

CodeMeter コードメータ

ユーザーズガイド Ver. 4.50c

No. 5007

2012 年 12 月 21 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/C 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.....	35
3-6. 使用有効期限（Expiration Time）を設定したプロテクトを行う.....	37
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.....	39
3-8. 使用期間（Usage Period）を設定したプロテクトを行う.....	41
3-9. メンテナンス期間を設定したプロテクトを行う.....	43
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.....	46
Chapter 4 自動暗号化ツール AxProtector について	47
4-1. 自動暗号化ツール AxProtector について.....	48
4-2. 日本語モードにする.....	50
4-3. AxProtector のメニュー画面.....	51
4-4. AxProtector の各入力画面の説明.....	52
4-5. データファイルを暗号化する.....	86
4-6. コマンドラインでの使用方法.....	88
Chapter 5 lxProtector/WUPI について	93
5-1. lxProtector とは.....	94
5-2. WUPI ファンクションについて.....	95
5-3. WUPI ファンクション一覧.....	96
5-4. WUPI ファンクションの使い方.....	98
5-5. WUPI ファンクション詳細.....	102

Chapter 6 CodeMeter Core API について 111

6-1. CodeMeter Core API.....	112
6-2. サンプルプログラムについて	112
6-3. CodeMeter API ガイドの使い方	113
6-4. CodeMeter API ガイドの使用例	114
6-5. CodeMeter Core API 一覧	117

Chapter 7 ネットワーク機能について 123

7-1. ネットワークライセンス管理とは	124
7-2. ネットワークカウンターの登録方法.....	125
7-3. コードメータサーバーの起動方法	128
7-4. ネットワーク対応型プロテクトの作成方法.....	130

Chapter 8 リモートアップデート機能について 133

8-1. リモートアップデート機能とは.....	134
8-2. リモートアップデート作業の流れ	135
8-3. ライセンス要求ファイルの作成（ユーザー側）	136
8-4. ライセンス更新ファイルの作成（貴社側）	139
8-5. CM-Stick を更新する（ユーザー側）	146

Chapter 9 コードメータ ライセンスエディタの使い方 149

9-1. コードメータ ライセンスエディタについて	150
9-2. ストラクチャとナビゲーション	150

Chapter 10 CmBoxPgm の使い方 157

10-1. CmBoxPgm について	158
10-2. コード登録の流れ.....	158
10-3. ファームコード (Firm Code) を登録する	159
10-4. プロダクトコード (Product Code) を登録する.....	160
10-5. プロダクトコード (Product Code) を削除する.....	161
10-6. 使用有効期限 (Expiration Time) を登録する.....	162
10-7. 各パラメータの説明	163

Chapter 11 コードメータ コントロールセンターの使い方 169

11-1. コードメータコントロールセンターとは	170
11-2. ストラクチャとナビゲーション	171
11-3. メニューバー	172
11-4. 「ライセンス」タブ	175
11-5. 「イベント」タブ	178
11-6. 「ライセンス貸出」タブ	179
11-7. ライセンス貸出・返却の方法	180
11-8. ライセンス貸出の有効期限について	185

Chapter 12 Web アドミン (WebAdmin) の使い方 187

12-1. コードメータ Web アドミン (WebAdmin) について	188
12-2. ホーム	190
12-3. 内容	192
12-4. 構成	195

Chapter 13 Adobe PDF ファイルにプロテクトをかける 203

13-1. Adobe PDF ファイルにプロテクトをかける	204
13-2. 作業に必要なもの	205
13-3. コードメータ開発ツールをインストールする	205
13-4. SmartShelter PDFAuthor Ver6.00 をインストールする	206
13-5. sample.pdf を暗号化する	208
13-6. CM-Stick にコードを登録する	211
13-7. 動作を確認する	211
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	212
13-9. エラーメッセージをカスタマイズする	214
13-10. プレインテキストパスワードについて	217
13-11. 複数の PDF ファイルを一括して暗号化する	218
13-12. コマンドラインで暗号化する方法	225
13-13. 暗号化された PDF ファイルをユーザーに配布する場合	227

Chapter 14 ユーザーに配布する場合 235

14-1. ユーザーに配布する場合	236
14-2. Windows アプリケーション (32bit 版) を配布する	236
14-3. Windows アプリケーション (64bit 版) を配布する	236
14-4. .NET アプリケーション (32bit/64bit 版) を配布する	237
14-5. 暗号化された PDF ファイルを配布する	237
14-6. Mac OS X アプリケーションを配布する	237
14-7. Linux アプリケーションを配布する	238
14-8. Sun Solaris アプリケーションを配布する	238

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、高機能コピープロテクトツール「コードメータ」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」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コピープロテクトツールです。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を防止するために必要な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を復号化して実行し、モジュール終了と同時に暗号化してメモリー上に展開するという「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. lxProtector/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。lxProtector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. コンテンツ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。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(CM-Stick)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個まで登録可能になりました。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(CM-Stick)で一元管理できます。

6. 貴社専用の CM-FSB (CodeMeter Firm Security Box)

コードメータキー(CM-Stick)にライセンスコードを登録するために必要なCM-FSB(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、拡張メモリー、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CM-Stick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CM-Stickにライセンスを貸出することで、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CM-Stick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CM-Stick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(CM-Stick)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. CmlIdentity (Web 認証) 機能

Webサーバーへのログイン管理が行える本格的なWeb認証機能"CmlIdentity"（シーエムアイデンティティ）を搭載。従来のIDとパスワードによるログイン認証と比べ、セキュリティが一段と向上します。特定Webサイトへのログイン管理、SaaS/ASPなどのユーザー認証に非常に効果的です。

14. CM-BOX タイマー機能

コードメータキー(CM-Stick)の中に、時刻を確実に刻むBOX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。

15. RoHS 指令対応