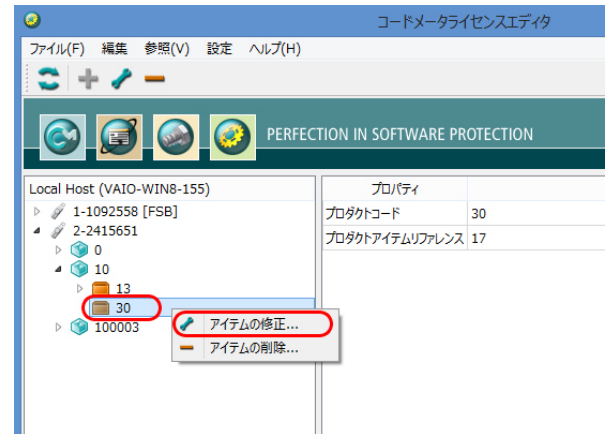
まず、コードメータFSB(CmFSB)とコードを登録するCmDongleをPCにそれぞれ装着します。必ず、2つを同時に装着します。

②コードメータライセンスエディタを起動する

スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

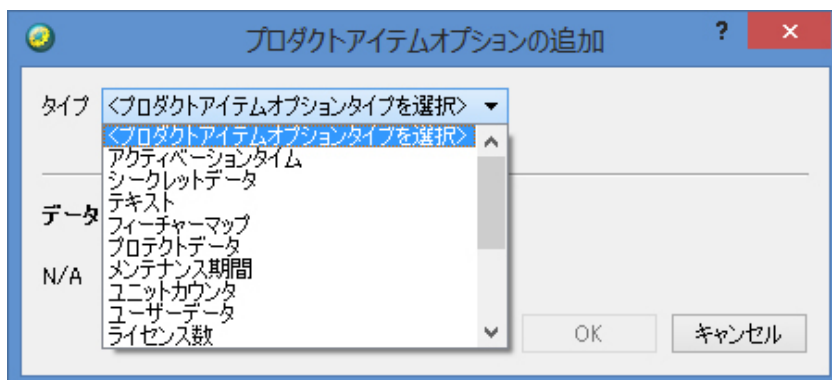
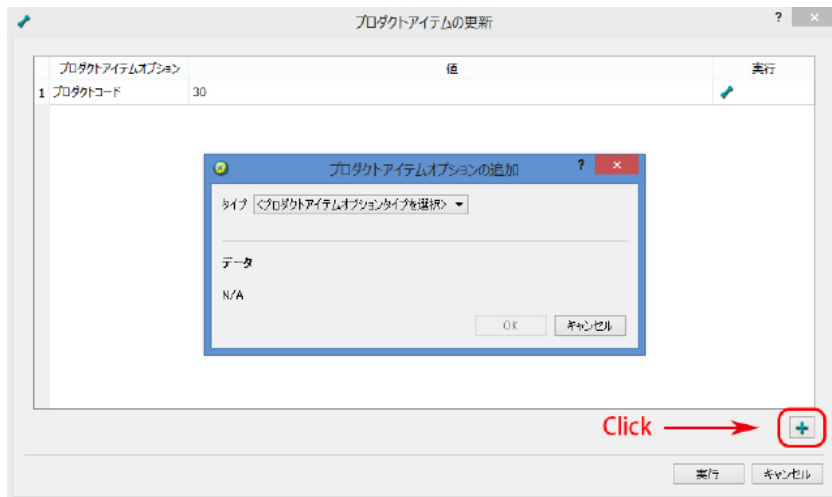
③ 「アイテムの修正 ...」を選択する

プロダクトコードを選択し、右マウスで「アイテムの修正...」を選択します。

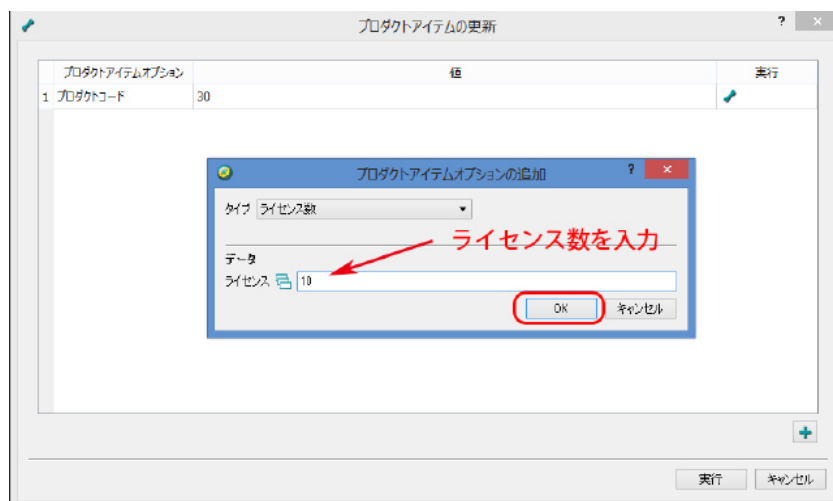


④プロダクトアイテムを修正する

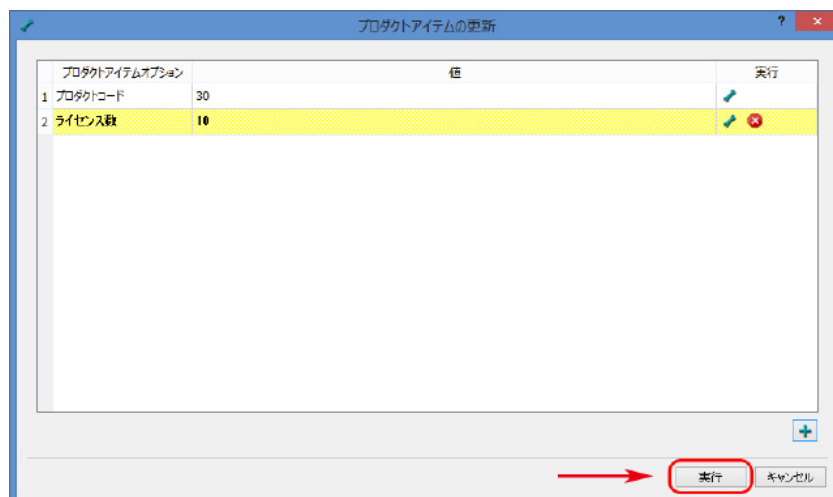
「プロダクトアイテムの更新」の「+」ボタンをクリックし、「プロダクトアイテムオプションの追加」画面で、プロダクトアイテムオプションタイプを選択します。



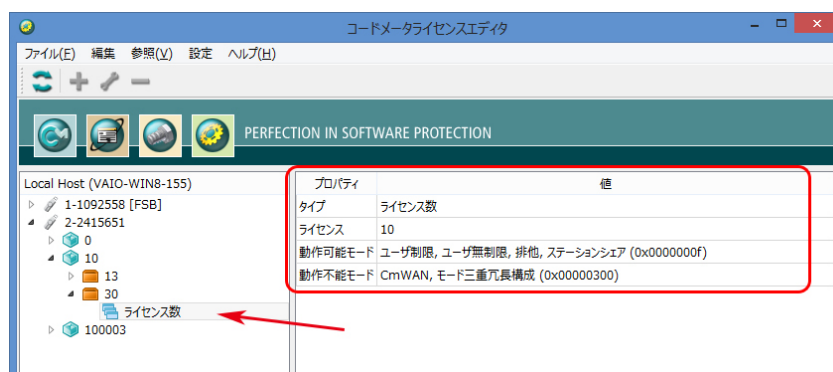
例として、タイプに「ライセンス数」を選択し、ライセンス数として"10"を入力し、「OK」ボタンをクリックします。



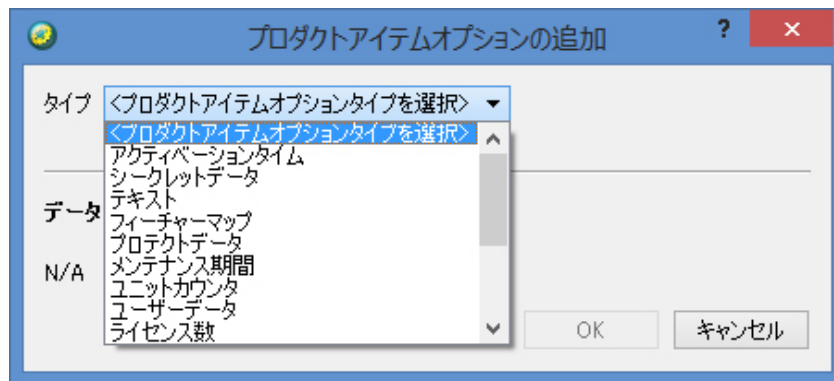
「実行」ボタンをクリックします。



ライセンス数が登録されます。



プロダクトアイテムオプションの種類

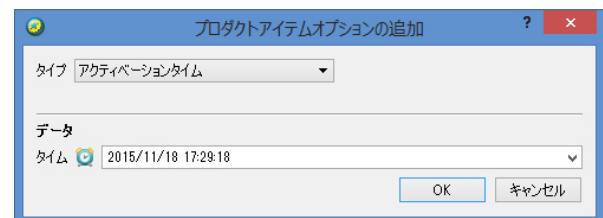


アクティベーションタイム

アクティベーションタイム (ライセンスの使用開始日) を設定します。

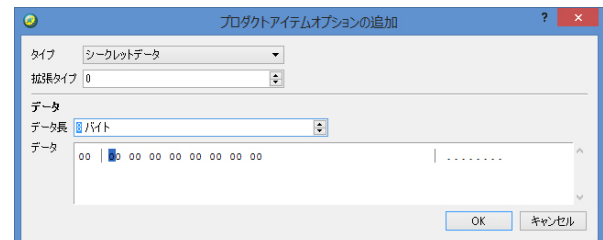
[NOTE]

アクティベーションタイムがCmContainerの認証タイム (Certified Time) よりも進んでいる場合は、インターネットによる認証タイムの更新が必要になります。



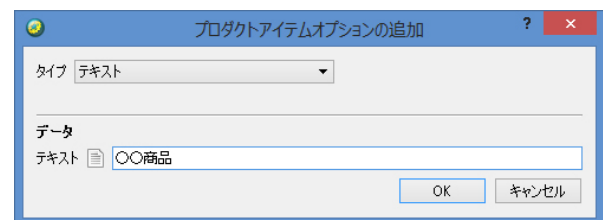
シークレットデータ

セキュアなバイナリーデータを追加します。(不可視)



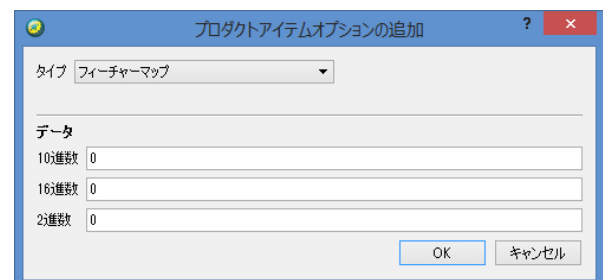
テキスト

プロダクトコードに関するテキストを作成します。



フィーチャーマップ

フィーチャーコードを指定します。フィーチャーコードはバージョン管理に利用します。16進数、10進数、2進数のいずれかで指定できます。



プロテクトデータ

セキュアなバイナリーデータを作成します。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'プロテクトデータ'. Under 'データ' (Data), 'データ長' (Data Length) is set to '1バイト' (1 byte). The 'データ' (Data) field contains '00 | 00 | .'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

メンテナンス期間

メンテナンス期間の「開始」と「終了」を設定します。「開始」「終了」それぞれ右端のボタンをクリックするとカレンダー画面が表示されます。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'メンテナンス期間'. Under 'データ' (Data), '入力モード' (Input Mode) has radio buttons for '日付と時刻' (Date and Time, selected) and '数' (Number). '開始' (Start) is set to '2016/01/01 17:35:10' and '終了' (End) is set to '2017/12/31 17:35:10'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

[NOTE]

「メンテナンス期間」を利用するには、CmDongleのファームウェアバージョン1.18以上が必要になります。

ライセンス数

ネットワーク上またはローカル上で同時アクセスできるライセンス数を設定します。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'ライセンス数'. Under 'データ' (Data), the 'ライセンス' (License) field contains '0'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

ユニットカウンタ

ユニットカウンタを設定します。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'ユニットカウンタ'. Under 'データ' (Data), the 'ユニットカウンタ' (Unit Counter) field contains '0'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

ユーザーデータ

バイナリーデータを作成します。(可視データ)

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'ユーザーデータ'. Under 'データ' (Data), 'データ長' (Data Length) is set to '1バイト' (1 byte). The 'データ' (Data) field contains '00 | 00 | .'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

リンガータイム

リンガータイムを指定します。(秒単位)

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to 'リンガータイム'. Under 'データ' (Data), the 'リンガータイム(秒)' (Ringer Time (seconds)) field contains '0'. 'OK' and 'キャンセル' (Cancel) buttons are at the bottom right.

使用期間

ライセンスが有効な日数を指定します。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '使用期間'. The 'データ' section has '期間' (Period) set to '0 日'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

拡張プロテクトデータ

セキュアなバイナリーデータを作成します。(可視データ)

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '拡張プロテクトデータ'. The '拡張タイプ' (Extension Type) is set to '0'. The 'データ' section has 'データ長' (Data Length) set to '1 バイト' and a text area containing '00 | 0'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

最低バージョン

コードメータの最小ランタイムバージョンを設定します。設定されてコードメータランタイムバージョンより低いバージョンでは起動しません。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '最低バージョン'. The 'データ' section has '最低バージョン' (Minimum Version) set to '5.22'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

有効期限

ライセンスの有効期限を設定します。カレンダーコントロールを使って入力することも可能です。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '有効期限'. The 'データ' section has 'タイム' (Time) set to '2017/12/31 17:51:12'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

非表示データ

セキュアなバイナリーデータを作成します。この非表示データはパスワードを利用してのみ読むことが可能です。

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '非表示データ'. The '拡張タイプ' (Extension Type) is set to '0'. The 'データ' section has 'アクセスコード' (Access Code) set to '0', 'データ長' (Data Length) set to '1 バイト', and a text area containing '00 | 0'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

顧客所有のライセンス情報

プロダクト固有のライセンス情報を作成します

The dialog box is titled 'プロダクトアイテムオプションの追加'. The 'タイプ' (Type) dropdown is set to '顧客所有のライセンス情報'. The 'データ' section has 'テキスト' (Text) set to 'ロンドン商事 Serial Number 5678932'. There are 'OK' and 'キャンセル' (Cancel) buttons at the bottom right.

Chapter 10

CmBoxPgm の使い方

- 10-1. CmBoxPgm について
- 10-2. コード登録の流れ
- 10-3. ファームコード (Firm Code) を登録する
- 10-4. プロダクトコード (Product Code) を登録する
- 10-5. プロダクトコード (Product Code) を削除する
- 10-6. 使用有効期限 (Expiration Time) を登録する
- 10-7. 各パラメータの説明

10-1. CmBoxPgm について

CmBoxPgmは、CmStickなどのコードメータキー(CmDongle)にファームコードやプロダクトコードなどのライセンスコードを登録したり、プログラムの使用期限や回数制限のセキュリティオプションを登録したり、セキュリティデータなどのデータを登録するツールです。作業はコマンドライン環境で行います。コード登録が簡単にできるだけでなく、バッチ処理で一度に大量のコードメータキーを作成することが可能です。

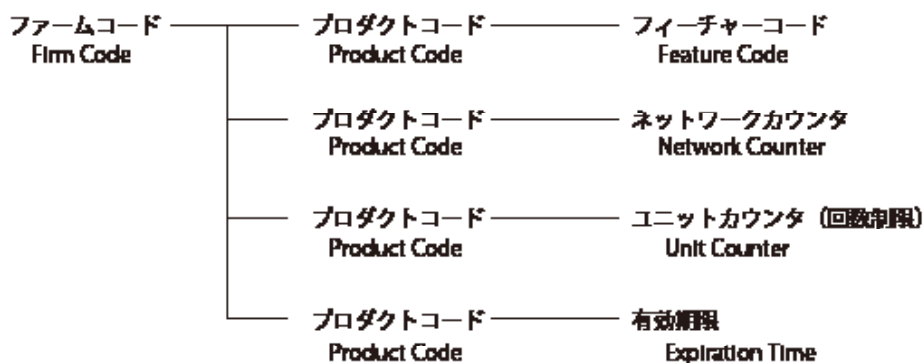
CmBoxPgmを使って、コードメータキーにコードを登録するには、下記の条件が必要です。

- ① コードメータ開発キットがインストールされていること。
- ② 貴社のライセンスファイルCmFirm.wbcがインストールされていること。
- ③ 貴社のコードメータFSB (CmFSB)がPCに装着されていること。

作業は、Windows Vista/7/8/10のコマンドライン環境で行います。

10-2. コード登録の流れ

コードメータのコード登録はファームコード (Firm Code)の登録から始まります。ファームコードの中に各プロダクトコード (Product Code)を登録します。そして、各プロダクトコードの中に、フィーチャーコード (Feature Map)や、ネットワークカウンター、回数制限 (ユニットカウンタ) や有効期限などのセキュリティオプション項目を追加します。



なお、評価用ファームコード=10は、出荷時点で登録されていないので、貴社にて登録する必要があります。(評価用ファームコード=10を使用する場合)

10-3. ファームコード (Firm Code) を登録する

評価用ファームコード=10をCmDongleに登録してみます。

① CmDongle とコードメータ FSB(CmFSB) をそれぞれ任意の USB ポートに挿入します。

CmDongleにファームコードを登録するには、必ずコードメータFSB(CmFSB)が必要になります。

② コマンドプロンプトを開きます。

【スタート】→【すべてのプログラム】→【アクセサリ】で【コマンドプロンプト】を起動します。

以後、コマンドラインでの操作になります。

③ ファームコードを登録する。

ファームコードを登録するためのコマンドラインは以下のとおりです。

CmBoxPgm /F10 /CA ↓ (↓ はEnterキーです)

[NOTE]

デフォルトでPathが有効になっているため、コマンドライン上からCmBoxPgm.exeをそのままタイプすると起動できますが、もしPathが有効になっていない場合は、CD (Change Directory)コマンドで以下のフォルダをカレントフォルダにしてから実行してください。

[例] コマンドプロンプトを開いて、

>CD %Program Files%CodeMeter%DevKit%bin ↓

```

C:\> cd %Program Files%\CodeMeter\DevKit\bin
C:\CodeMeter\DevKit\bin> CmBoxPgm /F10 /CA
Started at 2013-08-15 17:46:34
CmBoxPgm has Version 5.0.1077.500
... Skip Firm Security Box 1-1092558
*** Add Firm Item, CmContainer 2-1289206, FC=10
CmBoxPgm finished at 2013-08-15 17:46:35
  
```

個々のパラメータの意味は以下のとおりです。

/F10 ... /Fに続けて貴社のファームコード (Firm Code) を指定します。

/CA ... ファームコードまたはプロダクトコードを新規追加するパラメータです。

[NOTE]

ファームコードを1つ新規追加すると、コードメータFSB(CmFSB)のファームアイテム99/プロダクトコード(ファームコード)のユニットカウンタが1つ減ります。

[NOTE]

指定したファームコードがすでにCmDongle内に存在するとエラーになります。これは同じファームコードを2重に登録することを防ぐためです。

[NOTE]

各パラメータについては、「10-7. 各パラメータの説明」を参照してください。

10-4. プロダクトコード (Product Code) を登録する

ファームコード (Firm Code) の中にプロダクトコード (Product Code) を登録します。

① CmDongle とコードメータ FSB(CmFSB) をそれぞれ任意の USB ポートに挿入します。

CmDongleにプロダクトコードを登録するには、必ずコードメータFSB(CmFSB)が必要になります。

② コマンドプロンプトを開きます。

【スタート】→【すべてのプログラム】→【アクセサリ】で【コマンドプロンプト】を起動します。

以後、コマンドライン上での操作になります。

③ プロダクトコード (Product Code) を登録する。

ファームコード (Firm Code) = 10 の中に、プロダクトコード (Product Code) = 13 を登録します。コマンドラインから、下記のようにタイプしEnterキーを押します。

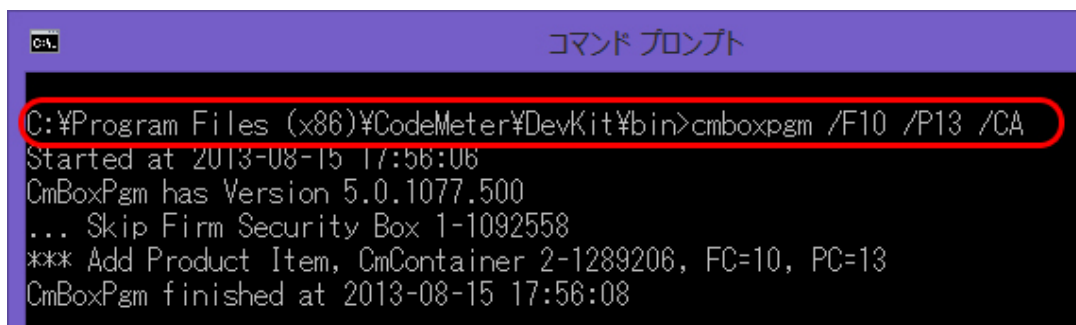
CmBoxPgm /F10 /P13 /CA ↓ (↓ は Enter キーです)

[NOTE]

デフォルトでPathが有効になっているため、コマンドライン上からCmBoxPgm.exeをそのままタイプすると起動できますが、もしPathが有効になっていない場合は、CD (Change Directory) コマンドで以下のフォルダをカレントフォルダにしてから実行してください。

[例] コマンドプロンプトを開いて、

>CD ¥Program Files¥CodeMeter¥DevKit¥bin ↓



```
コマンド プロンプト
C:\Program Files (x86)\CodeMeter\DevKit\bin>cmboxpgm /F10 /P13 /CA
Started at 2013-08-15 17:56:06
CmBoxPgm has Version 5.0.1077.500
... Skip Firm Security Box 1-1092558
*** Add Product Item, CmContainer 2-1289206, FC=10, PC=13
CmBoxPgm finished at 2013-08-15 17:56:08
```

個々のパラメータの意味は以下のとおりです。

/F ... /Fに続けて貴社のファームコード (Firm Code) を指定します。

/P ... /Pに続けて登録するプロダクトコード (Product Code) を指定します。

/CA ... Product Code、Firm Codeを新規に追加するコマンドです。

Webアドミン (WebAdmin) を開いて登録されたコードを確認してください。

10-5. プロダクトコード (Product Code) を削除する

登録されているプロダクトコード (Product Code) を削除します。

コードメータの場合、プロダクトコードやユニットカウンタ、使用有効期限などが登録されている1つのコードをプロダクトアイテム (Product Item) と表現します。また、プロダクトアイテム (Product Item) の集まりをファームアイテム (Firm Item) と表現します。

プロダクトコード (Product Code) を削除すると、プロダクトアイテム (Product Item) が削除されることとなります。

ファームコード (Firm Code) = 10, プロダクトコード (Product Code) = 13のプロダクトアイテムを削除するには、

CmBoxPgm /F10 /P13 /CD ↓ (↓はEnterキーです)

パラメータ"/CD"を使用します。プロダクトコード (Product Code) = 13を持つプロダクトアイテムが削除されます。Webアドミン (WebAdmin) を開いて確認してください。

また、ファームアイテム (Firm Item) を削除する場合は、

CmBoxPgm /F10 /CD ↓ (↓はEnterキーです)

ファームコード (Firm Code) = 10を持つファームアイテムが削除されます。

[NOTE]

Firm Code = 10を持つファームアイテムは、CmBoxPgmコマンドラインから削除できますが、貴社のFirm Codeを持つファームアイテムは削除できません。これは、誤って削除しないように削除防止を施しているためです。(ファームアイテムは削除できませんが、その中のプロダクトアイテムは削除できます。) また、空(カラ)のCmDongleに貴社のファームアイテムを新規登録する場合は、別途ライセンス(ユニットカウンタ)を弊社(サンカーラ)より購入する必要があります。

10-6. 使用有効期限 (Expiration Time) を登録する

Firm Code = 10、Product Code = 13、使用有効期限(Expiration Time) = 2015年12月31日を新規登録する場合：

CmBoxPgm /F10 /P13 /PETA2015Dec31 /CA ↓ (↓はEnterキーです)

パラメータ"/PETA"を使用します。

月(Month)は、

Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

で指定します。2016年4月1日は、2016Apr1になります。

また、すでにFirm Code = 10、Product Code =13が登録されている既存のプロダクトアイテムに対して、使用有効期限(Expiration Time)を追加する場合は、パラメータ"/CA"の代わりに"/CU"を使用します。(A = Add, U = Update)

CmBoxPgm /F10 /P13 /PETA2015Dec31 /CU ↓ (↓はEnterキーです)

既存のプロダクトアイテムに使用有効期限=2015年12月31日が追加されます。

また、使用有効期限を更新する場合は、

CmBoxPgm /F10 /P13 /PETA2016Aug31 /CU ↓ (↓はEnterキーです)

2015年12月31日が2016年8月31日に更新されます。

既存のプロダクトアイテムから使用有効期限を削除する場合は、

CmBoxPgm /F10 /P13 /PETA /CD ↓ (↓はEnterキーです)

パラメータ"/CD"を使用します。使用有効期限のオプション項目だけが削除されます。

(D = Delete)

[NOTE]

同じファームアイテム(Firm Item)の中に、同じプロダクトコード(Product Code)が2つ以上存在している場合、最初に見つけたプロダクトアイテムの使用有効期限を編集しますのでご注意ください。暗号化されたプログラムも最初に見つけたプロダクトアイテムをもとに動作しますので、CmDongleの中に同じプロダクトコードはできるだけ複数混在させないようにしてください。

10-7. 各パラメータの説明

各パラメータは小文字での使用も可能です。

(例) /CA = /ca

また、"/"の代わりに "-" を使用することも可能です。

(例) /CA = -CA

[基本的なパラメータ]

/CA

新しいエントリをCmDongleに追加する。(A=Add)

/CAU

既存のエントリ(ファームアイテムまたはプロダクトアイテム)を更新する、または指定したエントリが存在しない場合は新規に追加する。(AU=Add/Update)

/CU

既存のエントリ(ファームアイテム、プロダクトアイテム、ファームセキュリティボックス(FSB))を更新する。(U=Update)

/CD

既存のエントリ(ファームアイテム、プロダクトアイテム、プロダクトアイテムオプション)を削除する。(D=Delete)

/CDX

既存の可能なエントリ(ファームアイテム、プロダクトアイテム)を削除する。(DX=Delete if possible)

[CM-BOX オプション]

/QBx

インデックスxで指定したCmDongleを選択する。(xは10進数)

[NOTE]

/QNまたは/QSとの併用はできません。

/QNx[,y][:F]

インデックス範囲xで指定したCmDongleを選択する。yはオプション。yが省略された場合は、インデックスx以上のCmDongleを選択する。デフォルトは、FSBの選択を除く。:Fを指定すると、FSBの選択も行う。

/QS[m-]s

マスクmとシリアル番号でCmDongleを選択する。マスクmはオプション。

* CmDongleを選択しないでCmBoxPgmを実行した場合は、CmBoxPgmはFSB以外の最初に見つけたCmDongleを選択します。

/L

CmDongleの内容をリスト表示する。(L=List)

CmBoxPgm /L

最初に見つけたCmDongle(CmFSBは除く)の内容をリスト表示する。

CmBoxPgm /QB1 /L

インデックス 1 のCmDongle (最初に見つけたCmDongle)の内容をリスト表示する。

CmBoxPgm /QB2 /L

インデックス 2 のCmDongle (2 番目に見つけたCmDongle)の内容をリスト表示する。

CmBoxPgm /QS1-1234 /L

シリアル番号1-1234を持つCmDongleの内容をリスト表示する。

CmBoxPgm /QN2,4:F /L

インデックス範囲2から4のCmDongle (2 番目から 4 番目に見つけたCmDongle, 計 3 個)の内容をリスト表示する。CmFSBが含まれている場合もリスト表示する。

/R

Firm Itemの内容を削除する。ただし、CmFSBは除く。

CmBoxPgm /QN2,4 /R

インデックス範囲2から4のCmDongle (2 番目から 4 番目に見つけたCmDongle, 計 3 個)の内容を削除する。ただし、CmFSBの場合は除く。(削除しない)

[Firm Item オプション]

/F

ファームコード(Firm Code)を設定。

/FACx

ファームアクセスカウンター (Access Counter)を設定。xは0 - 0xffffまでの符号なし整数 (unsigned integer)を使用。デフォルト値は0xffff。

/FPT

Firm Precise Timeを設定。

/FTx

ファームアイテムテキスト(Firm Item Text)を設定。xにはテキスト文字を指定。テキスト文字はダブルクォーテーションで囲む。

/FUCx

ファームアップデートカウンターを設定。xは、符号なし整数 (unsigned integer)で、デフォルト値は0。

/CA

新規のファームアイテムを作成

/CU

既存のファームアイテムを更新

/CD

ファームアイテムを削除

(例)

CmBoxPgm /QN1,4 /F206 /FPTR0 /FT¥"The Firm Item Text¥" /CA

インデックス範囲1から4のCmDongle (1番目から 4番目に見つけたCmDongle, 計4個、ただし、CmFSBは除く)にファームコード=206を新規追加し、Firm Precise Timeをカレントに設定し、ファームアイテムテキストに"The Firm Item Text"を登録する。

CmBoxPgm /QB2 /F206 /FPTR1 /FUC42 /FAC0x1066 /CU

2番目に見つけたファームコード=206を持つCmDongleのFirm Precise Timeを1日加算し、ファームアップデートカウンターを42に設定し、ファームアクセスカウンターを0x1066に設定する。

CmBoxPgm /QS1-1234 /L /F206 /FPTA2015Nov04, 13:14:15PST /CU /L

シリアル番号1-1234を持つCmDongleの内容をリスト表示し、ファームコード=206を持つファームアイテムに対して、Firm Precise Timeを2015年11月4日 13時14分15秒 (太平洋標準時間:PST=Pacific Standard Time)に設定し、そのCmDongle内容をリスト表示する。

CmBoxPgm /F206 /CD

Firm Code = 206を持つFirm Itemを削除する。

[Product Item オプション]**/Px**

プロダクトコード(Product Code)を設定。xにはプロダクトコードを指定。

/PAT[A|R]x

Activation Time (使用開始期日)を設定。

/PATAの場合は、絶対時刻を指定。(例:/PATA2016Dec31,00:00:00)

/PATRの場合は、相対日数を指定。(例:/PATR30は、今日から30日後を意味する)

/PET[A|R]x

Expiration Time (使用有効期限)を設定。

/PETAの場合は、絶対時刻を指定。(例:/PETA2016Dec31,00:00:00)

/PETRの場合は、相対日数を指定。(例:/PETR30は、今日から30日後を意味する)

/PFMx

Feature Map (フィーチャーマップ)を設定。xは、0 - 0xffffffffの範囲の符号なし整数 (unsigned integer)。

/PHD

Hidden Dataを設定。

/PTx

プロダクトアイテムテキストを設定。

/PNWC

Network License Counter (ネットワーク・ライセンス・カウンター) を設定

/PPD

Protected Data を設定。

/PSD

Secret Data を設定。

/PT¥"<text>¥"

プロダクトアイテムテキストを設定。

/PUC

Unit Counter (ユニットカウンター) を設定

/PD

User Data を設定。

/CA

新規のプロダクトアイテムを作成

/CU

既存のプロダクトアイテムを更新

/CD

プロダクトアイテムを削除

/L

プロダクトアイテムをリスト表示する。

(例)

CmBoxPgm /QS1-1234 /F206 /P2001 /PETR30 /PUCA1492 /PFM0x8000 /CA

シリアル番号1-1234のCmDongleの中で、ファームコード=206を持つファームアイテムに対して、プロダクトコード=2001を追加し、30日の有効期限を設定し、ユニットカウンターを1492にし、フィーチャーマップを0x8000に設定する。

CmBoxPgm /QS1-1234 /F206 /P2001 /PETR335 /PUCA426 /pt¥"A text¥" /CU

シリアル番号1-1234のCmDongleの中で、ファームコード=206、プロダクトコード=2001を持つプロダクトアイテムに対して、有効期限を335日延長し、ユニットカウンタを426にし、プロダクトアイテムテキストに"A text"を登録する。

CmBoxPgm /QS1-1234 /F206 /P2001 /PET /CD

シリアル番号1-1234のCmDongleの中で、ファームコード=206、プロダクトコード=2001を持つプロダクトアイテムに対して、有効期限を削除する。

[Help オプション]

CmBoxPgm /?

CmBoxPgmのコマンドをすべて表示

CmBoxPgm /??

ヘルプオプションとヘルプトピックスに関する情報を表示

(例)

CmBoxPgm /?PFM

"PFM" (Feature Map)のヘルプを表示

CmBoxPgm /?productitem

Product Itemに関するヘルプ情報をすべて表示

Chapter 11

コードメータコントロールセンターの使い方

- 11-1. コードメータコントロールセンターとは
- 11-2. ストラクチャとナビゲーション
- 11-3. メニューバー
- 11-4. 「ライセンス」
- 11-5. 「イベント」
- 11-6. 「ライセンス貸出」
- 11-7. ライセンス貸出・返却の方法
- 11-8. ライセンス貸出の有効期限について

11-1. コードメータコントロールセンターとは

コードメータコントロールセンター (CodeMeter Control Center) とは、ユーザーが CmContainer およびコードメータソフトウェア (コードメータライセンスサーバーなど) をコントロール (管理) するツールです。このツールは、コードメータランタイムキットをインストールすると自動的にインストールされます。

コードメータコントロールセンター (CodeMeter Control Center) には、主に次の管理機能があります。

1. コードメータライセンスサーバー (CodeMeter License Server) の管理
2. ライセンス貸出・返却の管理
3. CmContainer の管理 (ファームウェアの更新、ステータスの変更、パスワードの変更等)
4. Web アドミンへのゲートウェイ
5. リモートアップデートファイルの作成・管理・インポート

コードメータコントロールセンターの起動方法

Windows

[スタート]-[すべてのプログラム]-[CodeMeter]-[CodeMeter Control Center] をクリックする。または、画面右下のアイコン (コードメータコントロールセンター) をクリックする。

CodeMeterCC.exe が起動する。CodeMeterCC.exe は下記フォルダに存在します。

¥Program Files (x86)¥CodeMeter¥Runtime¥bin¥CodeMeterCC.exe

Mac OS

[アプリケーション]-[CodeMeter] - [CodeMeter Control Center] をクリックする。

CodeMeterMacX が起動する。

Linux

[Applications]-[System] - [CodeMeter Control Center] または

[Applications] - [Accessories] - [CodeMeter Control Center]

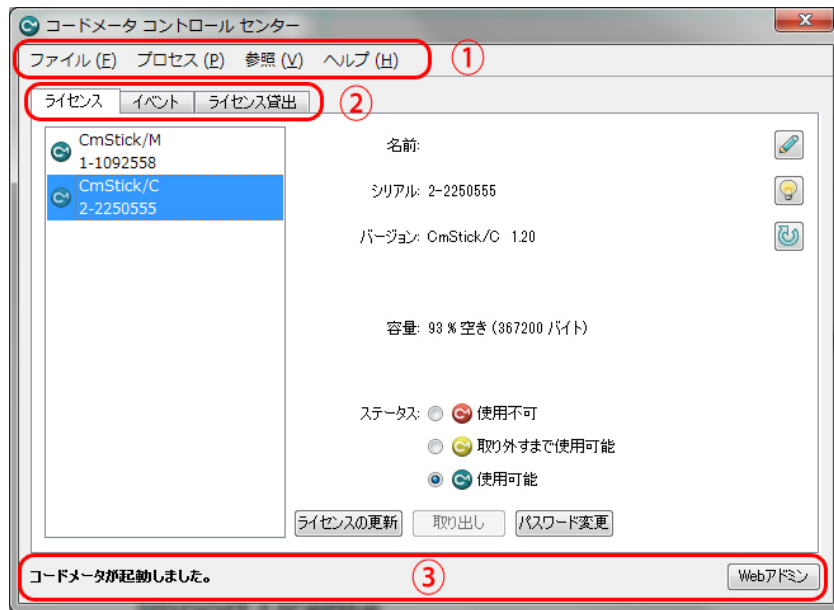
CodeMeterLin が起動する。

Sun Solaris

[/opt/CodeMeter/CodeMeterCC]

CodeMeterSun が起動する。

11-2. ストラクチャとナビゲーション



コードメータコントロールセンターのユーザーインターフェースは3つの部分に分かれます。






- ① メニューバー
- ② タブ領域
- ③ ステータス領域と「Webアドミン」スタートボタン

コードメータコントロールセンターを起動する方法

- 1.Windowsタスクバーにあるコードメータアイコンをクリックする。
- 2.Windowsタスクバーにあるコードメータアイコン上で右クリックし「表示」を選択する。
- 3.[スタート]-[すべてのプログラム]-[CodeMeter]-[CodeMeter Control Center]から起動する。
- 4.¥Program Files (x86)¥CodeMeter¥Runtime¥bin¥CodeMeterCC.exeから直接起動する。

コードメータコントロールセンターのアイコンの色

Windowsタスクバーに表示されるコードメータコントロールセンターのアイコンは色によって意味が異なります。

-  グレー: CmContainerが接続されていない。または、コードメータライセンスサーバーが起動していない。
-  グリーン: 有効なCmContainerが接続されている。
-  ブルー (2重): 複数のCmContainerが接続されている。
-  イエロー: 取り外すまで有効なCmContainerが接続されている。取り外した後は、CmContainerは無効になる
-  レッド: 無効なCmContainerが接続されている。

11-3. メニューバー

「ファイル (F)」メニュー

ライセンスのインポート(I)...

ライセンスのインポートを行います。コードメータライセンス コンテンツのインポート画面で、ライセンスファイル (*.WibuCmRaU; *.wbb; *.wbc) を選択し「インポート」ボタンをクリックします。

[NOTE]

「ライセンスのインポート(I)」メニューからでなく、ライセンスファイルをコードメータコントロールセンター画面上でドラッグ & ドロップしてインポートすることも可能です。

Webアドミン(W)

コードメータWebアドミンを起動します。<CTRL>+W

ログ(L)

コードメータのイベントをログファイルに保存します。<CTRL>+L

ログファイルは、¥Program Files (x86)¥CodeMeter¥Logsに保存されます。(Windowsの場合)

[NOTE]

「ログ(L)」にチェックを入れない場合、ログファイルは作成されませんのでご注意ください。

優先(e)

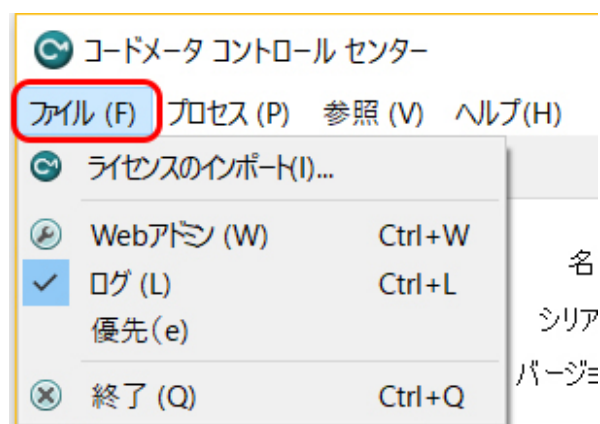
コードメータWebアドミンを起動し、「構成」/「サーバー検索リスト」画面を表示します。

終了(Q)

コードメータコントロールセンターを終了します。<CTRL>+Q

[NOTE]

コードメータコントロールセンターを終了してもコードメータライセンスサーバー(CodeMeter.exe)は終了しません。



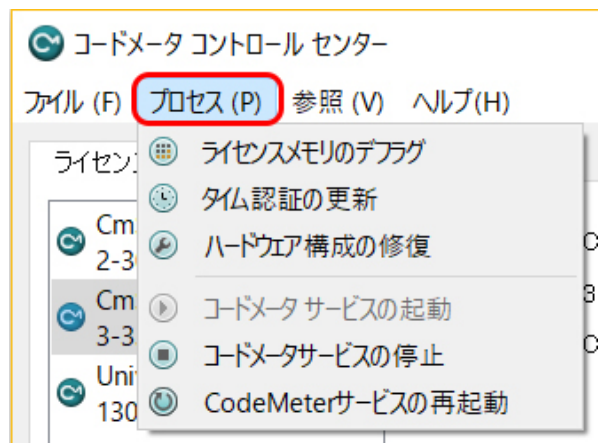
プロセス (P) メニュー

ライセンスメモリのデフラグ

選択したCmContainerのライセンスメモリのデフラグを行います。

タイム認証の更新

接続されているCmContainerのタイム認証更新を行います。接続されているすべてのCmDongleのタイムスタンプが更新されます。



ハードウェア構成の修復

SDカードやCFカードなどのCmDongleのハードウェア構成を修復します。CmCard(SD, CFなど)がコードメータコントロールセンターにリスト表示されない場合に使用します。

コードメータサービスの起動

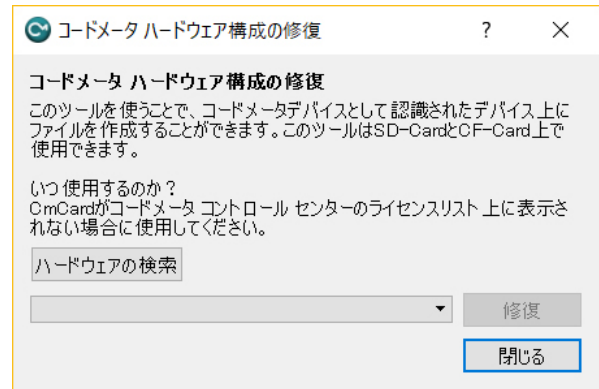
コードメータライセンスサーバーを起動します。

コードメータサービスの停止

コードメータライセンスサーバーを停止します。

CodeMeterサービスの再起動

コードメータライセンスサーバーを再起動します。

**「参照」メニュー****ウィンドウ表示最小化**

コードメータコントロールセンターウィンドウを最小化し、Windowsタスクバーに格納します。
<CTRL+M>

リフレッシュ

接続されているすべてのCmContainerをリフレッシュ表示します。<F5>

ズームイン

「イベント」タブ領域の表示を拡大します。<CTRL>++

ズームアウト

「イベント」タブ領域の表示を縮小します。<CTRL>+-

イベント内容のコピー

「イベント」タブ領域のイベントアクションをクリップボードにコピーします。<CTRL>+C

イベントウィンドウをクリア

「イベント」タブ領域のイベントアクションを削除します。<ALT>+C

接続されている全てのCmContainerを表示

接続されている全てのCmContainerの内容と詳細を「イベント」タブ領域に表示します。<ALT>+S

オープンされている全てのハンドルのリスト

オープンされている全てのハンドルを「イベント」タブ領域に表示します。

利用可能なライセンスエントリをすべて表示

全てのCmContainerライセンスエントリを「イベント」タブ領域に表示します。<ALT>+E

ールセンター

参照 (V) ヘルプ(H)

ウィンドウ表示最小化	Ctrl+M
リフレッシュ	F5
ズームイン	Ctrl++
ズームアウト	Ctrl+-
イベント内容のコピー	Ctrl+C
<input type="checkbox"/> イベントウィンドウをクリア	Alt+C
<input type="checkbox"/> 接続されている全てのCmContainerを表示	Alt+S
<input type="checkbox"/> オープンされている全てのハンドルのリスト	
<input type="checkbox"/> 利用可能なライセンスエントリをすべて表示	Alt+E

「ヘルプ」メニュー

ヘルプ

コードメータオンラインヘルプを開きます。

CmDongleの登録

CmDongleの登録サイト

<https://my.codemeter.com>

にアクセスします。

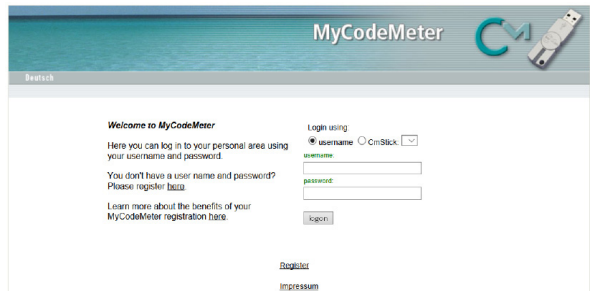
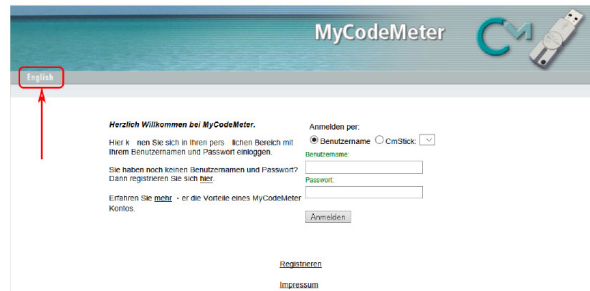
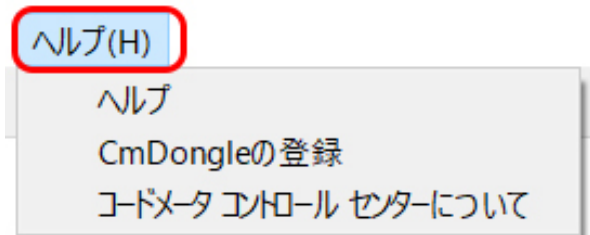
登録画面がブラウザ上で展開されます。左上の"English"ボタンをクリックすると英語表記になります。

CmDongleを登録する利点としては下記があります。

- ライセンスの購入
- CmDongleを紛失したときのロック
- 新しいCmDongleへのライセンス移転
- ソフトウェアの無料アップデート
- CodeMeter製品に関するお知らせ等

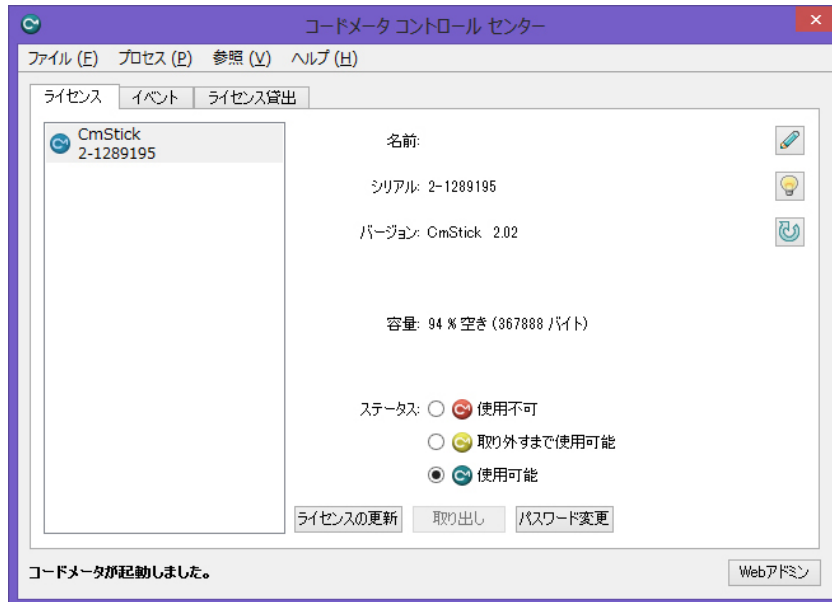
バージョン情報

コードメータコントロールセンターのバージョン情報を表示します。




11-4. 「ライセンス」

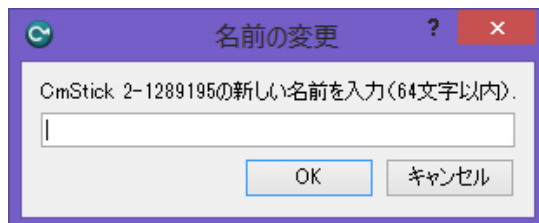
「ライセンス」では、接続されているCmContainer情報やステータス状態を表示したり、ファームウェアの更新などが行えます。また、ライセンス要求ファイルの作成やライセンス更新ファイルのインポート、CmDongle自身のパスワード変更などが行えます。




名前

選択したCmContainerの名前を変更します。

 アイコンをクリックすると名前の変更画面が表示されます。新しい名前を入力してOKボタンをクリックします。



 アイコンをクリックすると選択したCmDongleのLEDが点滅します。CmDongleが複数存在している場合に識別するときに役立ちます。

 アイコンをクリックすると選択したCmDongleのファームウェアをアップデートします。ファームウェアを最新にすることで、動作の安定性を確保することが可能になります。

[NOTE]

ファームウェアをアップデートするには、インターネット接続が必要になります。アイコンをクリックするとWibu-Systems社のファームウェアアップデートサーバーにアクセスします。アップデートには数分かかります。プロセスが完了するまで、CmDongleを取り外さないでください。

容量


選択したCmDongleのコードメータスマートカードチップの(空き)容量をパーセント表示と数値で表示します。


[NOTE]


ここで表示される数値はスマートカードチップの空き容量で、CmDongleのフラッシュメモリ領域の数値ではありません。

ステータス

選択したCmDongleのアクティベーションステータスを表示します。


 接続したCmContainerは使用不可の状態です。CmContainerの中のライセンス情報を利用することはできません。

 接続されている間はCmDongleは使用可能の状態です。一度、PCから取り外すとCmDongleは使用不可の状態になります。

 CmContainerは完全に使用可能の状態です。CmDongleの場合、PCから取り外しても使用可能の状態です。

ステータス:  使用不可
  取り外すまで使用可能
  使用可能

[NOTE]

「取り外すまで使用可能」ステータス  を推奨します。サーバー等に装着しているCmDongleが盗まれた場合や、個人で使用しているCmDongleを紛失した場合に第三者による不正使用を防止するためです。使用不可になったCmDongleをPCに装着すると、パスワードが要求されます。ここで正しいパスワードを入力すると、CmDongleは使用可能の状態(「取り外すまで使用可能」の状態)になります。パスワードが一致しないとCmDongleは使用不可のままです。

ステータスの変更

ステータスを変更するには、次のプロセスを行います。

1. 希望するステータスのラジオボタンをクリックする。
2. 表示されるコードメータ・パスワード画面にてCmDongleパスワードを入力します。
3. OKボタンをクリックします。



ライセンスの更新

選択したCmContainerに対して、ライセンスの更新や新規ライセンスの取得を行います。「ライセンスの更新」ボタンをクリックすると、CmFAS (CodeMeter Field Activation Service) アシスタントが起動します。



取り出し

選択したCmDongleを取り出すときに「取り出し」ボタンをクリックします。
CmDongleがOSからログオフし、PCから安全に取り出すことが可能になります。

パスワード変更

選択したCmDongleのパスワードを変更します。
「コードメータ パスワード変更」画面で、次のプロセスを行います。

1. 「現在のパスワード」に現在のパスワードを入力します。
2. 「新しいパスワード」に新しいパスワードを入力します。
3. 「パスワード再入力」に再度新しいパスワードを再入力します。
4. 「はい」ボタンをクリックするとパスワードが更新されます。

[NOTE]

初期パスワードは"CodeMeter"です。

コードメータ パスワード変更

CodeMeter

CmDongle 2-1289195 のパスワードを変更するには、データ欄にデータを入力して下さい。

マスターパスワードを使用する場合は はい を選択して下さい。

マスターパスワード入力? はい
 いいえ

現在のパスワード:

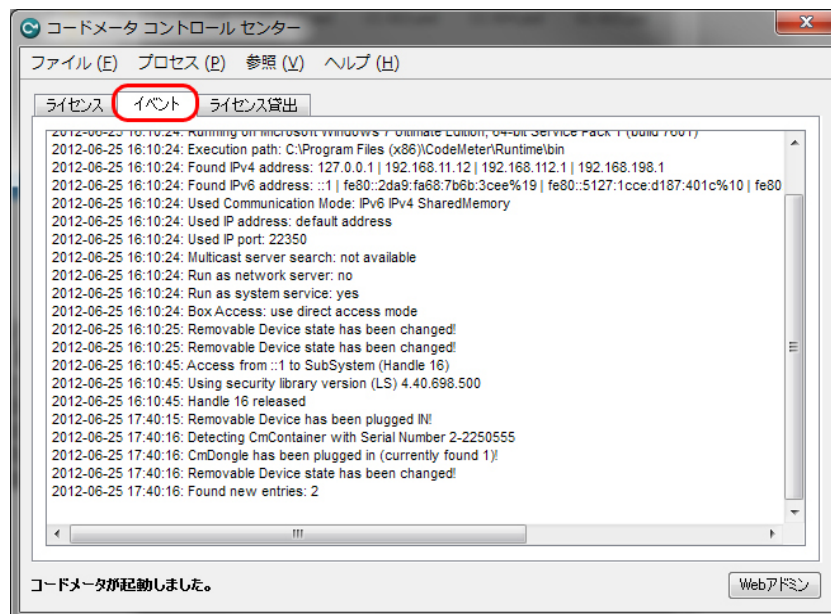
新しいパスワード:

パスワード再入力:

-データ内容が不完全です-

はい キャンセル

11-5. 「イベント」



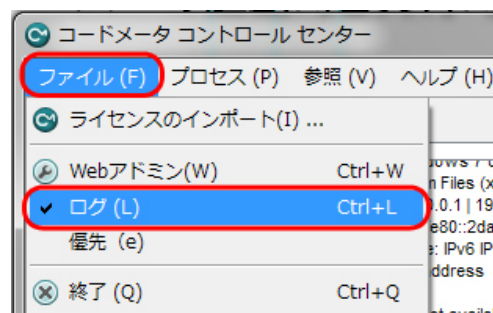
「イベント」タブでは、コードメータライセンスサーバーの起動時と実行時の情報ログや、次の内容を表示します。

- 接続しているCmContainerの数
- CmContainerエントリの数
- ファームアイテムレベルでのライセンスの数

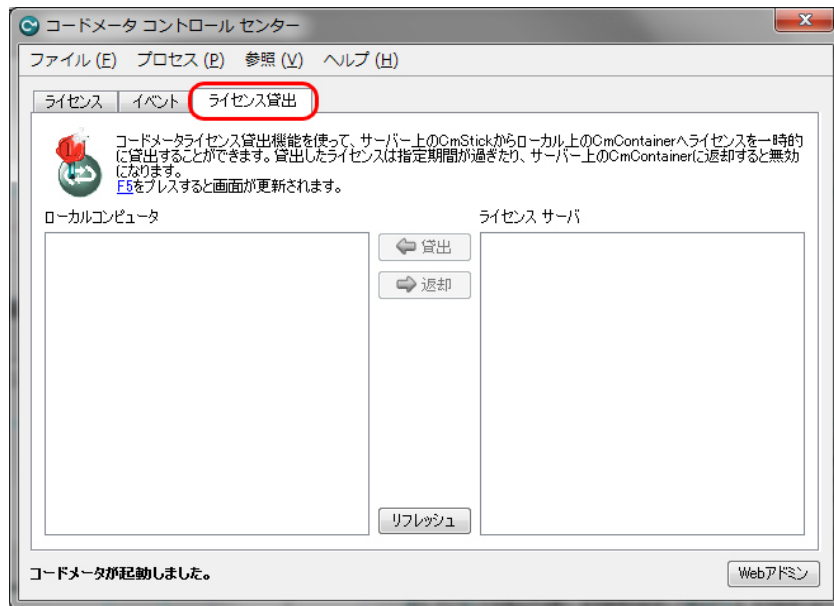
「参照(V)」メニューで、表示する内容を選択できます。

また、ログファイルを作成するには、「ファイル(F)」-「ログ(L)」にチェックを入れる必要があります。デフォルト状態ではチェックがはずれていますのでご注意ください。

ログファイルは、
¥Program Files (x86)¥CodeMeter¥Logsフォルダに
作成されます。(拡張子"log")



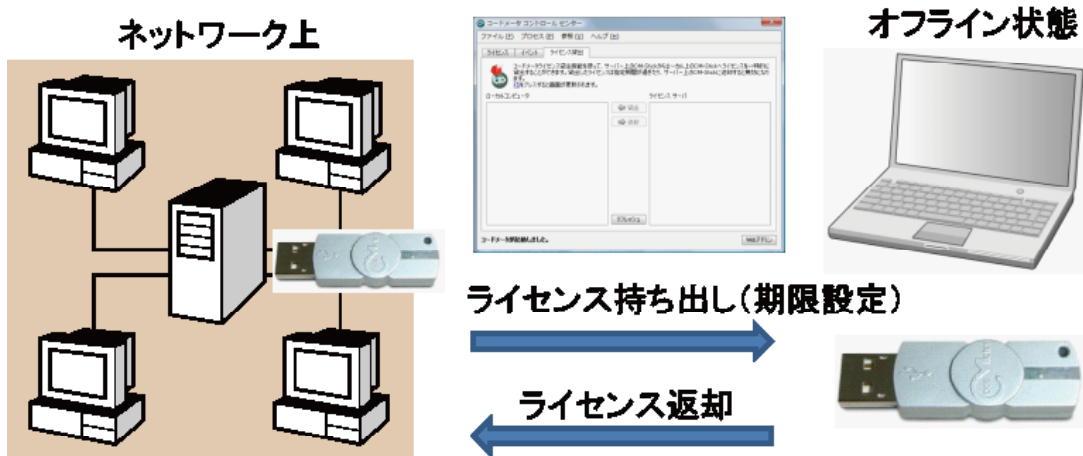
11-6. 「ライセンス貸出」



コードメータライセンス貸出機能を使って、ライセンスサーバー上のCmContainerからローカルコンピュータ上のCmContainerへライセンスを貸出することができます。ネットワークから外れてオフラインの状態でのアプリケーションを使用する場合に非常に便利です。ネットワーク上のフローティングライセンス数はオフライン上のCmContainerに貸出することにより減るため、全体のライセンス数に影響はありません。(全体のライセンス数は変化しない)

また、貸出したライセンスは、ライセンスサーバー上のCmContainerに返却することができます。返却後は、ネットワークライセンス数が復元され、オフライン上のCmContainerは使用できなくなります。(全体のライセンス数は変化しない)

また、ライセンスを貸出する際に、有効期限を設定して貸出すため、有効期間が過ぎるとオフライン上のCmContainerは使用できなくなります。有効期限の前にライセンスを返却しなかった場合、ネットワーク上のライセンス数は自動的に復元します。(全体のライセンス数は変化しない)



11-7. ライセンス貸出・返却の方法

ライセンスの貸出および返却を行うには、次の手順で行います。

1. サーバー用CmDongleをライセンス貸出用に設定する
2. クライアント用CmDongleをライセンス貸出用に設定する
3. コードメータコントロールセンター上でライセンス貸出・返却作業を行う

1. サーバー用 CmDongle をライセンス貸出用に設定する

まず、コードメータサーバーに装着するCmDongleをライセンス貸出ができるように設定する必要があります。サーバー用CmDongleにライセンス貸出設定が行われていないと、貸出作業ができません。

ユーザー先で既に使用しているサーバー用CmDongleにライセンス貸出設定が行われていない場合は、コードメータのリモートアップデート機能を使って、あとからライセンス貸出設定を行うことも可能です。(リモートアップデート機能参照)

①サーバー用 CmDongle とコードメータ FSB(CmFSB) を USB ポートに挿入します。

サーバー用CmDongleにライセンス貸出設定を行うにはコードメータFSB(CmFSB)が必要になります。

②コマンドライン上で設定する。

作業はコマンドライン上でCmBoxPgm.exeを使って行います。また、あらかじめ、サーバー用CmDongleのシリアル番号を確認しておいてください。(パラメータ入力時に必要)

```
CmBoxPgm.exe /qs2-1267216 /f10 /ft:"License Borrowing Server" /cau /p300 /pt:"License Borrowing with CmStick" /plq10 /bls:cm,10,300,0,5,28800,0x12345678 /ca
```

上記コマンドでは、

CmDongle シリアル番号: 2-1267216

Firm Code (ファームコード): 10

Product Code (プロダクトコード): 300

のCmDongleに対して、

貸出ライセンス数: 5

貸出期間:

28800 (minutes) = 480 (hours) = 20 (days)

の設定を行っています。

設定に成功すると、Webアドミン上からライセンス貸出設定の状況が確認できます。

なお、コマンドラインの各パラメータの説明は以下をご参照ください。

The screenshot shows the CodeMeter WebAdmin interface. A red box highlights the 'License Borrowing Server' configuration table. The table has columns for Product Code, Name, Unit Count, Validity, Activation Type, and Network Count. The row for 'License Borrowing with CmStick' shows a Product Code of 300, Unit Count of n/a, Validity of n/a, Activation Type of n/a, and Network Count of 10.

プロダクトコード	名前	ユニットカウンタ	有効期限	アクティベーションタイプ	ネットワークカウンタ
300	License Borrowing with CmStick	n/a	n/a	n/a	10

The screenshot shows the 'Product Item Details' page for CmStick 2-1267216. A red box highlights the 'License Borrowing' row in the table. The table has columns for Product Item Option, Type, Size (Bytes), Persistence, and Value.

プロダクトアイテムオプション	タイプ	サイズ (バイト)	依存性	値
テキスト		62		License Borrowing with CmStick
ライセンス数		4	data, serial, counter	10
ライセンス貸出	132	64	data, serial, counter	CodeMeter 10:300/0 - 5 License(s), expire(s) after 288000 minutes, SID=0x0000000012345678
現在貸出中のライセンス	133	64	serial	0
シークレットデータ	137	16	data, serial, counter	<secret>

[各パラメータの説明]

/qs

CmDongleのシリアル番号を指定する。

/f

ファームコード(Firm Code)を指定する。

/ft

ファームアイテムテキストを指定する。

(例) /ft:"License Borrowing Server"

/cau

CmDongleの既存のエントリを更新する。既存のエントリが存在しない場合は、新規で追加する。

/p

プロダクトコード(Product Code)を指定する。

/pt

プロダクトアイテムテキストを指定する。

(例) /pt:"License Borrowing with CmStick"

/plq

与えるネットワークライセンス数を指定する。

(例) /plq10 (10ネットワークライセンス数を設定する)

/bls

ライセンス貸出用サーバーCmDongleを作成する。

(Syntax)

/bls:[cm|ca],<fc>,<pc>,<fm>,<lqClient>,<duration> [, serverID]

[cm|ca]

cm=CodeMeter, ca=CmActLicense

<fc>

fc=Firm Code (ファームコード)

<pc>

pc=Product Code (プロダクトコード)

<fm>

fm=Feature Map (フィーチャーマップ/フィーチャーコード)

<lqClient>

貸出を許可するクライアント数

ここで設定したクライアント数が実際に貸出可能なクライアント数になります。この数字は当然のことながら/plqで指定したライセンス数を超えて設定することはできません。

(例) /bls:cm,100027,13,0,5,28800,0x12345678

ここでは、貸出を許可するクライアント数を"5"に設定しています。

<duration>

最大貸出期間を設定する。設定する数字は分(minutes)を使用します。

(例) /bls:cm,100027,13,0,5,28800,0x12345678

ここでは、28800(minutes)=480(hours)=20(days)に設定しています。

実際のライセンス貸出期間は、Webアドミンの[構成]/[借用]ページの最大貸出期間で設定された数字が反映されます。Webアドミンの[構成]/[借用]ページ上で何も設定されていない場合は、ここで指定した<duration>の数字が反映されます。ただし、Webアドミン上の"最大借用期間"とCmDongle(サーバー用)の"/bls"の<duration>を両方設定した場合は、"/bls"で設定した<duration>有効期限が優先されます。すなわち、Webアドミン上の"最大借用期間"は、"/bls"の<duration>の範囲内で設定する必要があり、範囲を超えて設定した場合、超過分は無視されますのでご注意ください。

[, serverID]

サーバーIDを8バイトで0x12345678の形式で任意に割り当てます。

このサーバーIDは、ライセンス貸出用クライアントCM-Stickで割り当てるサーバーIDと一致する必要があります。

(例) /bls:cm,100027,13,0,5,28800,0x12345678

ここでは、サーバーIDを0x12345678に設定しています。

/CA

新しいエントリを追加します。

2. クライアント用 CmDongle をライセンス貸出用に設定する

①クライアント用 CmDongle とコードメータ FSB(CmFSB) を USB ポートに挿入します。

クライアント用CmDongleにライセンス貸出設定を行うには、コードメータFSB(CmFSB)が必要になります。

②コマンドライン上で設定する。

作業は、サーバー用CmDongle同様、コマンドライン上でCmBoxPgm.exeを使って行います。また、あらかじめ、クライアント用CmDongleのシリアル番号を確認しておいてください。(パラメータ入力時に必要)

```
CmBoxPgm.exe /qs2-1264652 /f10 /cau /p300 /pt:"Borrowing License Client"  
/blc:cm,10,300,0x12345678 /ca
```

上記コマンドでは、

CmDongle シリアル番号：2-1264652のCmDongleに対してライセンス貸出設定を行っています。

[NOTE]

クライアント用CmDongle設定の場合、パラメータは"/blc"であることに注意してください。

サーバー用CmDongle設定の場合は、"/bls"です。"c"と"s"の違いがあります。

設定に成功すると、Webアドミン上からライセンス貸出設定の状況が確認できます。



The screenshot shows the CodeMeter WebAdmin interface. The main content area displays a table of licenses. A red box highlights the 'Borrow License Client' entry. Below it, there are sections for 'Bundling Articles' and 'SUNCARLA - for distribution'.

プロダクトコード	名前	ユニットカウンター	有効期限	アクティベーションタイム	ネットワークカウンター
300	Borrow License Client	n/a	n/a	[非貸出]	1

プロダクトコード	名前	ユニットカウンター	有効期限	アクティベーションタイム	ネットワークカウンター
1	SecurKey Lite	n/a	n/a	n/a	1

プロダクトコード	名前	ユニットカウンター	有効期限	アクティベーションタイム	ネットワークカウンター
13	-	n/a	n/a	n/a	ローカル
14	-	n/a	n/a	n/a	ローカル

[各パラメータの説明]

サーバー用CmDongleと重複するパラメータの説明は省略します。

/blc

ライセンス貸出用クライアントCmDongleを作成する。

(Syntax)

/blc:[cm|ca],<fc>,<pc> [, serverID]

[cm|ca]

cm=CodeMeter, ca=CmActLicense

<fc>

fc=Firm Code (ファームコード)

<pc>

pc=Product Code (プロダクトコード)

[, serverID]

ライセンス貸出用サーバーCmDongleに割り当てられているサーバーIDを指定します。

8バイトで0x12345678の形式で指定します。

3. コードメータコントロールセンター上でライセンス貸出・返却作業を行う

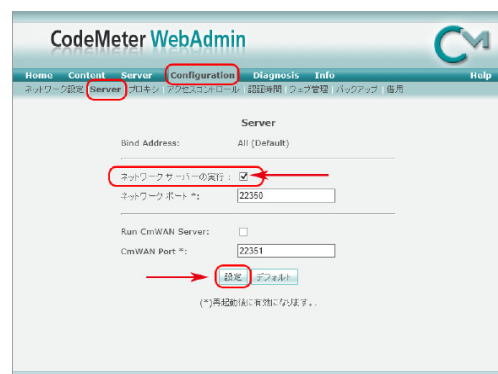
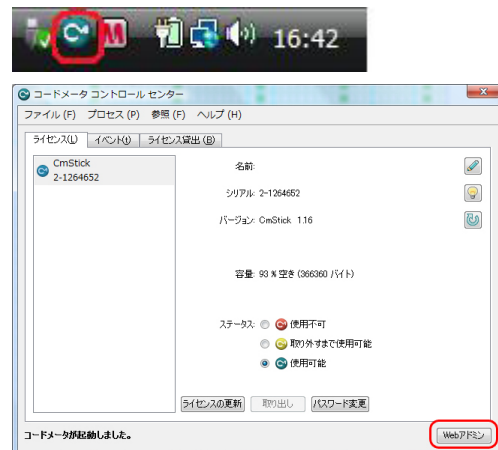
①サーバー用 CmDongle をコードメータサーバーに装着する

ライセンス貸出設定済みのサーバー用CmDongleをコードメータサーバーに装着します。すでに装着されている場合は、「③クライアント用CmDongleを装着する」に進んでください。

② ネットワークサーバーを実行する

タスクバーにあるコードメータアイコンをクリックし、コードメータコントロールセンターを開き、右下の「Webアドミン」ボタンをクリックし、CodeMeter WebAdminを起動します。

「構成」メニューをクリックし、「Server」画面を開き、「ネットワークサーバーの実行」にチェックを入れ、画面下の「設定」ボタンをクリックします。



③ クライアント用 CmDongle を装着する

ライセンス貸出設定が済んでいるクライアント用 CmDongleをクライアントPCに装着します。

④ コードメータコントロールセンターを開く

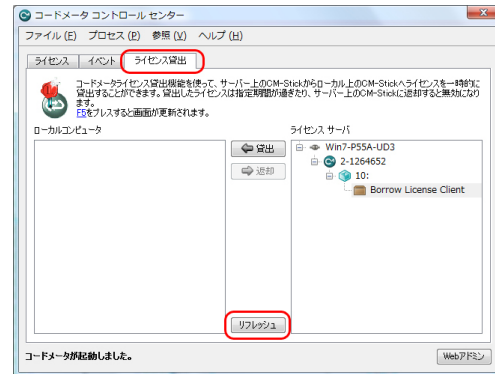
コードメータコントロールセンターを開き、「ライセンス貸出」タブを選択し、「リフレッシュ」ボタンをクリックします。しばらくすると、「ライセンスサーバー」ウィンドウに貸出可能なライセンスが表示されます。

[NOTE]

「リフレッシュ」ボタンの代わりに"F5"をプレスすることも可能です。また、ネットワーク状況により、「ライセンスサーバー」ウィンドウにライセンスが表示されるまで時間がかかる場合があります。

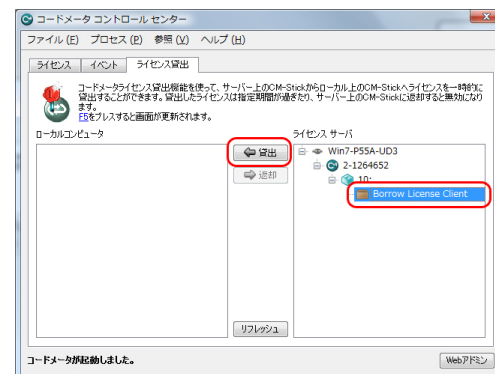
[NOTE]

ライセンス貸出を行うには、ローカルPCはネットワークに接続している必要があります。

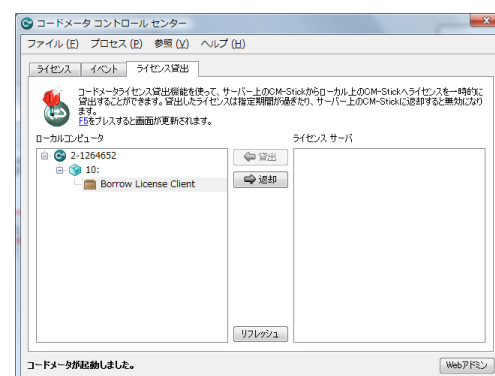


⑤ 「貸出」 ボタンをクリックする

"Borrow License Client"を選択し、「貸出」ボタンをクリックします。



貸出に成功すると、「ローカルコンピュータ」ウィンドウにライセンスが表示されます。これでライセンス貸出は完了です。

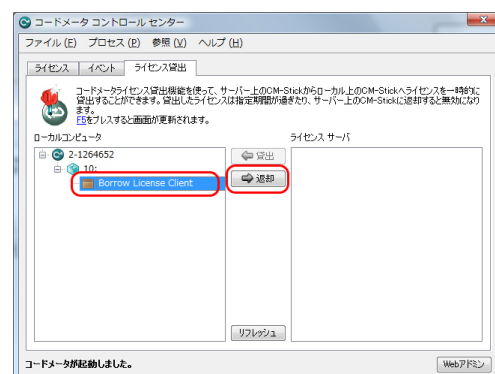


⑥ ライセンスを「返却」する場合

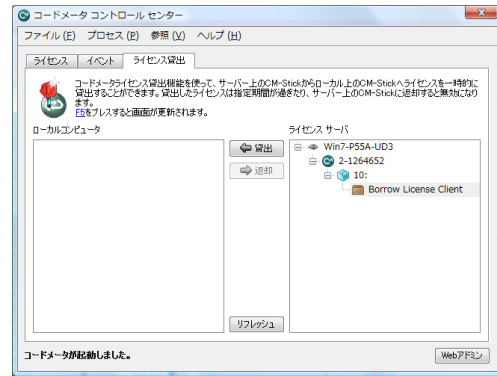
クライアント用CmDongleをローカルPCに装着後、「リフレッシュ」ボタンをクリックして「ローカルコンピュータ」ウィンドウにライセンスを表示させます。「Borrow License Client」を選択し、「返却」ボタンをクリックすると、サーバーCmDongleにライセンスが返却されます。

[NOTE]

ライセンス返却を行うには、ローカルPCはネットワークに接続している必要があります。



返却に成功すると、"ライセンスサーバー"ウィンドウにライセンスが表示されます。これでライセンス返却は完了です。



Chapter 12

Web アドミン (WebAdmin) の使い方

- 12-1. コードメータ Web アドミン (WebAdmin) について
- 12-2. ホーム
- 12-3. 内容
- 12-4. 構成
- 12-5. 診断
- 12-6. 情報

12-1. コードメータ Web アドミン (WebAdmin) について

Webアドミン(WebAdmin)は、PCに接続されているCmContainerの内容を表示したり、コードメータライセンスサーバーの設定・管理を行うツールです。コードメータランタイムキットをインストールすると自動的にPCにインストールされます。Webアドミン(WebAdmin)で行える内容は以下の通りです。

1. ホスト、CmContainerのステータス情報取得:
2. ネットワークサーバー、プロキシ設定、アクセスプロテクション、リモートアクセス、タイムサーバー、バックアップの設定:
3. ライセンス情報の表示:
4. ネットワークライセンスの管理
5. ログ取得
6. バックアップ/リストア

基本

TCP/IP ベース:

コードメータWebAdminと接続されたCmContainerの通信は、ブラウザベースで、ネットワークコンポーネントを利用します。ネットワークプロトコルはTCP/IP、ローカルホスト(localhost)127.0.0.1

ファイアウォール設定:

ファイアウォールがWebAdminとCmContainerの通信をブロックしないように設定してください。コードメータライセンスサーバーは、ローカルPCやネットワークとの通信に特定IPポート(22350)を使用します。この22350は、IANA (Internet Assigned Numbers Authority)においてコードメータ(WIBU-SYSTEMS)通信として登録されています。22350ポートをブロックしないようにファイアウォールを設定してください。

コミュニケーションモード:

コードメータランタイムバージョン: 4.40以前 (4.40は含まず)

- '1' TCP/IP (デフォルト)
- '2' Shared Memory (共有メモリー)

コードメータライタイムバージョン: 4.40以降

- '1' Platform-specific (デフォルト)
 - Windows: IPv6, IPv4; Shared Memory
 - Linux/Mac: IPv6, IPv4
 - WinCE: IPv4, Shared Memory

'2' Shared Memory

'4' IPv4

'8' IPv6

コミュニケーションモードは下記レジストリ内のApiCommunicationModeで設定します。特別の理由がない限り、デフォルト設定での利用をお奨め致します。

Windows: HKLM/SOFTWARE/WIBU-SYSTEMS/CodeMeter/Server/CurrentVersion

Mac OS: /Library/Preferences/com.wibu.CodeMeter.Server.ini

Linux: /etc/wibu/CodeMeter/Server.ini

コードメータ WebAdmin を起動する

コードメータWebAdminは、標準インターネットブラウザで表示されるWebベースのツールです。次の起動方法があります。

Windows:

- ・タスクバーにあるコードメータアイコンを右クリックし、「WebAdmin(W)」を選択する。
- ・コードメータコントロールセンターから「WebAdmin」ボタンをクリックする。
- ・インターネットブラウザから、URLを直接入力する。
http://localhost:22350 または http://127.0.0.1:22350

Mac OS / Linux:

- ・タスクバーにあるコードメータを右クリックし、「WebAdmin(W)」を選択する。
- ・コードメータコントロールセンターから「WebAdmin」ボタンをクリックする。
- ・インターネットブラウザから、URLを直接入力する。
http://localhost:22350 または http://127.0.0.1:22350

もし、コードメータWebAdminが起動しない場合、次の確認を行ってください。

1. 使用しているブラウザが「オフライン」モードになっていないか。
2. 使用しているブラウザがJavaScriptをサポートしているか。

[NOTE]

コードメータWebAdminを利用するにはJavaScriptを有効にする必要があります。

3. ブラウザから直接下記URLアドレスを入力して起動します。
http://localhost:22350 または http://127.0.0.1:22350

12-2. ダッシュボード

全体的な情報

「ダッシュボード」ページでは、PC、コードメータライセンスサーバー、コードメータWebアドミンの全体のステータス情報を表示します。



現在のサーバ:

コードメータライセンスサーバーが起動しているPC名が表示されます。

IP-Address:

現在のサーバのIPアドレスが表示されます。

OS:

使用中のOSが表示されます。

サーバスタートアップ:

コードメータライセンスサーバーの起動時刻を表示します。

サーババージョン:

使用中のコードメータライセンスサーバーのバージョンを表示します。

ランタイムバージョン:

使用中のコードメータランタイムバージョンを表示します。

12-3. コンテナ

「コンテナ」-「すべてのコンテナ」ページでは、接続されているCmContainerをリスト表示します。



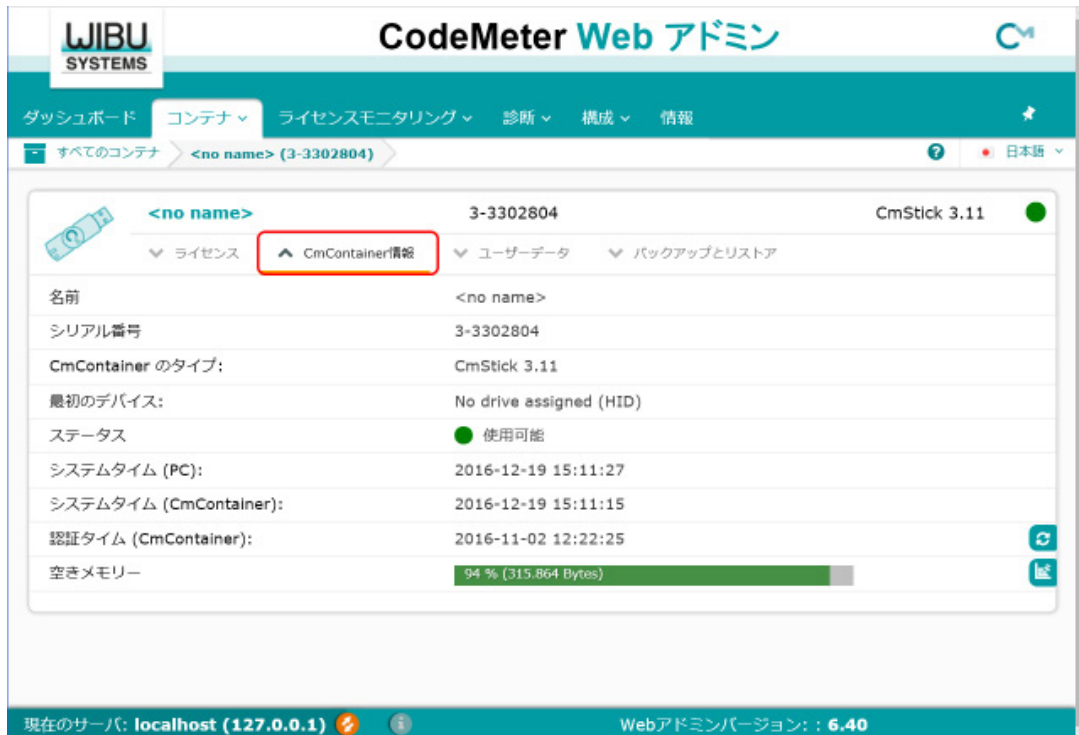
ライセンス:

「ライセンス」をクリックすると、CmContainer内のライセンス情報が表示されます。



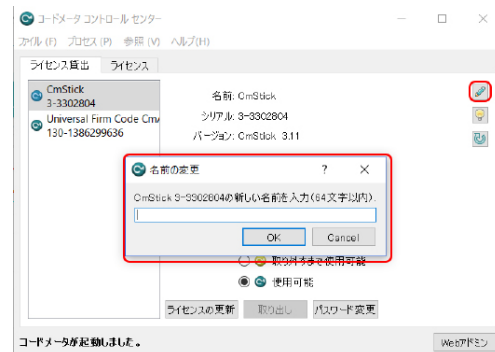
CmContainer情報:

「CmContainer情報」をクリックすると、CmContainer内の基本情報が表示されます。



名前:

CmContainerの名前が表示されます。デフォルトは<no name>ですが、コードメータコントロールセンター上で名前を変更することができます。



シリアル番号:

CmContainerのシリアル番号が表示されます。

CmContainerのタイプ:

CmContainerのタイプとファームウェアのバージョンが表示されます。

最初のデバイス:

CmContainerのデバイス情報が表示されます。デフォルトでは、No drive assigned (HID)が表示されます。(シリアル番号3以降) No drive assigned とはHuman Interface Device (=HID)のことで、CmContainerが外部記憶装置としてでなく、キーボードやマウスのようにHuman Interface Deviceとして認識されます。(シリアル番号2の場合は手動でHIDに変更することが可能です)

ステータス:

CmContainerの現在のステータスを表示します。

システムタイム (PC)

コードメータライセンスサーバーが起動したときのPCのシステムタイムを表示します。

システムタイム (CmContainer)

CmContainerの内部システムタイム (Box Time)を表示します。

認証タイム (CmContainer)

CmContainerに保存されている認証タイムを表示します。右側の「更新」ボタンをクリックすると、インターネット経由でコードメータの認証タイムサーバーにアクセスし、上記3つのタイムを最新状態にすることができます。

空きメモリー

CmDongleのスマートカードチップの空き容量を表示します。ライセンスなどの追加可能な空き容量を表示します。右側の「デフラグ」ボタンをクリックするとCmDongle内のメモリー領域をデフラグします。

The screenshot shows the CodeMeter Web Admin interface. The main content area displays details for a container named '<no name>' with ID '3-3302804' and type 'CmStick 3.11'. The status is '使用可能' (Available). The table lists the following information:

名前	<no name>
シリアル番号	3-3302804
CmContainer のタイプ:	CmStick 3.11
最初のデバイス:	No drive assigned (HID)
ステータス	● 使用可能
システムタイム (PC):	2016-12-19 15:11:27
システムタイム (CmContainer):	2016-12-19 15:11:15
認証タイム (CmContainer):	2016-11-02 12:22:25
空きメモリー	94 % (315.864 Bytes)

On the right side of the table, there are two buttons: a refresh button (circular arrow icon) and a defragment button (trash can icon). Red arrows point to these buttons with the following labels:

- タイムを最新状態にする (Update time to latest state)
- CmDongleのメモリー領域をデフラグする (Defragment CmDongle memory area)

ユーザーデータ:

各プロダクトコードに割り当てられた「ユーザーデータ」を表示します。

The screenshot shows the CodeMeter Web Admin interface. The top navigation bar includes 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The current page is 'すべてのコンテナ' for container '<no name> (3-3302804)'. The main content area shows the 'ユーザーデータ' tab selected, displaying a table of product codes and their associated data.

プロダクトコード	名前	ユニットカウンタ	有効期限	ネットワークカウンタ	フィーチャーマップ
0	-	n/a	n/a	1	n/a
50	Field Update	n/a	n/a	1	n/a
60	PKI	n/a	n/a	1	n/a
1000	-	n/a	n/a	1	n/a

現在のサーバ: localhost (127.0.0.1) WebAdminバージョン: 6.40

バックアップとリストア:

CmContainerの「バックアップ」と「リストア」を行います。

The screenshot shows the CodeMeter Web Admin interface with the 'バックアップとリストア' tab selected. The page provides instructions for backing up and restoring license data.

バックアップ
すべてのライセンスデータをバックアップファイルに書きます:
[バックアップ実行](#) 最後のバックアップ: 2016-12-19 11:48:14

リストア
パーソナルライセンスデータ (CMパスワードマネージャを含む) をリストアする:
 [参照...](#)
[リストア](#) バックアップパス名: C:\ProgramData\CodeMeter\Backup

For information how to restore the license data inside a Firm Code not equal 0, contact our [サポート](#).

現在のサーバ: localhost (127.0.0.1) WebAdminバージョン: 6.40

ライセンスモニタリング

「すべてのライセンス」をクリックすると現在使用可能なライセンスおよび使用状況を表示します。

The screenshot shows the CodeMeter Web Admin interface. The top navigation bar includes 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The 'ライセンスモニタリング' menu is highlighted in red. Below the navigation bar, the 'すべてのライセンス' button is highlighted in red. The main content area shows '利用可能なライセンス' with a sub-section for 'Test Kit Firm Code'.

プロダクトコード	名前	フィーチャーマップ	ネットワークカウンター	使用中	利用可能
13	-	-	30	0	30

Below this, there is a section for '6000153 SUNCARLA Corporation' with a message: '有効なプロダクトアイテムがありません'.

At the bottom, it says '最後に更新された情報 2016-12-19 16:46:09' and 'Webアドミンバージョン: 6.40'.

「セッション」をクリックすると、現在使用中のセッション情報を表示します。

The screenshot shows the CodeMeter Web Admin interface. The top navigation bar includes 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The 'ライセンスモニタリング' menu is highlighted in red. Below the navigation bar, the 'セッション' button is highlighted in red. The main content area shows 'セッション' with a table of active sessions.

クライアント	CmContainer	フォームアイテム	プロダクトアイテム	アクセスモード
LAPTOP-56F14FRE	(3-3302804)	10 : Test Kit Firm Code	13 : -	ユーザー制限
ASUS-WIN10PRO	(3-3302804)	10 : Test Kit Firm Code	13 : -	ユーザー制限

At the bottom, it says '最後に更新された情報 2016-12-19 16:57:59' and 'Webアドミンバージョン: 6.40'.

[NOTE]

Webアドミン上ではライセンス内容を表示するだけで、編集することはできません。

診断 / イベント

「イベント」をクリックするとCmContainerのログが表示されます。



The screenshot displays the CodeMeter Web Admin interface. At the top, the logo for WIBU SYSTEMS and the title "CodeMeter Web アドミン" are visible. A navigation bar contains several menu items: "ダッシュボード", "コンテナ", "ライセンスモニタリング", "診断", "構成", and "情報". The "診断" (Diagnosis) menu is highlighted with a red box, and its sub-menu "イベント" (Events) is also highlighted with a red box. Below the navigation bar, the "イベント" (Events) section is active, showing a list of system logs. The logs are timestamped and include details about logging activation, system startup, network configuration, and CmContainer detection. The status bar at the bottom indicates the URL "http://localhost:22350/diagnosis/events.html", the version "0.0.1", and the Web Admin version "6.40".

```
2016-12-18 18:23:01: Logging activated.
2016-12-18 18:23:01: CodeMeter for Windows (B6.40.2396.500.32.180)
2016-12-18 18:23:01: Running on Microsoft Windows 10 Pro 1607, 64-Bit
2016-12-18 18:23:01: Execution path: C:\Program Files (x86)\CodeMeter\Runtime\bin
2016-12-18 18:23:01: Found IPv4 address: 127.0.0.1 | 192.168.111.1 | 192.168.227.1 | 192.168.56.1
2016-12-18 18:23:01: Found IPv6 address: ::1 | fe80::19bb:f3c0:9272:41a3%13 | fe80::508:f021:6aaa:e15d%10 |
fe80::5849:1:28af:8f0d%15 | fe80::7501:45f2:4ad4:ac13%9 | fe80::c1d1:5alb:224a:1da9%7 | fe80::c858:9552:3913:9aca%12
2016-12-18 18:23:01: Used Communication Mode (Client): IPv6 IPv4 SharedMemory
2016-12-18 18:23:01: Used Communication Mode (Server): IPv6 IPv4 SharedMemory
2016-12-18 18:23:01: Used IP address: default address
2016-12-18 18:23:01: Used IP port: 22350
2016-12-18 18:23:01: Used CmWAN port: 22351
2016-12-18 18:23:01: Multicast server search: not available
2016-12-18 18:23:01: Run as network server: no
2016-12-18 18:23:01: Run as CmWAN server: no
2016-12-18 18:23:01: Run as system service: yes
2016-12-18 18:23:01: Service startup delay: 1:05 minutes
2016-12-18 18:23:01: BOM Access: use direct access mode
2016-12-18 18:23:04: Detecting CmContainer with Serial Number 2-3032655
2016-12-18 18:23:04: A new CmContainer is available (currently found 1)!
2016-12-18 18:23:04: The list of available CmContainers has been updated!
2016-12-18 18:23:05: License file "6000153_8200d6622bdb0744d718ae96f4b6138e4644c325.WibuCmActLic" loaded successfully
2016-12-18 18:23:05: Detecting CmContainer with Serial Number 130-1386295636
2016-12-18 18:23:05: A new CmContainer is available (currently found 1)!
2016-12-18 18:23:05: The list of available CmContainers has been updated!
2016-12-18 18:23:05: Found new entries: 4
2016-12-18 19:35:42: Access from local (IPv6) to Subsystem (Handle 16)
2016-12-18 19:35:42: Handle 16 released
2016-12-18 19:35:42: Some parts of the operating system has changed.
```


12-4. 構成

構成 / 標準 / サーバー検索リスト

サーバー検索リストにコードメータサーバーのIPアドレスを直接設定します。サーバー上にあるCmContainerへのアクセスが不安定な場合に非常に有効です。この設定はクライアント側のPCで行います。

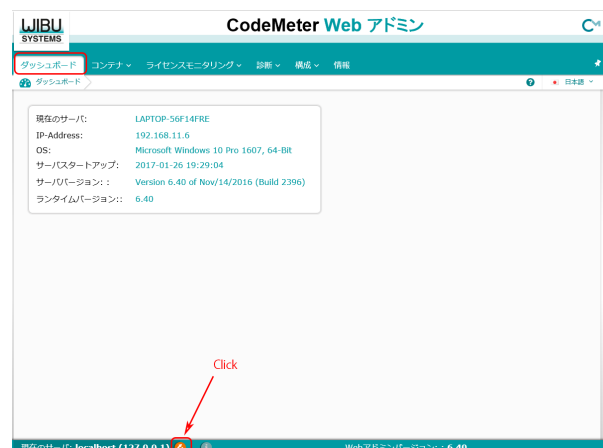


サーバー検索リスト:

コードメータネットワークサーバとアクセスする順番を指定するために、サーバー検索リストを利用します。アクセスするPC名またはIPアドレスを指定することで、コードメータネットワークのパフォーマンスを向上させることができます。

コードメータネットワークサーバーが別のサブネットに存在している場合は、UDPブロードキャストトラブルを回避するために、このサーバー検索リストにIPアドレスを常に指定しておく必要があります。

サーバー接続に成功しているかどうかを調べるには、「ダッシュボード」画面で、「現在のサーバー」ボタンをクリックし、サーバーとしてPCが追加されているかを確認してください。



サーバーのIPアドレスを入力後、「追加」ボタンをクリックします。



「追加」ボタンをクリックすると、サーバー検索リストにサーバーIPアドレスが追加されます。「適用」ボタンをクリックするとIPアドレスが登録され有効になります。



[NOTE]

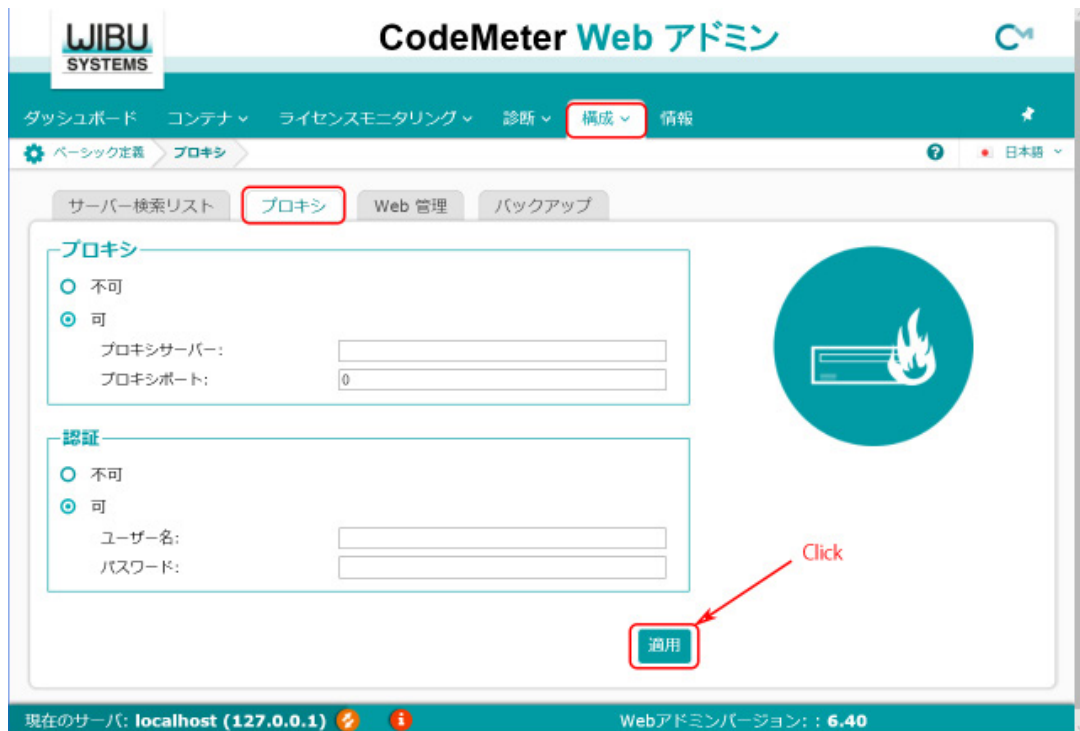
「適用」ボタンをクリックするとIPアドレスが有効になるため、あえてコードメータサーバーを再起動する必要はありません。

[NOTE]

サーバー検索リストの設定は各クライアント側で行います。

構成 / 標準 / プロキシ

コードメータサーバーにプロキシを利用する場合は、プロキシサーバー・プロキシポート、認証情報を設定し、「適用」ボタンをクリックします。



プロキシ:

プロキシサーバーを使用するためのオプションです。プロキシサーバーを使用する場合は、「可」にチェックをいれ、「プロキシサーバー」にはIPアドレスかDNS名を入力し、「プロキシポート」にはプロキシサーバーのポート番号を入力します。

認証:

プロキシサーバー認証を使用可能にします。

プロキシサーバーのプロキシユーザー名とパスワードを入力します。

構成 / 標準 / Web 管理

WebAdminにアクセスするための認証を設定します。

ローカルアクセスのみを許可する場合は、「ローカルアクセスのみ許可」を選択し、「リモート読み許可」のチェックを外して「適用」ボタンをクリックします。クリック後、コードメータサーバーを再起動します。コードメータサーバーを再起動しないと設定が反映されませんのでご注意ください。



WebAdminへのアクセスに認証を設定する場合は、「認証が必要です」を選択して、ユーザー名とパスワードを設定し「適用」ボタンをクリックします。また、サーバー以外のPCからWebAdminにアクセス許可する場合は、「リモート読み許可」にチェックを入れます。設定後、最後にコードメータサーバーを再起動します。(コードメータサーバーを再起動しないと設定が反映されませんのでご注意ください)



構成 / 標準 / バックアップ

Webアドミンのバックアップファイルを作成します。



バックアップパス名:

CmDongleのバックアップファイルが保存される場所を指定します。
デフォルトは、使用中のOSに依存します。

バックアップ間隔:

自動的にバックアップを行う間隔(時間)を指定します。
デフォルトは、24時間間隔です。

バックアップ前に認証時間を更新する:

バックアップを行う前に認証時間の更新を行う場合はチェックをいれます。

「適用」ボタンをクリックして変更を保存します。

「デフォルトを元に戻す」ボタンをクリックするとデフォルトの状態に戻ります。

構成 / サーバー / サーバーアクセス

コードメータサーバーをローカルネットワークサーバーとして認識させる場合は、「ネットワークサーバー」を「可」に設定します。ネットワークポートは22350(デフォルト)のままにします。セグメントを超えてアクセスする場合や、VPN回線でネットワークを構築している場合も、この「ネットワークサーバー」を「可」にすることで対応が可能です。ただし、サーバー検索リスト(クライアントPC側)で、サーバーIPアドレスを登録する必要があります。

また、WANサーバーとして認識させる場合は、「CmWANサーバー」を「可」に設定します。CmWANポートは22351(デフォルト)のままにします。

「ネットワークサーバー」と「CmWANサーバー」を同時に「可」にすることも可能です。設定後は、「適用」ボタンをクリックし、コードメータサーバーを再起動してください。

The screenshot shows the 'CodeMeter Web アドミン' (CodeMeter Web Admin) interface. The top navigation bar includes 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The '構成' (Configuration) menu is highlighted. Below the navigation, there are tabs for 'サーバ定義' and 'サーバアクセス', with 'サーバアクセス' (Server Access) selected. The main content area is divided into two sections: 'ネットワークサーバー' (Network Server) and 'CmWANサーバー' (CmWAN Server). Both sections have radio buttons for '不可' (Disabled) and '可' (Enabled), with '可' selected. The 'ネットワークサーバー' section has a 'ネットワークポート *:' field with the value '22350'. The 'CmWANサーバー' section has a 'CmWANポート *:' field with the value '22351'. At the bottom of the main content area, there are two buttons: '適用' (Apply) and 'デフォルトを元に戻す' (Reset to Default). A note below the buttons states: '(*) 再起動後に有効になります。' (Note: Becomes effective after restart). The footer of the interface shows '現在のサーバ: localhost (127.0.0.1)' and 'Webアドミンバージョン: : 6.40'.

構成 / サーバー / ライセンスアクセス許可

コードメータサーバーにアクセスできるクライアントを制限する場合は、「ライセンスアクセス許可」の「基本」モードを選択し、アクセスを許可するクライアントのIPアドレスを登録します。ここで登録されたIPアドレス以外のクライアントからはコードメータサーバーにアクセスできないため、アプリケーションを起動することができません。



クライアント:

コードメータライセンスサーバーを利用できる権限をもつクライアントを登録します。ここで登録されないクライアントは、コードメータライセンスサーバーのドングルにアクセスできません。

[NOTE]

このリストに何も書かれていない場合は、ネットワーク上のすべてのクライアントがコードメータライセンスサーバーのドングルにアクセスできます。

FSBアクセス許可:

ネットワーク上でFSB (Firm Security Box)を複数ユーザーで共有して使用したい場合にチェックをいれます。チェックをいれると、クライアントからCmContainerへの書込み・編集や、アプリケーションの暗号化処理が可能になります。

「適用」ボタンをクリックして変更を保存します。「デフォルトを元に戻す」ボタンをクリックすると初期設定の状態になります。クライアントリストは空白になり、FSBアクセスは無効になります。

アクセス設定を行った場合、コードメータサービスの再起動が必要になります。ただし、CmContainerを取り外したり、非アクティブにする必要はありません。コードメータサービスの停止や再起動はコードメータコントロールセンターから行えます。

ライセンスアクセス許可設定をさらに詳細に行う場合は「アドバンス」を選択し、「新しいアクセスルールを追加します」ボタンをクリックして設定します。



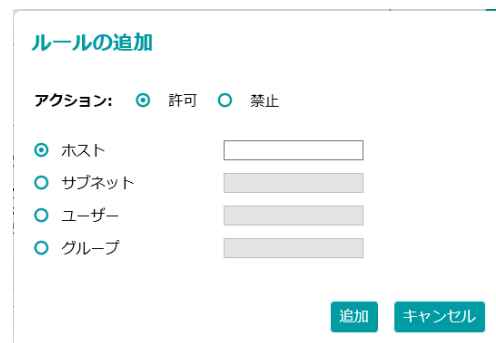
アドバンスモードには、「グローバルアクセスルール」と「特定のアクセスルール」の2種類があります。「グローバルアクセスルール」はすべてのCmContainerのライセンスアクセスをコントロールします。特定のアクセスルールを定義した場合、グローバルルールは無視されます。「特定のアクセスルール」は指定したライセンスエントリ(ファームコード、プロダクトコード)のライセンスアクセスをコントロールします。特定のアクセスルールを定義した場合、グローバルルールは無視されます。

グローバルアクセスルール:

デフォルトは「許可」に設定されているため、新規ルールに何も追加しない場合は、すべてのクライアントからアクセスが可能になります。新規ルールを追加する場合は、「新しいアクセスルールを追加します」ボタンをクリックして設定します。

新規ルールの追加として、

- ホスト (Host)
 - サブネット (Subnet)
 - ユーザー (User)
 - グループ (Group)
- の設定が可能です。



特定のアクセスルール:

「特定のアクセスルール」は指定したライセンスエントリ(ファームコード、プロダクトコード)のライセンスアクセスをコントロールします。特定のアクセスルールを定義した場合、グローバルルールは無視されます。

現在のサーバ: localhost (127.0.0.1) Webアドミンバージョン: : 6.40

「ファームコードの追加」画面でファームコードを指定します。

構成 / アドバンス / タイムサーバー

タイムサーバー

CmContainerの認証タイムを更新する際に使用するタイムサーバーを設定します。

現在、下記4つのタイムサーバーが登録され、正確な時刻を同期しています。

cmtime.codemeter.com

cmtime.codemeter.us

cmtime.codemeter.fr

cmtime.codemeter.de

The screenshot shows the 'CodeMeter Web アドミン' interface. The '構成' (Configuration) menu is selected, and the 'タイムサーバー' (Time Servers) sub-menu is active. A list of four time servers is displayed:

順位	タイムサーバー	操作
1.	cmtime.codemeter.com	音 ↓
2.	cmtime.codemeter.us	音 ↓ ↑
3.	cmtime.codemeter.fr	音 ↓ ↑
4.	cmtime.codemeter.de	音 ↑

Below the list, there is a '+ 新しいタイムサーバーの追加' (Add new time server) button and a 'タイムアウト: 20 秒' (Timeout: 20 seconds) field. At the bottom, there are '適用' (Apply) and 'デフォルトを元に戻す' (Reset to default) buttons. The status bar at the bottom indicates '現在のサーバ: localhost (127.0.0.1)' and 'Webアドミンバージョン: 6.40'.

CmContainerの認証タイムを更新するには、「コンテナ」/CmContainer情報」画面の「認証タイム (CmContainer)」の右端の更新ボタンをクリックします。上記タイムサーバーに接続され、「認証タイム (CmContainer)」を正確な時刻に更新します。

The screenshot shows the 'CodeMeter Web アドミン' interface for a specific container. The 'コンテナ' (Containers) menu is selected, and the container '<no name> (3-3302804)' is selected. The 'CmContainer情報' (CmContainer Information) tab is active, displaying the following details:

項目	値	操作
名前	<no name>	
シリアル番号	3-3302804	
CmContainer のタイプ:	CmStick 3.12	
最初のデバイス:	No drive assigned (HID)	
ステータス	● 使用可能	
システムタイム (PC):	2017-01-27 17:36:11	
システムタイム (CmContainer):	2017-01-27 17:36:15	
認証タイム (CmContainer):	2017-01-27 17:36:11	更新
空きメモリー	94 % (315.864 Bytes)	

A red box highlights the '認証タイム (CmContainer):' row, and a red arrow labeled 'Click' points to the update button in the right margin. The status bar at the bottom indicates '現在のサーバ: localhost (127.0.0.1)' and 'Webアドミンバージョン: 6.40'.

構成 / アドバンス / エクストラ

APIコミュニケーションモード

CodeMeter License Serverが使用するコミュニケーションモードを設定します。

デフォルトでは、

IPv4

IPv6

シェアードメモリー

が設定されています。

[NOTE]

特別な理由がない限り、デフォルトの状態での使用をお勧めいたします。

ネットワークアドバンス

CodeMeter License Serverの環境設定を変更します。

デフォルトでは、

ネットワークポート：22350

ネットワークタイムアウト：100 秒

UDP(User Datagram Protocol)待ち時間：1000 ミリ秒 (1秒)

が設定されています。

[NOTE]

デフォルト設定を変更するとサービスのパフォーマンスが変わります。緊急の場合以外は、デフォルト状態でご使用されることをお勧めいたします。

The screenshot shows the CodeMeter Web Admin interface. The top navigation bar includes 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The '構成' menu is highlighted. Below the navigation bar, there are tabs for 'アドバンス定義' and 'エクストラ'. The 'APIコミュニケーションモード' section is active, showing three checked options: 'IPv4', 'IPv6', and 'シェアードメモリー'. A red box highlights the '編集' (Edit) button next to these options. Below this, the 'ネットワークアドバンス' section shows input fields for 'ネットワークポート' (22350), 'ネットワークタイムアウト' (100 秒), and 'UDP待ち時間' (1000 1000分の1秒). A red box highlights the '編集' (Edit) button next to these fields. A large circular logo with '{API}' is visible on the right side of the settings area. The footer shows '現在のサーバ: localhost (127.0.0.1)' and 'Webアドミンバージョン: : 6.40'.

12-5. 情報



The screenshot displays the CodeMeter Web Admin interface. At the top left is the WIBU SYSTEMS logo. The main title is 'CodeMeter Web アドミン'. A navigation bar contains 'ダッシュボード', 'コンテナ', 'ライセンスモニタリング', '診断', '構成', and '情報'. The '情報' page features a header image with various CodeMeter hardware tokens (CmActLicense, CmDongle) and a paragraph of Japanese text explaining the technology. Below this is a support contact section with links to a support home page and an email address. The footer shows the current server as 'localhost (127.0.0.1)' and the version as '6.40'.

WIBU SYSTEMS CodeMeter Web アドミン

ダッシュボード コンテナ ライセンスモニタリング 診断 構成 情報

情報 ? 日本語



CodeMeterは確実なハードウェアベースとソフトウェアベースのソフトウェアプロテクション、デジタルコンテンツのためのライセンステクノロジーを提供します。CodeMeterライセンスはコンピュータ (CmActLicense)やハードウェアトークン (CmDongle)に格納されます。CodeMeterハードウェアは異なる形状が利用できます。

トラブルシューティングまたはサポートの場合 [サポートホームページ](#) Eメールを送信 support@wibu.com
追加のツールやヘルプはWebサイト上で探すことができます。 www.wibu.com

現在のサーバ: localhost (127.0.0.1) Webアドミンバージョン: : 6.40

Chapter 13

Adobe PDF ファイルにプロテクトをかける

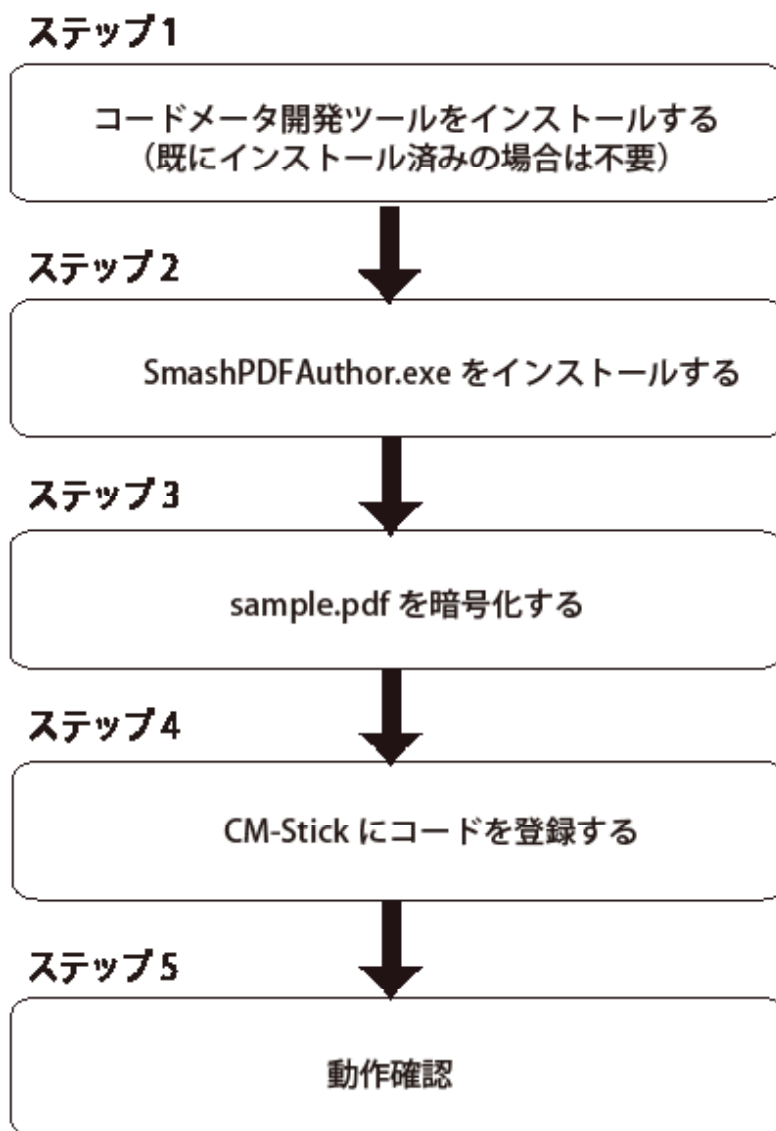
- 13-1. Adobe PDF ファイルにプロテクトをかける
- 13-2. 作業に必要なもの
- 13-3. コードメータ開発ツールをインストールする
- 13-4. SmartShelter PDF Author Ver18.04 をインストールする
- 13-5. sample.pdf を暗号化する
- 13-6. CmStick にコードを登録する
- 13-7. 動作を確認する
- 13-8. 複数のファームコード/プロダクトコードを入れて暗号化する
- 13-9. プレインテキストパスワードについて
- 13-10. 複数の PDF ファイルを一括して暗号化する
- 13-11. コマンドラインで暗号化する方法
- 13-12. 暗号化された PDF ファイルをユーザーに配布する場合
- 13-13. 起動時に保護モードを無効化する方法
(Adobe Reader DC/XI)

13-1. Adobe PDF ファイルにプロテクトをかける

コードメータには、Adobe PDFファイルのプロテクトするSmartShelterPDF (スマートシェルタPDF) 機能が搭載されています。このSmartShelterPDF (スマートシェルタPDF) 機能を利用することで、Adobe PDFファイルを簡単に暗号化することができます。貴社の重要な技術情報やマニュアル、仕様書や見積書など、限られた人に見せたい場合、さらに閲覧期間や閲覧回数、印刷回数を設定したい場合、またPDFファイルの外部流出を防ぎたい場合などに役に立ちます。ユーザーへのデータ配布、本支店間や取引先間でのデータのやりとり、社内の情報漏えい対策に、またはコンテンツ自体の販売に非常に効果的です。(暗号化アルゴリズム AES 128ビット)

コードメータDVDの中にあるsample.pdfにプロテクトをかけてみます。sample.pdfは、コードメータDVDのTools¥SmartShelterPDFフォルダの中に格納されています。PCのローカルディスクにコピーしてお使いください。

作業の流れとして、以下のようになります。



13-2. 作業に必要なもの

Adobe PDFファイルを暗号化するために必要なものは下記になります。

- ① Adobe Acrobat 9/X/XI/DC/2017のいずれか (Adobe Readerでは暗号化作業ができません)
- ② 貴社のコードメータFSB (CmFSB)

作業はWindows 7/8/8.1/10 (32bit/64bit)上で行います。

[NOTE]

暗号化作業はWindowsのAdobe Acrobat上で行う必要がありますが、暗号化されたPDFファイルは、Windows上のAdobe AcrobatおよびAdobe Reader、Mac OS X上のAdobe AcrobatおよびAdobe Reader上でそれぞれ開くことができます。

13-3. コードメータ開発ツールをインストールする

まずはじめに、コードメータ開発ツールをインストールします。すでに、コードメータ開発ツールがインストールされている場合は、次の「13-4. SmartShelter|PDF Authorをインストールする」に進んでください。まだ、コードメータ開発キットのインストールが済んでいない場合は、SmartShelter|PDF Authorをインストールする前にコードメータ開発ツールをインストールしてください。コードメータ開発ツールのインストールについては、「Chapter 2 コードメータ開発ツールをインストールする」をご参照ください。

13-4. SmartShelter PDF Author Ver18.04 をインストールする

SmartShelter PDF Author Ver18.04 をインストールします。SmartShelter PDF Author Ver18.04 は、コードメータDVD Ver6.80a のSmartShelterPDFフォルダの中の"SmashPdfAuthor.exe"を実行してインストールします。また、最新のSmashPDFAuthor.exeは、弊社下記サイトよりダウンロードできます。

<http://www.suncarla.co.jp/download/>

① SmashPdfAuthor.exe を起動する

コードメータDVDをDVDドライブに挿入し、SmartShelterPDFフォルダの中の"SmashPdfAuthor.exe"をダブルクリックして起動します。

SmartShelter PDF Authorインストール画面が表示されますので、「次へ」をクリックします。

② 「インストール範囲」を選択する

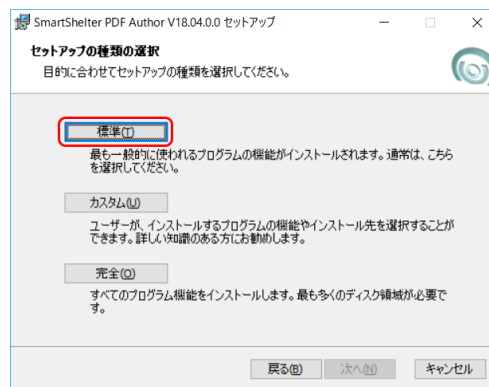
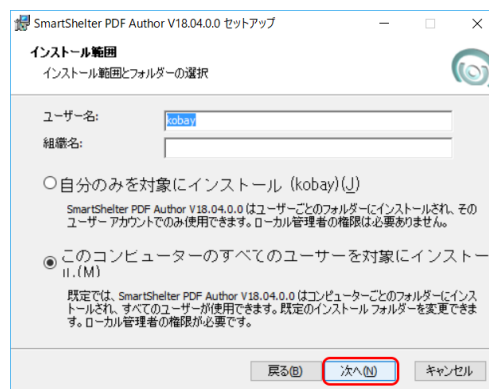
SmartShelterPdfAuthorをインストール範囲を選択します。

③ 「標準」 ボタンをクリックする

セットアップの種類を選択画面が表示されますので、「標準」ボタンを直接クリックします。

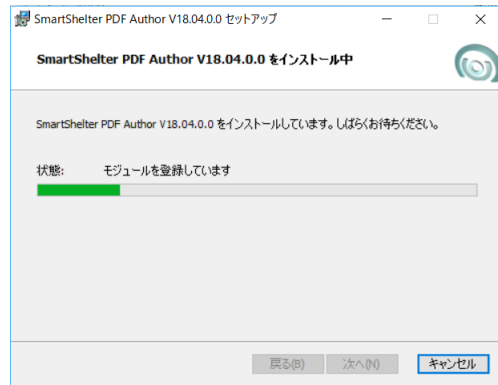
④ 「インストール」 ボタンをクリックする

SmartShelter PDF Author V18.04.0.0のインストール準備完了画面で「インストール(I)」をクリックします。



⑤ インストールが開始される

SmartShelter PDF Author V18.04.0.0がインストールされます。

**⑥ 正常にインストールされた**

インストールが正常に行われると右の画面が表示されます。「完了(F)」ボタンをクリックしてインストール画面を閉じます。



13-5. sample.pdf を暗号化する

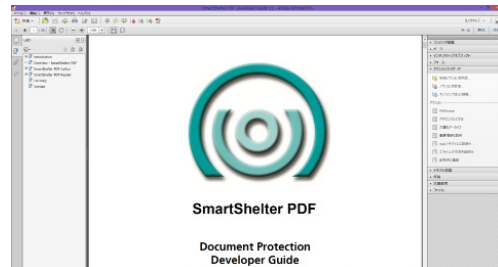
① コードメータ FSB (CmFSB) を装着する

まずはじめに、貴社のコードメータFSB(CmFSB)をPCに装着します。PDFファイルを暗号化する場合、必ずコードメータFSB(CmFSB)が必要になります。

② Adobe Acrobat を起動する

次に、Adobe Acrobatを起動し、[ファイル]/[開く]メニューから、sample.pdfを開きます。sample.pdfは、コードメータDVDのTools¥SmartShelterPDFフォルダにあります。

ここでは、Acrobat Pro XI の画面で説明します。Acrobat 9/X/DC の場合も基本的操作は同じです。

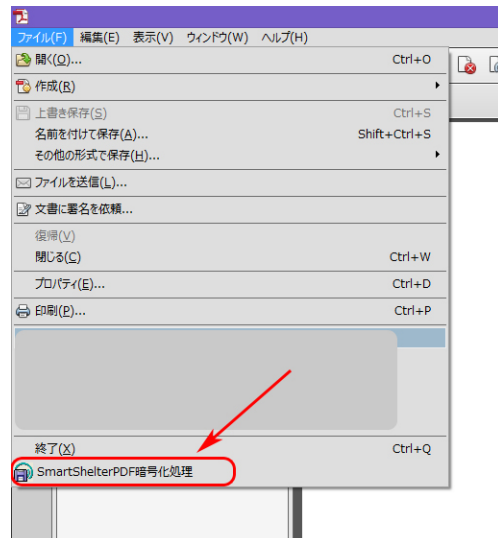


③ 「SmartShelterPDF 暗号化処理」を選択

[ファイル]メニューから、「SmartShelterPDF暗号化処理」メニューを選択しクリックします。

[NOTE]

「SmartShelterPDF暗号化処理」メニューが表示されない場合は、SmartShelter PDF Authorが正しくインストールされていない場合が考えられます。再度、「13-4. SmartShelter PDF Author Ver18.04をインストールする」に戻り、SmartShelter PDF Authorをインストールしてください。また、PDFファイルをAdobe Readerから開いていないかも確認してください。



④ 暗号化設定画面が表示される

SmartShelter PDF暗号化設定画面が表示されます。この画面で、貴社のファームコードやプロダクトコード、その他のセキュリティオプションを設定します。

PDFファイルは、ここで指定されたファームコードやプロダクトコード等を要素に暗号キーを生成し、その暗号キーをベースに暗号化されます。

従い、暗号化仕様は、常に貴社独自の仕様になります。



⑤ ファームコードを設定する

ファームコード項目に、貴社のファームコードを入力します。ここでは、サンプルファームコード"10"を入力します。

⑥ プロダクトコードを設定する

プロダクトコード項目に、プロダクトコードを入力します。ここでは、"13"を入力します。

プロダクトコードは、1~4294967295の範囲の整数値(32ビット)が可能です。

[NOTE]

プロダクトコード 1~4294967295以外の数値を指定すると、プロダクトコード=0で暗号化されますのでご注意ください。

⑦ セキュリティオプションを設定する

必要に応じて、セキュリティオプションを設定します。

開始期日チェック

暗号化したPDFファイルをいつから開くことができるかの閲覧開始期日設定を行います。

ユニットカウンタチェック

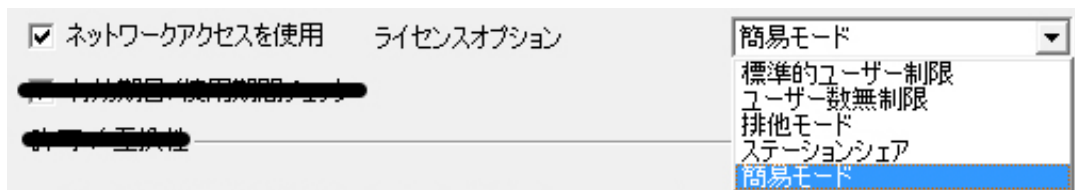
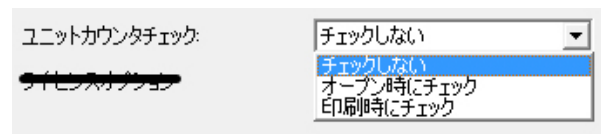
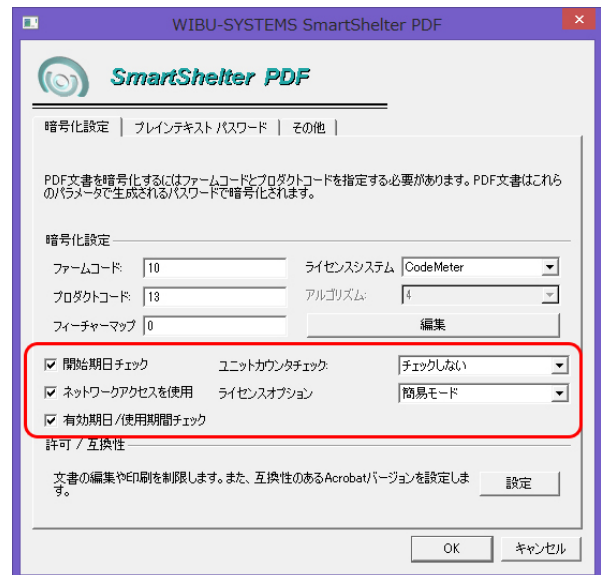
暗号化したPDFファイルのオープン回数または印刷回数を設定します。ファイルオープンまたは印刷を行うたびに、ユニットカウンタが1つずつ減ります。とくにユニットカウンタを使用しない場合は、"チェックしない"を選択します。

(3つの選択モード)

- チェックしない
- オープン時にチェック
- 印刷時にチェック

ネットワークアクセスを使用

コードメータのネットワーク機能を使用します。サーバーにCmStickを装着することで、クライアントから暗号化済みPDFファイルを開くことが可能になります。また、ネットワークカウンタを使って、フローティングライセンス(ネットワークライセンス)を制御することも可能です。ライセンスモードは5通りあります。



ライセンスオプション:

○ 標準的ユーザー制限

暗号化済みのPDFファイルを開くごとに1つのライセンスを割り当てます。

(1 PDFファイル = 1ライセンスの考え方)

○ ユーザー数無制限

コードメタキー (CmDongle) がネットワーク(LAN)上で見つければ、ライセンス数の制限にかかわらず暗号化済みのPDFファイルは無制限で開きます。ライセンス数(ネットワークカウンタ)の制約を受けません。(当然、コードメタキーにアクセスできない場合は開きません)

○ 排他モード(クライアントごとに1つのPDFアプリケーションのみ許可)

1台のPCで同一PDFファイル(ファームコード、プロダクトコードなどが同一のPDFファイル)の重複オープン(開く)を禁止します。ファームコード、プロダクトコードが異なるPDFファイルは別のファイルと処理するためオープン(開く)できます。(ただし、排他モードでのオープンになります)

○ ステーションシェア(クライアント単位で制限)

1台のPCで同一PDFファイル(ファームコード、プロダクトコードなどが同一のPDFファイル)を同時に複数回オープン(開く)した場合でも1ライセンスとして扱われます。

(1 PC = 1ライセンスの考え方)

○ 簡易モード(ネットワーク上では標準的なユーザー制限、ローカル上は制限なし)

ネットワーク(LAN)上のコードメタキー (CmDongle) に対しては「標準的ユーザー制限(1 PDFファイル = 1ライセンス)」として動作しますが、ローカルPC上のコードメタキー (CmDongle) に対しては制限がありません。

* CmDongleをローカルPCに装着すると、そのPC上では同一PDFファイル(ファームコード、プロダクトコードなどが同一のPDFファイル)が無限に開けるが、そのCmDongleをコードメタサーバーに装着すると、「標準的ユーザー制限(1アプリケーション=1ライセンス)」として機能する。

有効期日/使用期間チェック

暗号化したPDFファイルをいつまで開くことができるかの有効期限を設定します。有効期日 (Expiration Time) および使用期間 (Usage Period) の両方が使用可能です。

[NOTE]

実際のファームコード、プロダクトコード、開始期日、有効期限、ユニットカウンタの数値はCmStick(ドングル)側に登録します。

⑧ 「設定」 ボタンをクリックする

ファームコード、プロダクトコード、セキュリティオプションを入力したら、「許可/互換性」設定を行います。「設定」ボタンをクリックします。

⑨ 「許可 / 互換性」 項目を設定する

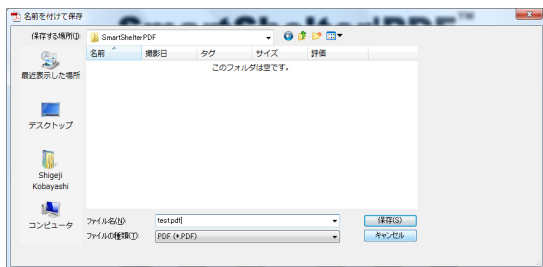
貴社のセキュリティニーズに従い、「許可/互換性」項目を設定します。

ここでは、プリントスクリーンを禁止する「プリントスクリーン禁止」、スクリーンキャプチャを禁止する「プロセスチェック」、デバッガーによる暗号解析を防止する「アドバンスドデバッガーチェック」を選択します。その他の項目については、必要に応じて選択してください。

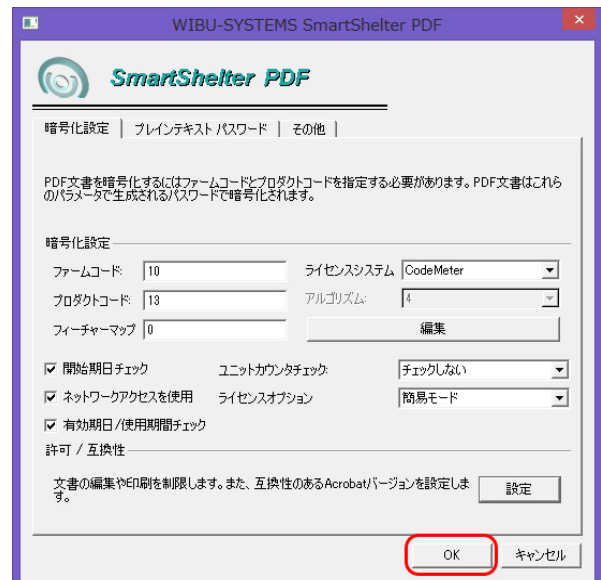
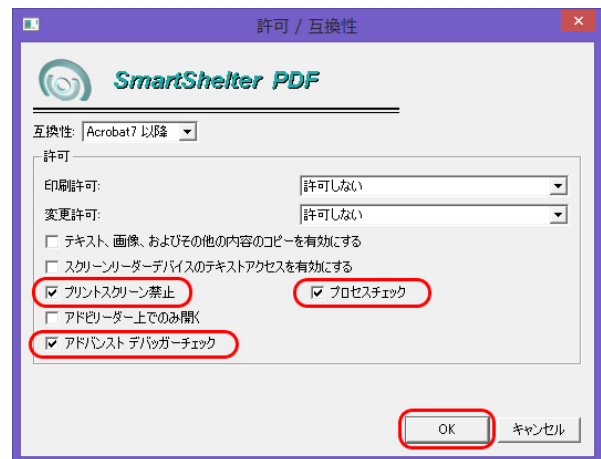
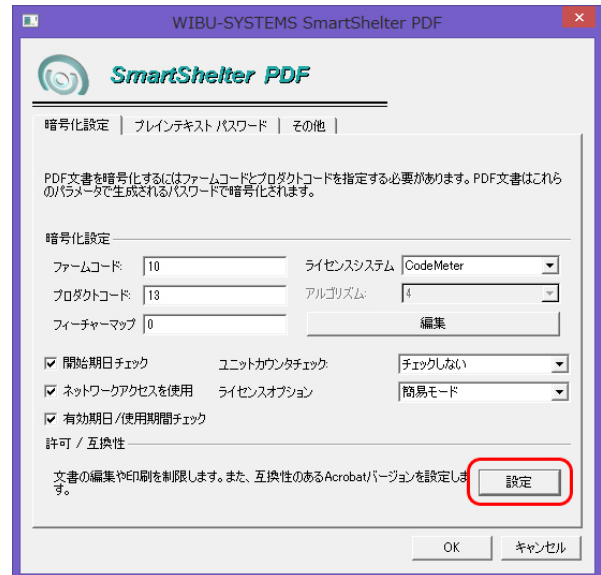
チェックを入れたら、「OK」をクリックし、暗号化設定画面に戻ります。

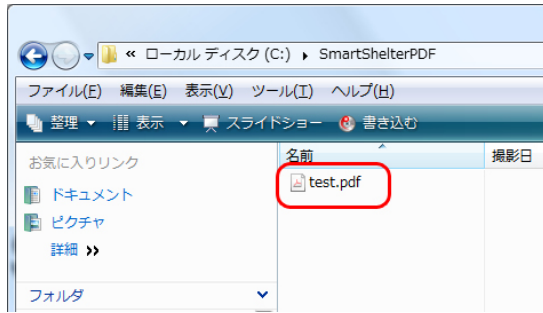
⑩ 暗号化処理を開始する

暗号化設定画面の「OK」をクリックすると、「名前を付けて保存」画面が表示されますので、ファイル名を入力して保存します。ここで保存されるときに、暗号化されたPDFファイルが作成されます。



ここでは、「test.pdf」と入力して保存します。Acrobatを閉じて、指定したフォルダに「test.pdf」が作成されていることを確認してください。sample.pdfが暗号化されてtest.pdfとして作成されました。





これで、sample.pdfの暗号化処理は終了です。次に、ファームコード=10、プロダクトコード=13を持つCmStickを装着して暗号化されたtest.pdfを開いてみます。

13-6. CmStick にコードを登録する

コードメータCmStickにファームコード=10、プロダクトコード=13を登録します。CmStickにコードを登録するには、コードメータライセンスエディタ(CodeMeter License Editor)を使用します。登録の方法は、「Chapter3 実行形式プログラムにプロテクトをかける/3-3.CmStick(CmDongle)にコードを登録する」をご参照ください。

[NOTE]

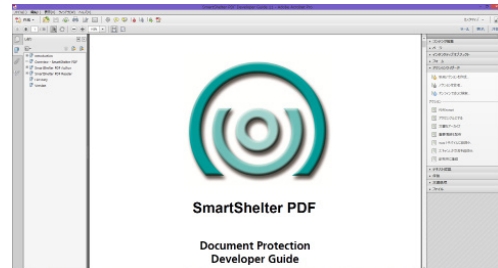
ファームコードやプロダクトコードなどを登録する方法は、CmStick, CmStick/M2GB, CmCard/SD, CmCard/MicroSD, CmCard/CF, CmCard/Eなどすべて共通です。

13-7. 動作を確認する

作成したCmStickをPCに装着し、先ほど暗号化したtest.pdfファイルを開きます。CmStickが装着されていれば開き、装着されていないと開かないことが確認できます。

① CmStick が装着されていると開く

ファームコード=10、プロダクトコード=13が登録されたCmStickをPCに装着した状態で、test.pdfを開くとオリジナルファイルと同じように開きます。



② CmStick がないと開かない

CmStickをPCからはずした状態でtest.pdfを開くとエラーになります。これで、ファイルがプロテクトされていることが確認できます。



[NOTE]

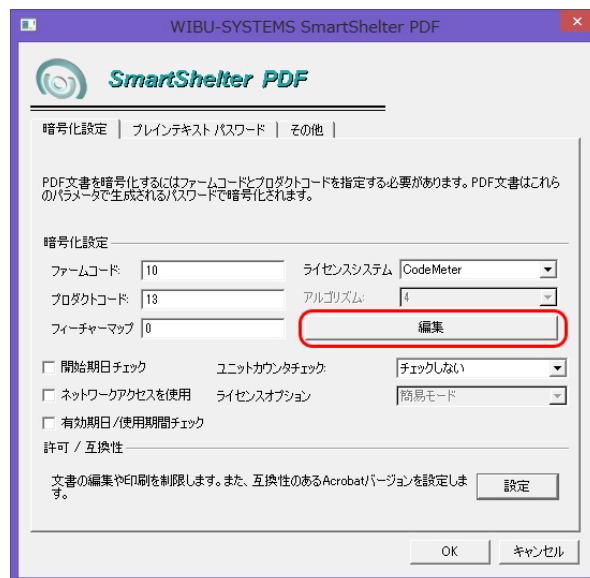
エラーメッセージをカスタマイズすることが可能です。「13-9. エラーメッセージをカスタマイズする」を参照してください。

13-8. 複数のファームコード / プロダクトコードを入れて暗号化する

SmartShelterPDFでは、1つのPDFファイルを暗号化する際、複数のファームコード・プロダクトコード等を入れて暗号化するマルチコード暗号化が可能です。異なるファームコードやプロダクトコードを持つCmStickでも、共通して開けるPDFファイルを作成することができます。

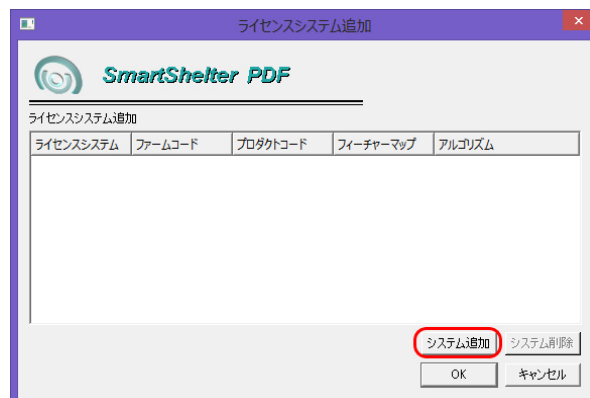
①「編集」ボタンをクリックする

暗号化設定画面の「編集」ボタンをクリックします。



②「システム追加」ボタンをクリックする

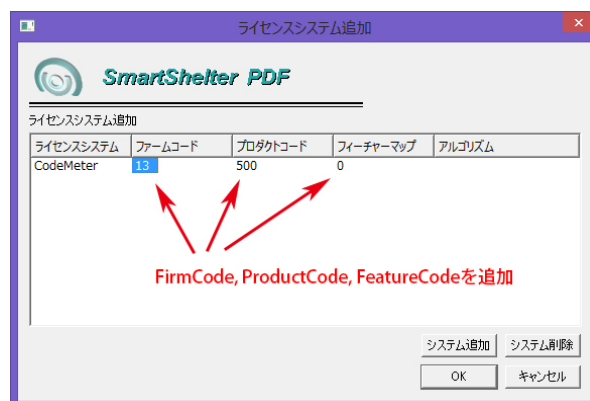
ライセンスシステム追加画面で「システム追加」ボタンをクリックします。



③ コードを追加する

各数値にカーソルを合わせ、ダブルクリックすると編集できます。複数追加する場合は、「システム追加」ボタンをクリックします。

追加可能な内容は、
ファームコード
プロダクトコード
フィーチャーマップ
です。



[NOTE]

ライセンスシステムでWibuKeyも選択できますが、WibuKeyはSmartShelterPDFを正式サポートしていません。WibuKeyでもプロテクト可能ですが、動作保証はしていませんのでご注意ください。

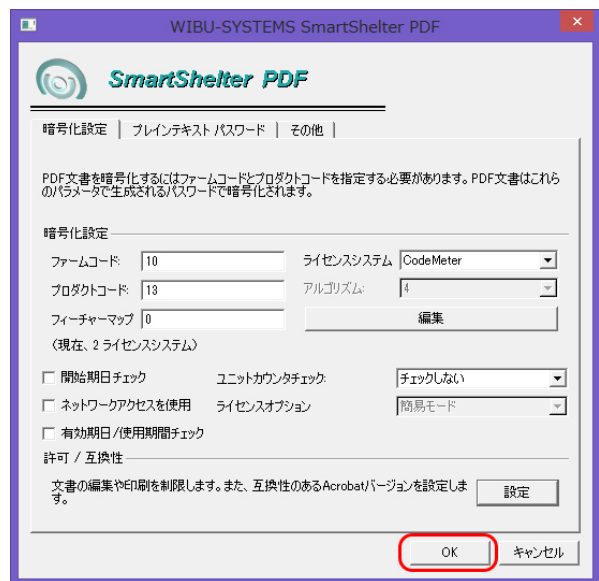
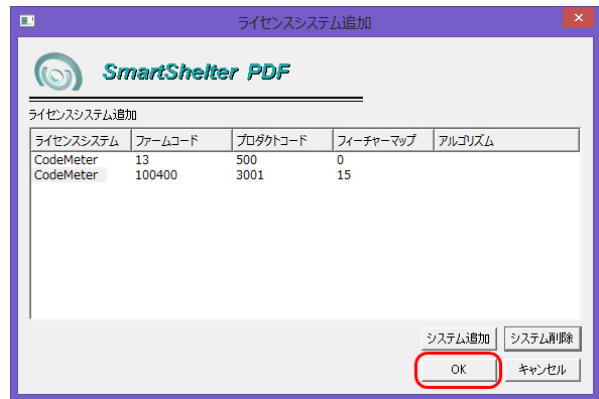
④ 「OK」 ボタンをクリックする

ライセンスシステムの追加が完了したら「OK」ボタンをクリックします。

⑤ あとは通常どおり暗号化する

暗号化設定画面に戻り、「OK」ボタンをクリックし、通常通り暗号化作業を行います。

作成されたPDFファイルは、暗号化設定画面で指定したファームコード/プロダクトコード等と、ライセンスシステム追加画面で指定したファームコード/プロダクトコード等の両方を取り込みます。いずれかのCmStickが存在すればPDFファイルは開きます。



13-9. プレインテキストパスワードについて

PDFファイルを暗号化する際、ファームコードやプロダクトコード、またその他の要素をベースにプレインテキストパスワードが生成されます。これは、CmStickが無くても暗号化されたPDFファイルを開くためのパスワードです。ユーザーには解放しないようにしてください。「表示」ボタンをクリックすると、可視状態になります。

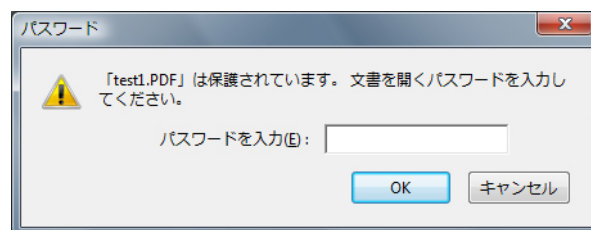
パスワードには、プレインテキストパスワードとオーナーパスワードの2種類があります。

プレインテキストパスワード

暗号化されたPDFファイルを開くことが可能です。ただし、ドキュメントの編集はできません。

オーナーパスワード

暗号化されたPDFファイルを開くことが可能です。また、ドキュメントの編集も可能です。



13-10. 複数の PDF ファイルを一括して暗号化する

複数のPDFファイルを一括して暗号化する場合は、Acrobatのバッチ処理機能を利用します。このバッチ処理機能を利用するには、Acrobatのプロフェッショナル版が必要です。Acrobatのスタンダード版にはバッチ処理機能がありませんので、この作業はできません。(Acrobat Xの場合はアクションウィザード処理)

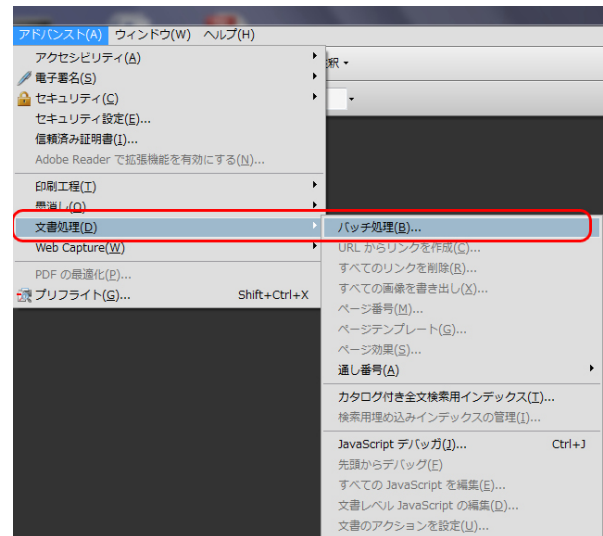
Acrobat 9 ProとAcrobat X Proでのバッチ処理方法をご説明いたします。その他のAcrobatバージョンでも、メニュー画面が異なりますが、基本的操作は同じです。ただし、Acrobat XIまたはAcrobat DCの場合、PDFファイルを暗号化するたびにファイルを開くため、ファイル数が多いとあまり効率的ではありません。その場合は、後述するコマンドラインによる暗号化処理をお奨め致します。

Acrobat 9 Pro の場合

① 「バッチ処理」メニューを選択する

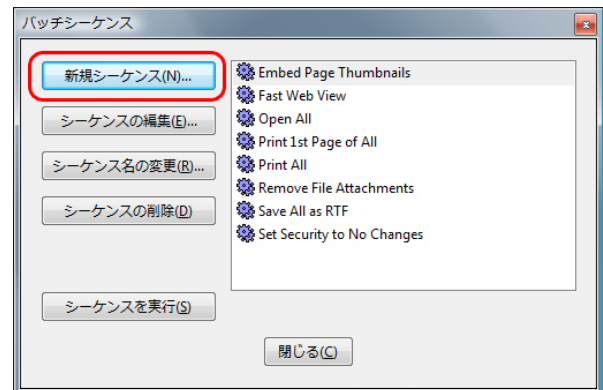
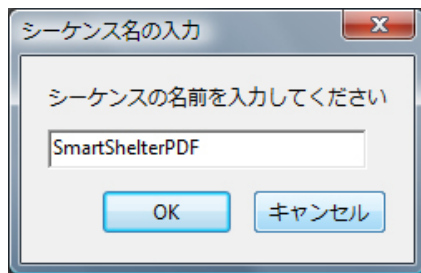
Acrobatの「アドバンスド」/「文書処理」メニューから「バッチ処理」を選択します。Acrobat6/7/8の場合、メニュー画面が多少異なりますが、「バッチ処理」を選択してください。

「バッチ処理」をクリックすると、バッチシーケンス画面が表示されます。このバッチシーケンスにPDF暗号化処理を登録します。



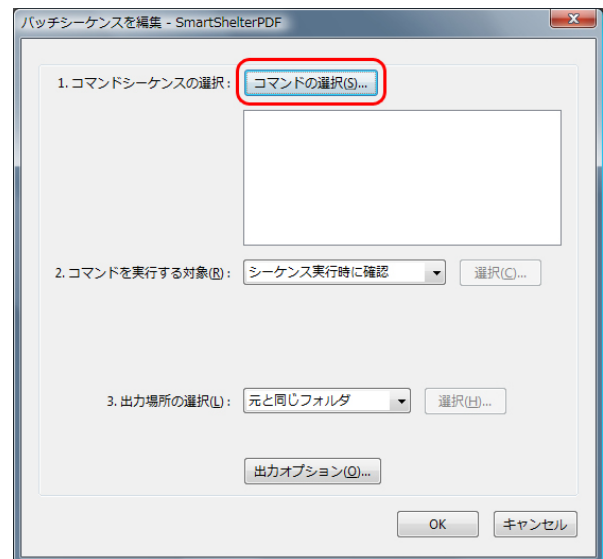
② 「新規シーケンス」をクリックする

「新規シーケンス」をクリックすると、シーケンス名の入力画面が表示されますので、任意の名前を入力します。ここでは、SmartShelterPDFと入力します。



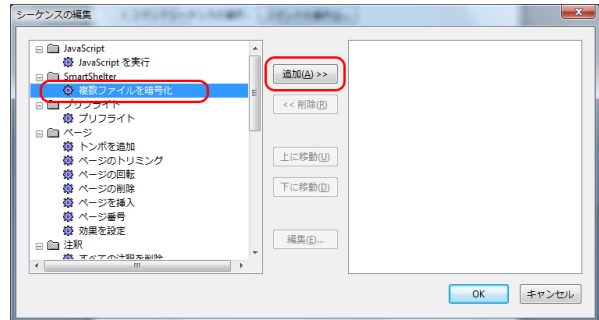
③ 「コマンドの選択 (S)」をクリックする

「コマンドの選択(S)」をクリックし、シーケンスの編集画面を開きます。

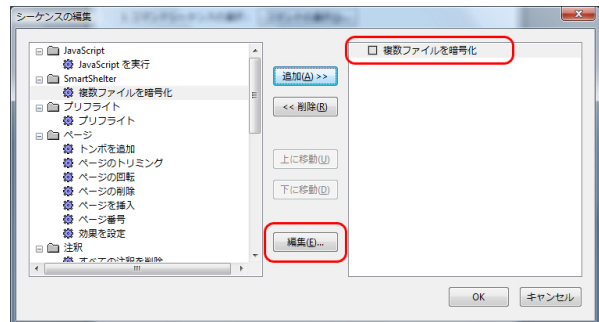


④ 「複数ファイルを暗号化」を選択する

シーケンスの編集画面で、SmartShelterの「複数ファイルを暗号化」を選択し、「追加(A)」ボタンをクリックします。

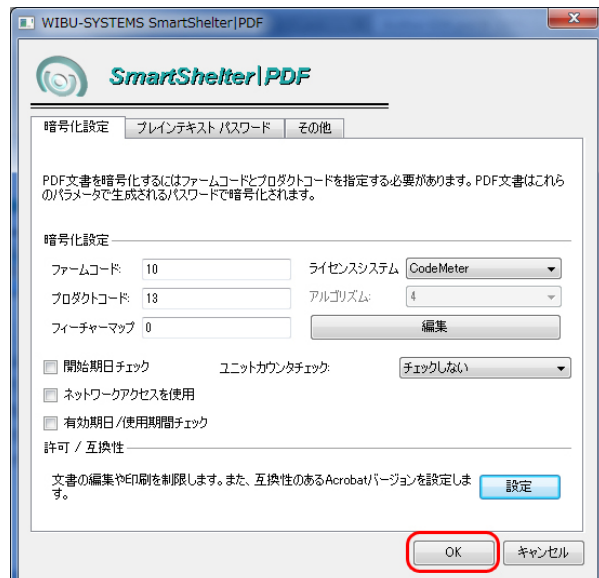
**⑤ 「編集(E)」ボタンをクリックする**

「複数ファイルを暗号化」が追加されたことを確認したら、「編集(E)」ボタンをクリックします。

**⑥ 暗号化内容を設定する**

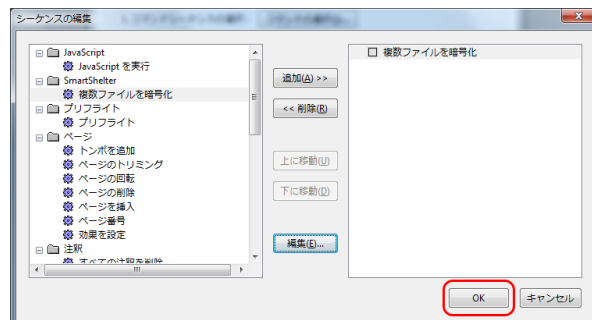
SmartShelterPDFの暗号化内容の設定画面が開きますので、ここで暗号化設定を行います。必要に応じてオプション項目を設定します。

暗号化設定が完了したら、「OK」ボタンをクリックします。



⑦ 「OK」 ボタンをクリックする

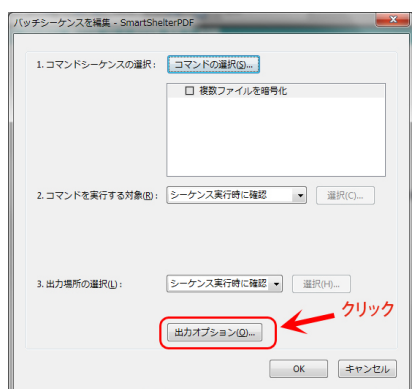
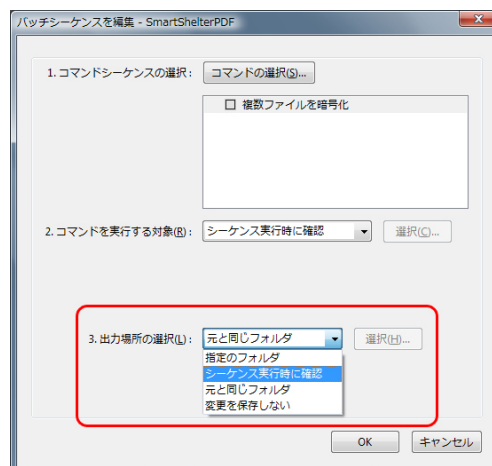
シーケンスの編集画面に戻り、「OK」ボタンをクリックします。



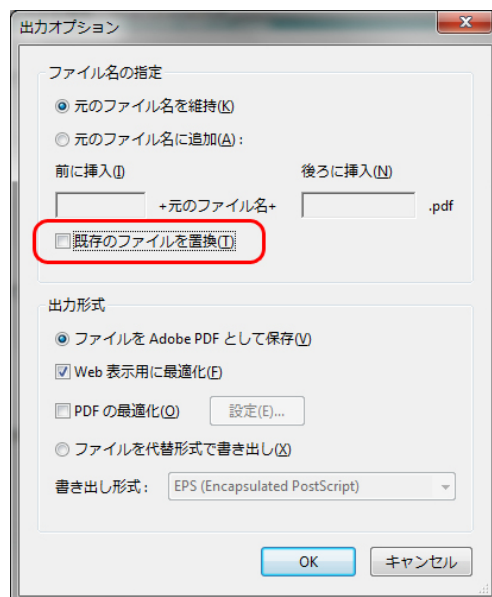
⑧ バッチシーケンスを編集する

「バッチシーケンスを編集」画面の「出力場所の選択」で「シーケンス実行時に確認」を選択します。「元と同じフォルダ」を選択すると、オリジナルファイルが暗号化ファイルに上書きされるのでご注意ください。

また、念のため、「出力オプション」を選択し、「既存のファイルを置換(T)」のチェックを外しておく、と、上書き保存の防止になります。なお、バッチシーケンス編集についての詳しい操作方法はAcrobatのマニュアル等をご参照ください。



「OK」ボタンをクリックすると、バッチシーケンスが登録されます。

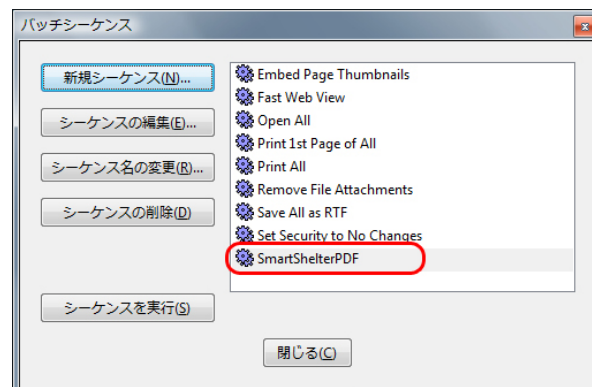


⑨ SmartShelterPDF が登録された

「バッチシーケンス」画面に「SmartShelterPDF」シーケンスが登録されたことを確認します。

「閉じる」ボタンをクリックすると、バッチ処理が終了します。

登録するバッチシーケンスは、暗号化内容に応じて複数作成することができます。貴社のセキュリティ内容に応じて使い分けてください。



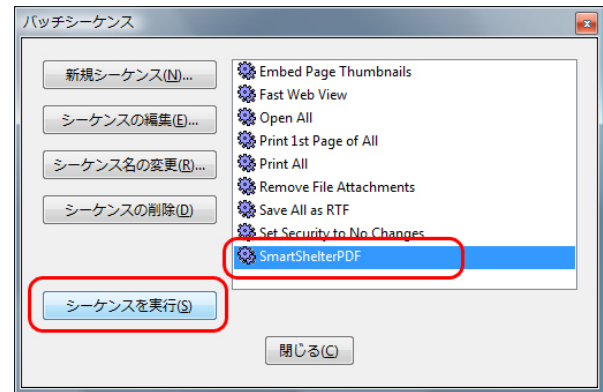
バッチシーケンスを実行して、PDF ファイルを一括暗号化する

① コードメータ FSB を装着する

まず、コードメータFSBをPCに装着します。PDFファイルを暗号化するには、必ずコードメータFSBが必要になります。

② バッチシーケンスを開く

Acrobatを起動し、「アドバンスド」/「文書処理」メニューから「バッチ処理」を選択し、バッチシーケンスを開きます。

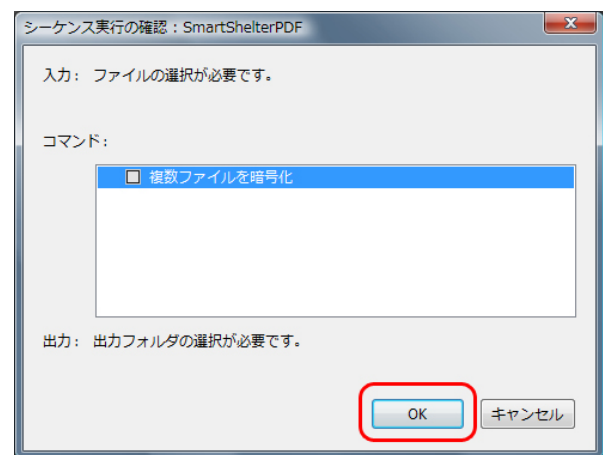


③ バッチシーケンスを実行する

SmartShelterPDFを選択し、「シーケンスを実行」をクリックします。

④ シーケンス実行の確認

「シーケンス実行の確認」画面で確認後、「OK」ボタンをクリックします。

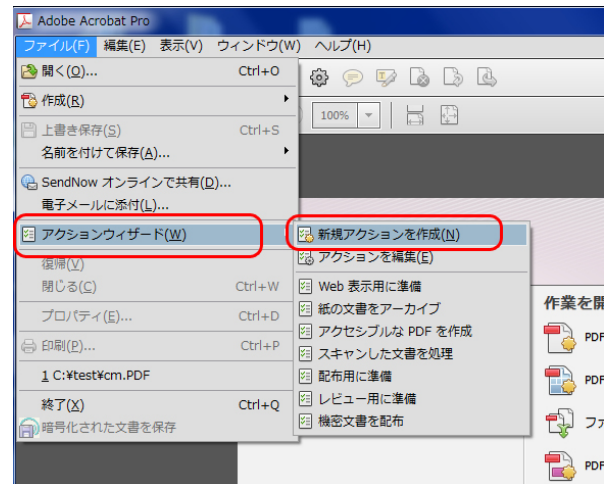


あとは、暗号化するPDFファイルを選択し、指定するフォルダに作成するだけです。

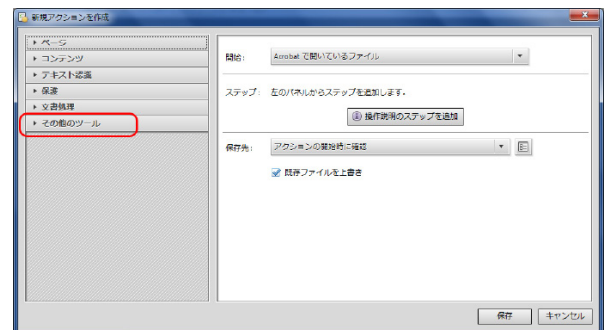
Acrobat X Pro の場合

① 「新規アクションを作成」メニューを選択

Acrobat X Proの「ファイル(F)」「アクションウィザード(W)」メニューから「新規アクションを作成(N)」を選択します。

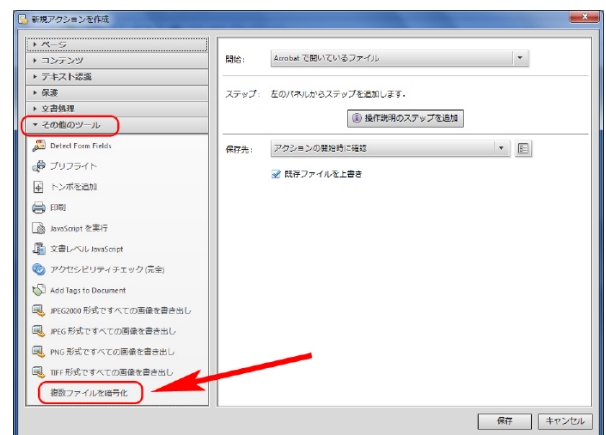


「新規アクションを作成(N)」をクリックすると、「新規アクションを作成」画面が表示されます。



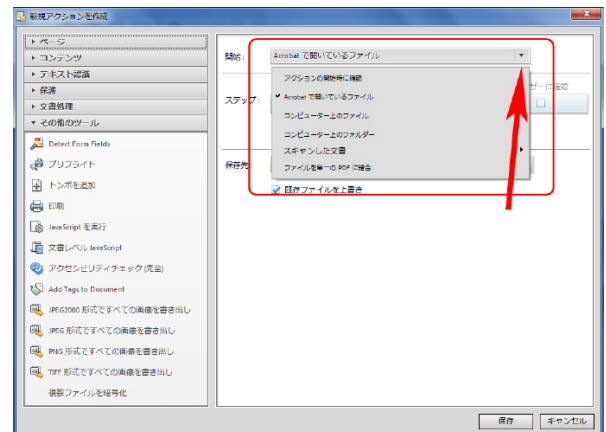
② 「複数ファイルを暗号化」を選択する

新規アクションを作成画面で、「その他のツール」を選択し、「複数ファイルを暗号化」をクリックします。

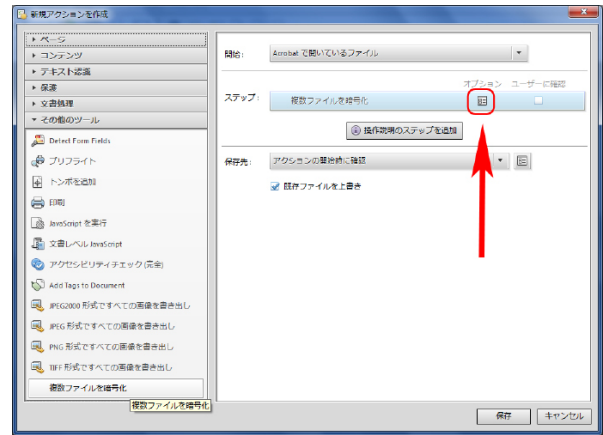


③ 暗号化するファイルを選択する

「開始」項目で、暗号化するファイルを選択します。



④ 「オプション」 アイコンをクリックする
「ステップ」項目のオプションアイコンをクリックします。

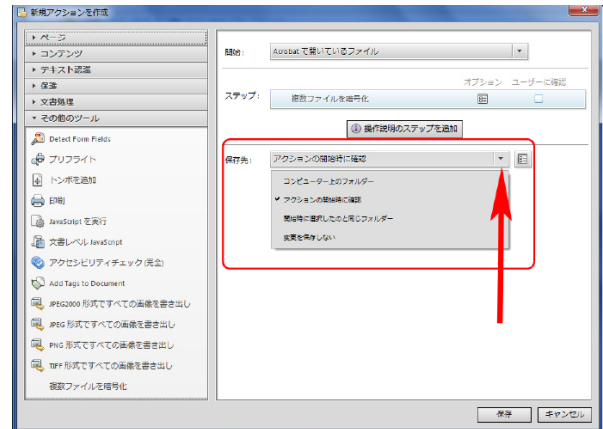


⑤ 暗号化内容を設定する
SmartShelterPDFの暗号化内容の設定画面が開きますので、ここで暗号化設定を行います。必要に応じてオプション項目を設定します。



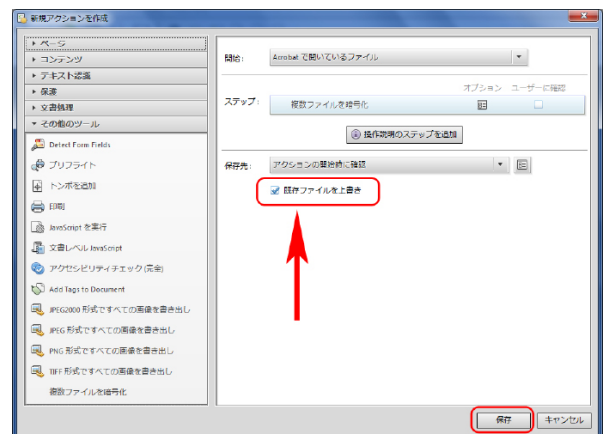
暗号化設定が完了したら、「OK」ボタンをクリックします。

⑥ 保存先を指定する
暗号化されたファイルの保存先を指定します。



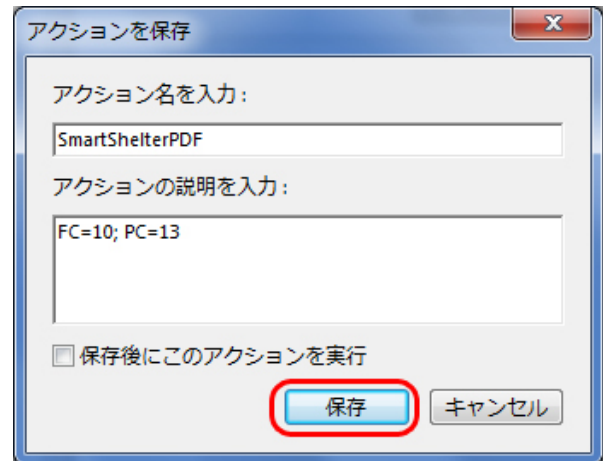
既存ファイルを上書きする場合は、「既存ファイルを上書き」にチェックを入れます。

指定が完了したら「保存」ボタンをクリックします。



⑦ アクション名を入力する

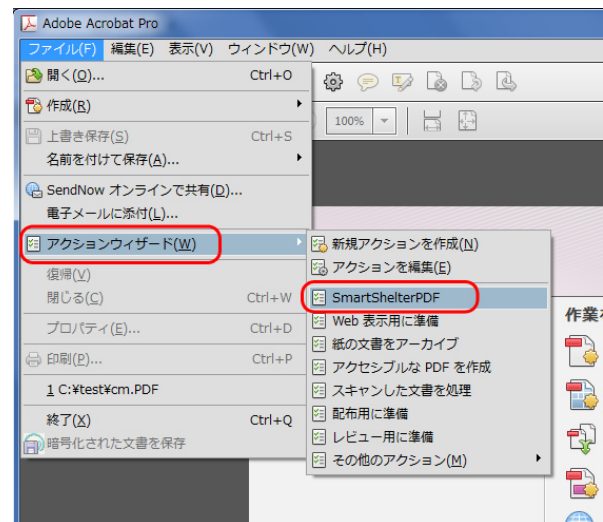
アクションを保存画面で、アクション名を入力します。必要に応じて、アクションの説明を入力してください。入力完了したら「保存」ボタンをクリックします。これで、アクションは保存されます。



⑧ アクションを実行する

アクションを実行する場合は、「ファイル(F)」/「アクションウィザード(W)」から、先ほど保存した「SmartShelterPDF」を選択します。

選択後、メッセージに応じて作業を進めてください。なお、暗号化作業には、貴社のコードメータFSBが必要になりますので、ご注意ください。



13-11. コマンドラインで暗号化する方法

PDFファイルをDOSコマンドライン上で暗号化することができます。暗号化処理を自動プログラム化する際に便利な機能です。

なお、コマンドライン上で暗号化処理を行う場合、Adobe Acrobatが起動している必要があります。また、コードメータFSB(CmFSB)を装着している必要があります。

コマンドラインで暗号化するには、SmashPdfCmd.exeを使用します。SmashPdfCmd.exeは、SmashPdfAuthor.exeをインストールすると、¥ Program Files (x86) ¥ SmartShelter PDFフォルダに生成されます。

例えば、Cドライブのtestフォルダにあるsample.pdfをコマンドラインで暗号化し、同じtestフォルダの中のprotectedフォルダに保存するには、下記コマンドを実行します。

```
>SmashPdfCmd.exe -f10 -p13 -k:CM -o:c:¥test¥protected¥sample.pdf -i:c:¥test¥sample.pdf
```

-f	ファームコード
-p	プロダクトコード
-k:	コードメータを指定 (CM)
-o:	保存先のファイル名を指定
-i:	オリジナルのファイル名を指定

複数のPDFファイルを一括で暗号化する場合はアスタリスク"*.*"を使うと便利です。

(例)

¥ testフォルダにあるPDFファイルを一括で暗号化し、¥ test ¥ protectedフォルダに保存する場合は、

```
>SmashPdfCmd.exe -f10 -p13 -k:CM -o:c:¥test¥protected -i:c:¥test¥*.*
```

また、Acrobatを自動的に起動してから暗号化する場合は、"-startacrobat"を使うと便利です。

```
>SmashPdfCmd.exe -startacrobat -f10 -p13 -k:CM -o:c:¥test¥protected -i:c:¥test¥*.*
```

[NOTE]

-i: パラメータは、一番最後に記述してください。

-i: パラメータの後に記述したパラメータは無視されますので、ご注意ください。

[NOTE]

コマンドを実行する前に、Adobe Acrobatを先に起動しておく必要があります。

オプションパラメータ "-startacrobat" (後述) にて、コマンドラインからAcrobatを起動することも可能です。

[NOTE]

コマンドツールはWindows OSのみ対応しています。Mac OS上ではコマンドラインツールが使用できません。

各パラメータの説明

[必要なパラメータ]

- f <NUM> ファームコードを指定
- p <NUM> プロダクトコードを指定
- k: <Hardware> CMを指定 (CodeMeter, CodeMeterAct)
- i: <FileSpec> オリジナルファイル名を指定
 * [注意] このパラメータは一番最後に指定します。
- o: <FileSpec> 保存先のファイル名を指定

[オプションパラメータ]

- cf<Number> フィーチャーコードを指定
- ad アクティベーション日を指定
- ed 有効期限 (Expiration Date)または有効期間(Usage Period)を指定
- nw ネットワークアクセスを指定
- uco オープン時にユニットカウンタを1つ減らす
- ucp 印刷時にユニットカウンタを1つ減らす
- comp<Numer> Acrobat互換性
 - 6 Acrobat 6以降
 - 7 Acrobat 7以降

- perm:<Permissionstring> パーミッション項目を指定
 - PRINT_LOW 低解像度
 - PRINT_HIGH 高解像度
 - EDIT_INSERT ページの挿入・削除・回転
 - EDIT_FILLANDSIGN ファームフィールドの入力と署名
 - EDIT_COMMENT 注釈の作成
 - EDIT_ANY ページの抽出を除くすべての操作
 - COPY コピーを有効にする
 - ACCESSIBILITY アクセシビリティを有効にする
 - PRTSCREEN プリントスクリーンをブロックする
 - EXT_PRTSCREEN キャプチャリングソフトをブロックする
 - READER_ONLY Adobe Readerのみで開く
 - EXT_DEBUGCHECK デバッガーチェックを有効にする

- additionalssystems:<Str> システム追加の設定
 CM, FC, PC [,FeatureCode]
 [例] FC=10, PC=500, FeatureCode=1とFC=10,PC=600,FeatureCode=0を追加する場合、
 -additionalssystems:"CM,10,500,1;CM,10,600,0"

- usermessage:<str> ユーザーメッセージの指定
 拡張子なしのユーザーメッセージDLLファイル名を指定する
 [例] -usermessage:UserMsgUs

- startacrobat Acrobatが起動していない場合、Acrobatを起動させる

- clear:<Str> タスクフォルダのタスクファイルを削除する
 - ALL: 全てのタスクファイルを削除する

FINISHED: 正常に処理されたタスクファイルのみを削除する

タスクフォルダは、
<%\User Directory%\Application Data\WIBU-SYSTEMS\SmartShelter PDF
です。

13-12. 暗号化された PDF ファイルをユーザーに配布する場合

暗号化処理をしたPDFファイルを開くには、あらかじめコードメータランタイムキットとSmartShelterPDFランタイムキットをPCにインストールする必要があります。従い、ユーザーに配布する場合、下記が必要になります。

Windows OS の場合：

- ① CmStick (指定したファームコード、プロダクトコードなどが登録されたもの)
- ② コードメータランタイムキット (CodeMeterRuntime.exe など)
- ③ SmartShelterPDFランタイムキット (SmashPdfRdr.exe)
- ④ 暗号化済みPDFファイル

コードメータランタイムキットには3種類あります。

CodeMeterRuntime.exe (32bit/64bit Windows OS統合版)

CodeMeterRuntime32.exe (32bit Windows OS専用版)

CodeMeterRuntime64.exe(64bit Windows OS専用版)

上記ファイルは、コードメータDVDのCodeMeter-User¥Windowsフォルダにあります。

SmartShelterPDFランタイムキット(SmashPdfRdr.exe)は、コードメータDVDのSmartShelterPDFフォルダにあります。

Mac OS X の場合：

- ① CmStick (指定したファームコード、プロダクトコードなどが登録されたもの)
- ② コードメータランタイムキット (CmRuntimeUser_x.xx.xxxx.xxx.dmg)
- ③ SmartShelterPDFランタイムキット (SmashPdfRdr_11.15.1699.dmg)
- ④ 暗号化済みPDFファイル

コードメータランタイムキット(CmRuntimeUser_x.xx.xxxx.xxx.dmg)は、コードメータDVDのCodeMeter-User¥MacOSフォルダにあります。Mac OS Xのバージョンによって使い分けてください。

(「Chapter 14 ユーザーに配布する場合」参照)

SmashPdfRdr_11.15.1699.dmgは、コードメータDVDのSmartShelterPDFフォルダにあります。

CmStickおよび暗号化されたPDFファイルはWindows/Mac共通で使用できます。また、各コードメータ関連ファイルを貴社のCD/DVDに入れて配布しても、ライセンス上問題ありません。

また、最新のファイルは弊社サイトからダウンロードできます。

<http://www.suncarla.co.jp/download/>

[NOTE]

SmartShelterPDF用ファイルには、2種類のプログラムがあります。(Windowsの場合)

1つは、開発用 (Author用) プログラム "SmashPdfAuthor.exe"

もう1つは、ユーザー用 (Reader用) プログラム "SmashPdfRdr.exe"

です。ユーザーに配布するのは、ユーザー用プログラム "SmashPdfRdr.exe" になります。

動作環境 (OS)

Windows 7/8/8.1/10 (32bit/64bit)

Windows Server 2008(R2)/2012(R2)/2016 (32bit/64bit)

Mac OS X 10.9, 10.10, 10.11, 10.12, 10.13, 10.14

Adobe Acrobat 9/X/XI/DC/2017 (32bit/64bit) または Adobe Reader 9/X/XI/DC (32bit/64bit)

(暗号化されたPDFファイルは、Adobe AcrobatとAdobe Readerの両方で開きます。)

Windows 用 SmartShelterPDF ランタイムキット "SmashPdfRdr.exe" について

ユーザーに配布するSmashPdfRdr.exeには3種類のインストール方法があります。

標準

SmartShelterPDFプラグインソフト "SmashPDFRdr.api"をインストールします。インストール先は、Adobe AcrobatおよびAdobe Readerすべてが対象になります。通常はこの「標準」をお使いください。

(例)

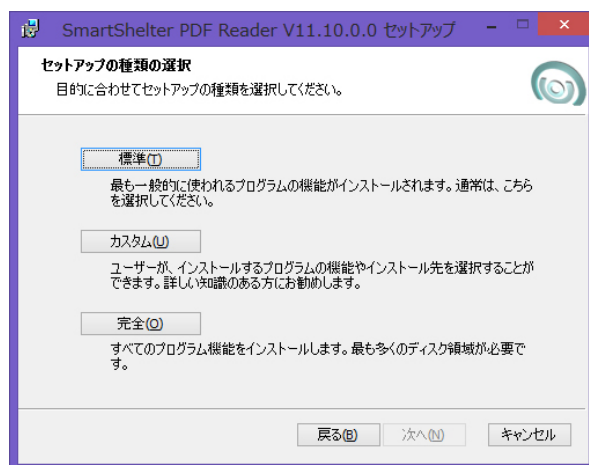
1台のPCにAdobe AcrobatとAdobe Readerの2つが存在している場合、プラグインソフト "SmashPDFRdr.api"はAdobe AcrobatとAdobe Readerそれぞれに自動インストールされます。

カスタム

SmartShelterPDFプラグインソフト "SmashPDFRdr.api"のインストール先を選択できます。「標準」では、すべてのAcrobat/Readerにプラグインソフトがインストールされますが、「カスタム」ではインストール先を選択できます。

完了

現在のところ、「標準」と同じ動作になります。



Mac OS X 用 SmartShelterPDF ランタイムキット "SmashPdfRdr_11.15.1699.dmg" について

Mac OS X用SmartShelterPDFランタイムキットをインストール方法は次のようになります。

① "SmashPdfRdr_11.15.1699.dmg" をダブルクリックする

Mac上で"SmashPdfRdr_11.15.1699.dmg"をダブルクリックします。



② SmartShelterPDF のインストール画面で「続ける」ボタンをクリックする

SmartShelterPDFのインストール画面で「続ける」ボタンをクリックします。



③ 「インストール」ボタンをクリックする

「インストール」ボタンをクリックするとインストールが開始されます。



13-13. 起動時に保護モードを無効化する方法 (Adobe Reader DC/XI)

SmartShelterPDFで暗号化されたPDFファイルをWindows OS上のAdobe Reader XI/DCで開く場合は、セキュリティ上の理由から、サンドボックスによる保護の「起動時に保護モードを有効にする(M)」のチェックをはずす必要があります。このオプションが有効の状態ではSmartShelterPDFで暗号化されたPDFファイルを開くとエラーになります。これは、Adobe Reader XI/DCの保護モードにより、SmartShelterPDF機能の一部が使用できなくなる理由からです。なお、Adobe Reader XI/DC(WindowsOS)以外は保護モードを無効化する必要がありません。

下記環境では、保護モードを無効化する必要がありません。

Windows: Adobe Acrobat 9/X/XI/DC/2017および Adobe Reader 9/X

Max OS X: Adobe Acrobat 9/X/XI/DC および Adobe Reader 9/X/XI/DC

[NOTE]

保護モードを無効化する環境は、Windows上のAdobe Reader XI/DCのみです。

Adobe Reader DC (Windows) の場合

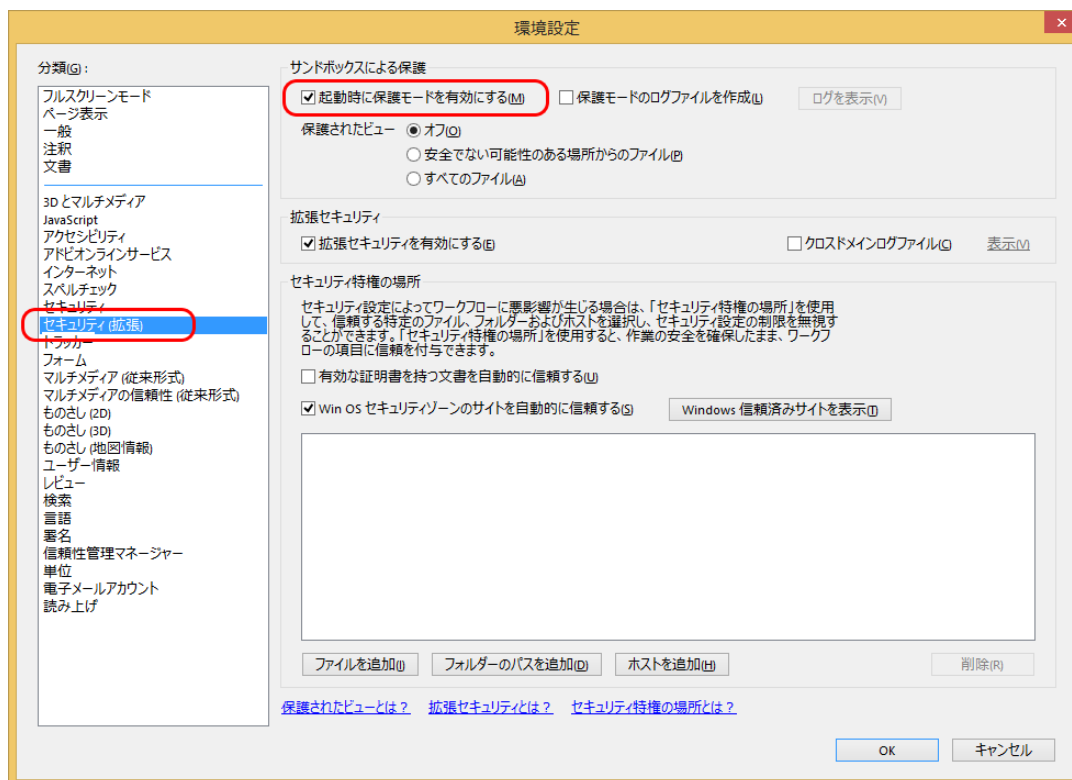
13-13-1. Adobe Reader DC の「環境設定」を開く

Adobe Reader DCを起動し、「編集(E)」メニューの「環境設定(N)」をクリックします。

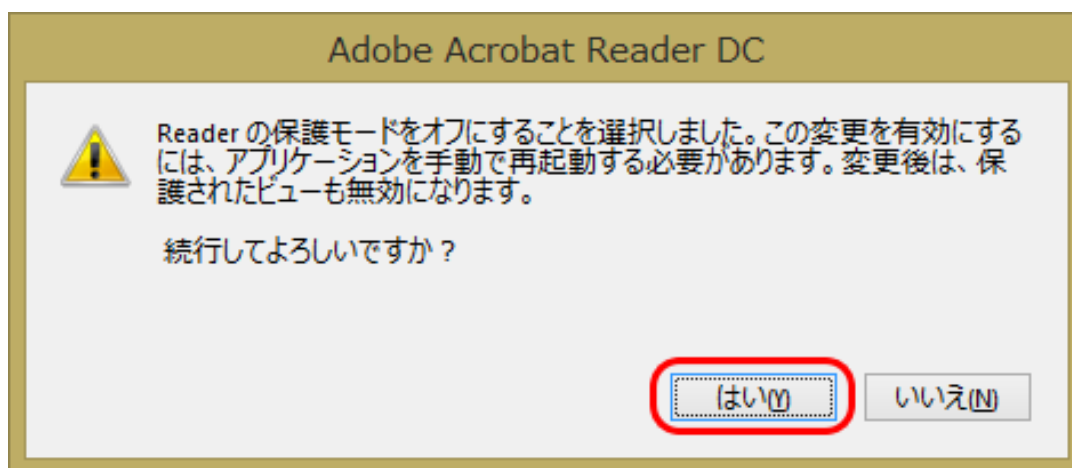


13-13-2. 「起動時に保護モードを有効にする (M)」のチェックをはずす

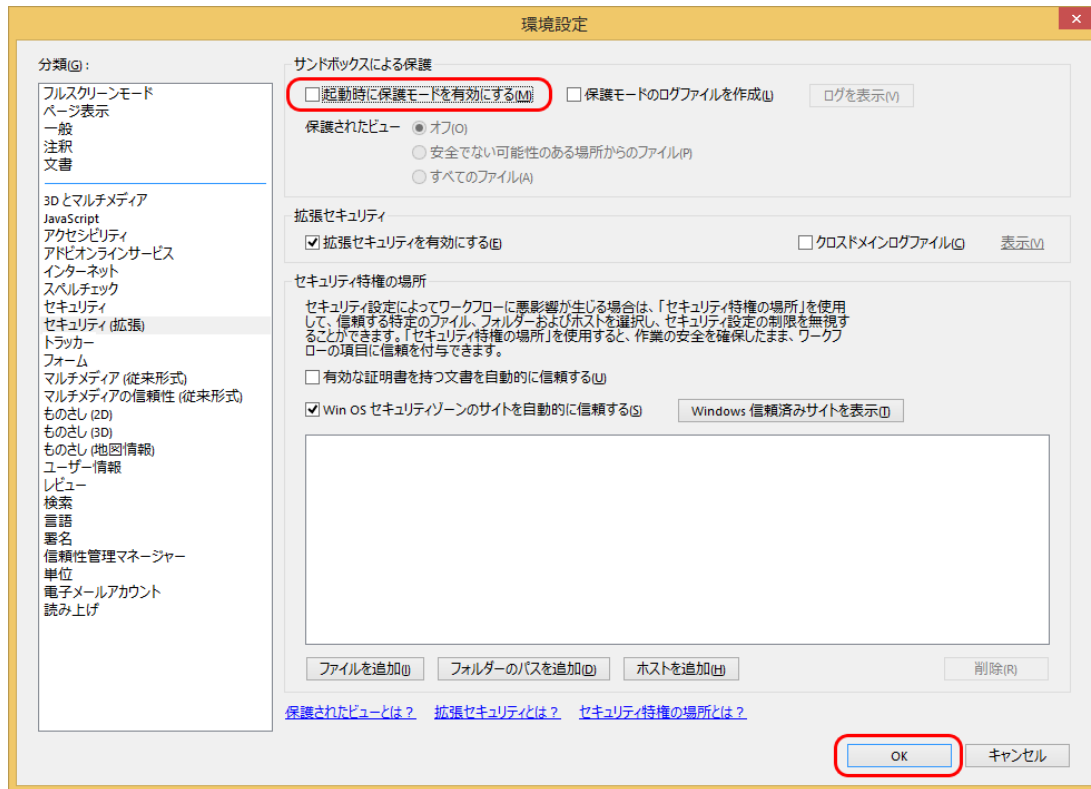
「環境設定」画面の「分類」項目から「セキュリティ[拡張]」を選択し、サンドボックスによる保護の「起動時に保護モードを有効にする(M)」のチェックをはずします。



チェックをはずすと下記メッセージが表示されますので、「はい(Y)」ボタンをクリックします。

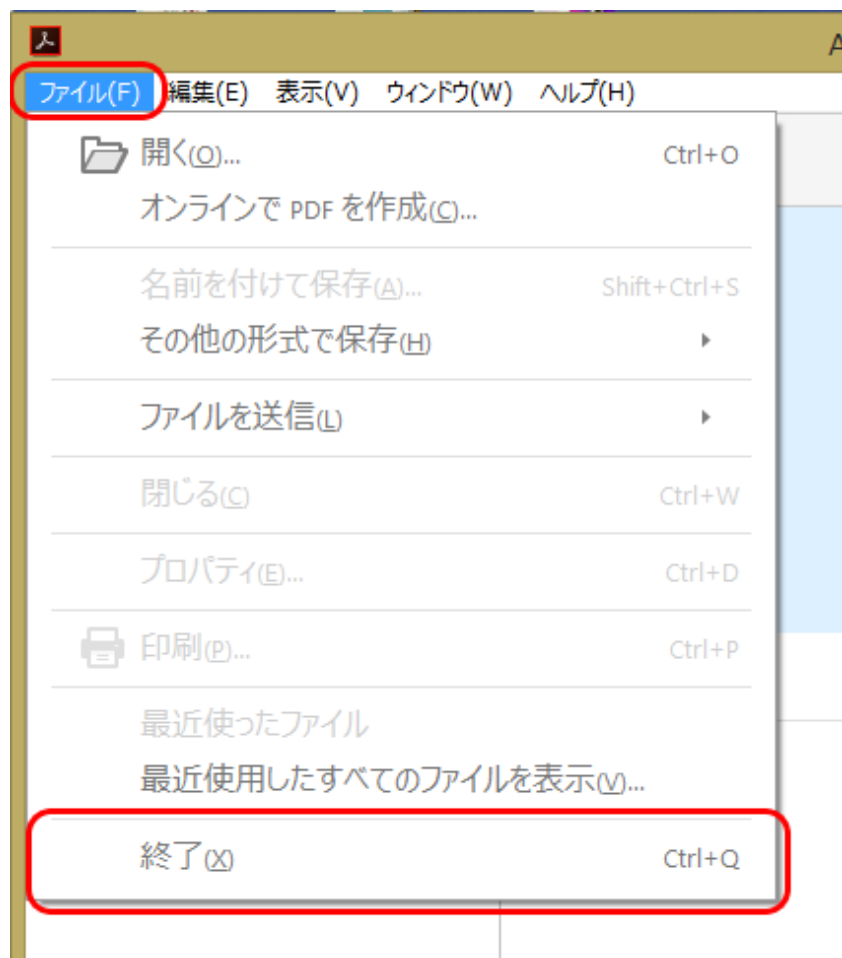


サンドボックスによる保護の「起動時に保護モードを有効にする(M)」のチェックがはずれたことを確認し、右下の「OK」ボタンをクリックします。



13-13-3. Adobe Reader DC を一度終了させる。

「ファイル(F)」メニューから「終了(X)」をクリックし、一度、Adobe Reader DCを終了させます。



Adobe Reader DC終了後、再度、PDFを開いてください。

[NOTE]

暗号化されたPDFファイルを開くには、正しいコードメータCmDongleまたはCmActLicenseがPCに装着されている必要があります。

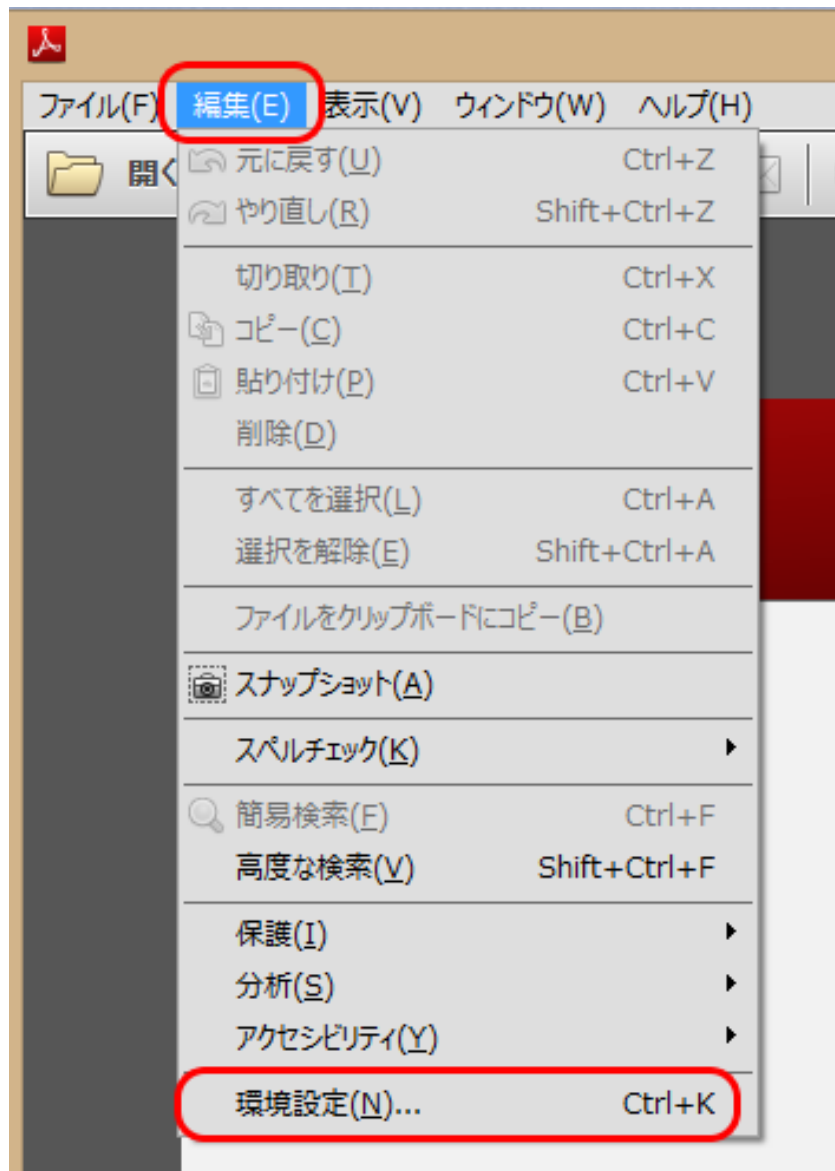
[NOTE]

SmartShelterPDFで暗号化されたPDFを開くために、「起動時に保護モードを有効にする(M)」のチェックをはずしましたが、閲覧終了後は、必要に応じ、保護モードを有効にしてください。

Adobe Reader XI (Windows) の場合

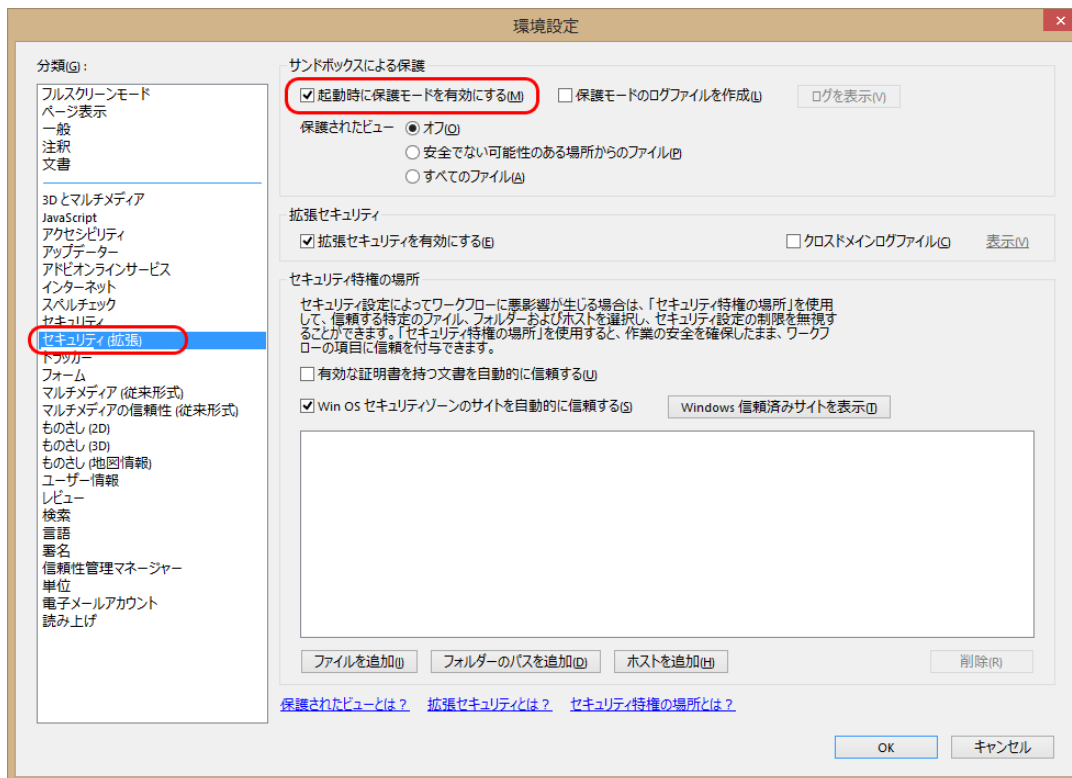
13-13-4. Adobe Reader XI の「環境設定」を開く

Adobe Reader XIを起動し、「編集(E)」メニューの「環境設定(N)」をクリックします。

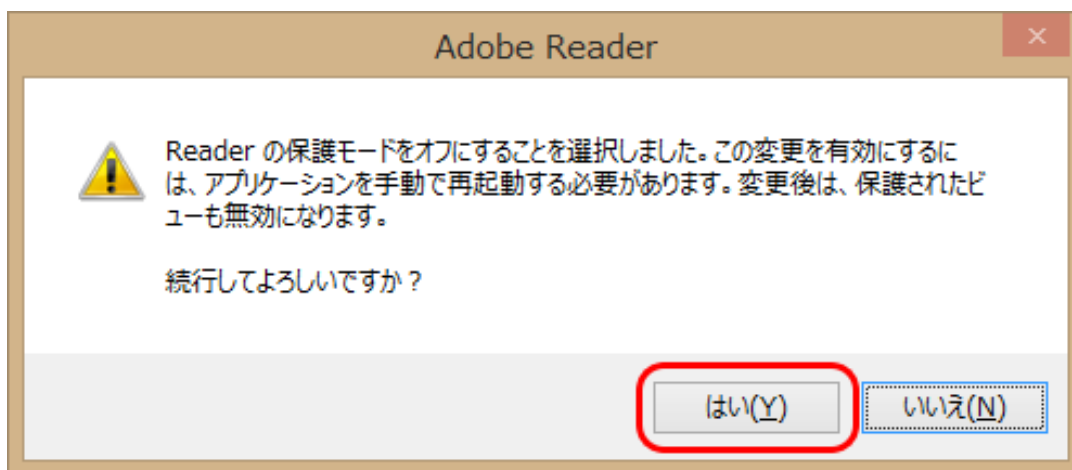


13-13-5. 「起動時に保護モードを有効にする (M)」 のチェックをはずす

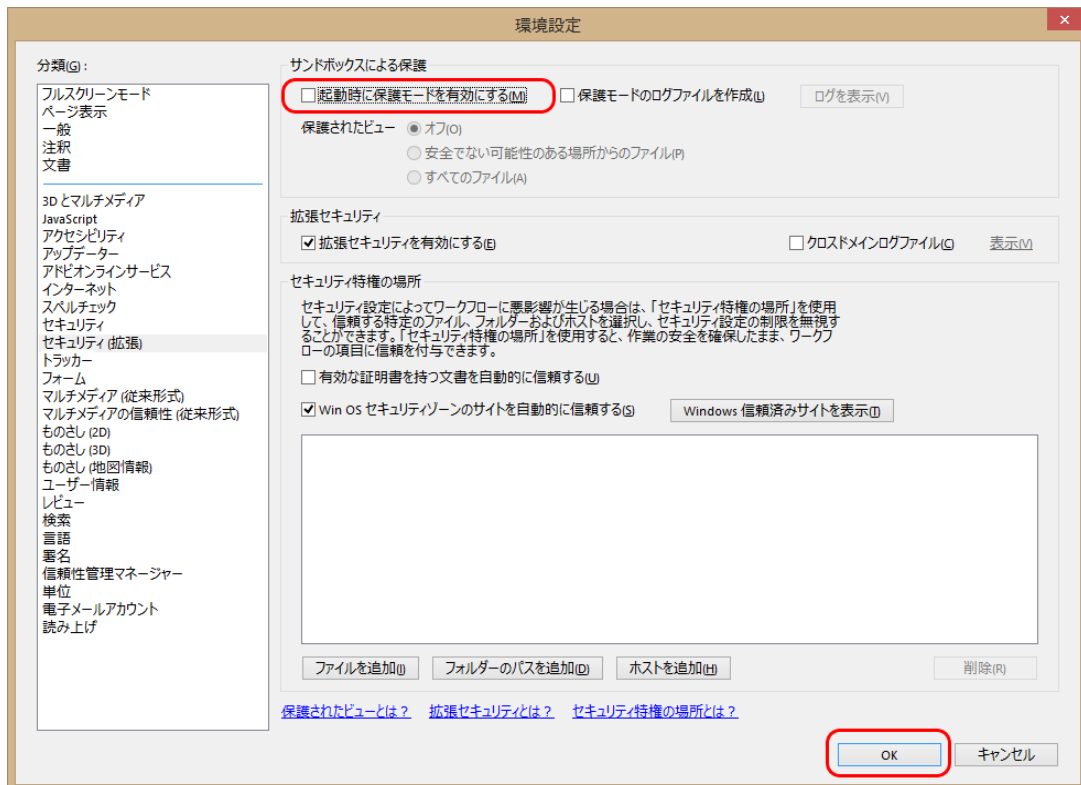
「環境設定」画面の「分類」項目から「セキュリティ[拡張]」を選択し、サンドボックスによる保護の「起動時に保護モードを有効にする(M)」のチェックをはずします。



チェックをはずすと下記メッセージが表示されますので、「はい(Y)」ボタンをクリックします。

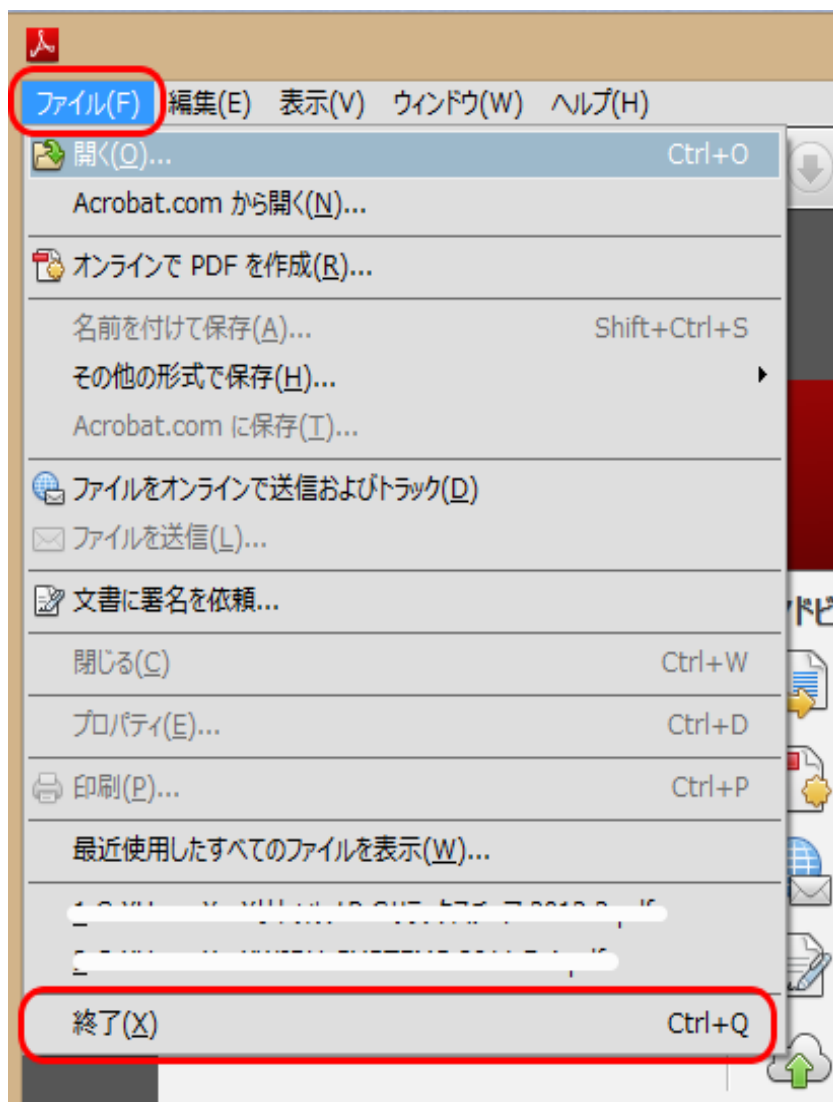


サンドボックスによる保護の「起動時に保護モードを有効にする(M)」のチェックがはずれたことを確認し、右下の「OK」ボタンをクリックします。



13-13-6. Adobe Reader XI を一度終了させる。

「ファイル(F)」メニューから「終了(X)」をクリックし、一度、Adobe Reader XIを終了させます。



Adobe Reader XI終了後、再度、PDFを開いてください。

[NOTE]

暗号化されたPDFファイルを開くには、正しいコードメータCmDongleまたはCmActLicenseがPCに装着されている必要があります。

[NOTE]

SmartShelterPDFで暗号化されたPDFを開くために、「起動時に保護モードを有効にする(M)」のチェックをはずしましたが、閲覧終了後は、必要に応じ、保護モードを有効にしてください。

Chapter 14

ユーザーに配布する場合

- 14-1. ユーザーに配布する場合
- 14-2. Windows アプリケーション / .NET アプリケーションを配布する
- 14-3. 暗号化された PDF ファイルを配布する (SmartShelterPDF)
- 14-4. Mac OS X アプリケーションを配布する
- 14-5. Linux アプリケーションを配布する
- 14-6. Sun Solaris アプリケーションを配布する

14-1. ユーザーに配布する場合

コードメータで暗号化したプログラムやコンテンツファイルを起動するには、あらかじめコードメータランタイムキットをPCにインストールする必要があります。インストールするコードメータランタイムキットは、使用するOS環境により異なります。また、暗号化方法により、コードメータランタイムキット以外のファイルも一緒にインストールする必要があります。それぞれの使用環境によって、ユーザーに配布するファイルを確認してください。なお、コードメータに関連するファイルは、貴社のアプリケーションと一緒に配布してもライセンス上問題ありません。

コードメータランタイムキットは、コードメータDVDのRuntimeフォルダに格納されています。

14-2. Windows アプリケーション /.NET アプリケーションを配布する

32bit版/64bit版のWindowsアプリケーションまたは.NETアプリケーションを配布する場合はコードメータランタイムキットのインストールが必要になります。Windows版コードメータランタイムキットには3種類あります。

CodeMeterRuntime.exe (32bit/64bit Windows OS統合版)

CodeMeterRuntime32.exe (32bit Windows OS専用版)

CodeMeterRuntime64.exe (64bit Windows OS専用版)

32bit/64bit 統合版の場合、対象になるOSが32bitか64bitをチェックし、そのOSに応じたランタイムをインストールします。上記ファイルは、コードメータDVDのCodeMeter-User¥Windowsフォルダに存在します。

対応OS:

Windows 7/8/8.1/10 (32bit/64bit)

Windows Server 2008(R2)/2012(R2)/2016 (32bit/64bit)

[NOTE]

C++/CLIで作成した混在コードの場合でも、インストール内容は同じです。

14-3. 暗号化された PDF ファイルを配布する (SmartShelterPDF)

SmartShelterPDFで暗号化されたPDFファイルを配布する場合は、コードメータランタイムキットの他に、SmartShelterPDFランタイムキット"SmashPDFRdr.exe"が必要になります。

Windows OS(32bit/64bit) の場合

- ① コードメータランタイムキット
- ② SmartShelterPDFランタイムキット "SmashPDFRdr.exe"

コードメータランタイムキットには3種類あります。

CodeMeterRuntime.exe (32bit/64bit Windows OS統合版)

CodeMeterRuntime32.exe (32bit Windows OS専用版)

CodeMeterRuntime64.exe (64bit Windows OS専用版)

上記ファイルは、コードメータDVDのCodeMeter-User¥Windowsフォルダにあります。

SmartShelterPDFランタイムキット(SmashPdfRdr.exe)は、コードメータDVDのCodeMeter-SDK¥SmartShelterPDF¥Winフォルダにあります。

Mac OS X の場合：

- ① コードメータランタイムキット (CmRuntimeUser_6.80.3312.500.dmg)
- ② SmartShelterPDFランタイムキット (SmashPdfRdr_11.15.1699.dmg)

コードメータランタイムキット(CmRuntimeUser_6.80.3312.500.dmg)は、コードメータDVDのCodeMeter-User¥MacOSフォルダにあります。SmashPdfRdr_11.15.1699.dmgは、コードメータDVDのCodeMeter-SDK¥SmartShelterPDF¥Macフォルダにあります。

SmartShelterPDF 動作環境

対応OS:

Windows 7/8/8.1/10 (32bit/64bit),

Windows Server 2008(R2)/2012(R2)/2016 (32bit/64bit)

Mac OS X 10.9/10.10/10.11/10.12/10.13/10.14

Adobe Acrobat 9/X/XI/DC/2017 または Adobe Reader 9/X/XI/DC

14-4. Mac OS X アプリケーションを配布する

Mac OS X 10.4以降のアプリケーションを配布する場合は、Mac用コードメータランタイムキットをインストールします。

Mac OS X 10.9 - 10.14

① Mac OS X用コードメータランタイムキット"CmRuntimeUser_6.80.3312.500.dmg"

Mac OS X 10.8

① Mac OS X用コードメータランタイムキット"CmRuntimeUser_6.10.2018.501.dmg"

Mac OS X 10.6 - 10.7

① Mac OS X用コードメータランタイムキット"CmRuntimeUser_5.22.1514.502.dmg"

Mac OS X 10.5

① Mac OS X用コードメータランタイムキット"CmRuntimeUser_4.50.909.504.dmg"

Mac OS X 10.4

① Mac OS X用コードメータランタイムキット"CmRuntimeUser_4.30.498.504.dmg"

14-5. Linux アプリケーションを配布する

RPM package for SuSe, Red Hat etc

32bit:

CodeMeter-6.80.3312-500.i386.rpm

CodeMeter-lite-6.80.3312-500.i386.rpm

64bit:

CodeMeter-6.80.3312-500.x86_64.rpm

CodeMeter-lite-6.80.3312-500.x86_64.rpm

DEB package for Debian, Ubuntu etc

32bit:

codemeter_6.80.3312.500_i386.deb

codemeter-lite_6.80.3312.500_i386.deb

64bit:

codemeter_6.80.3312.500_amd64.deb

codemeter-lite_6.80.3312.500_amd64.deb

codemeter-lite_6.80.3312.500_armhf.def

サンカーラ株式会社

〒103-0012 東京都中央区日本橋堀留町1-6-5 丸彦ビル3F

TEL: 03-3249-3421 / Fax:03-3249-3444

E-mail: support@suncarla.co.jp

www.suncarla.co.jp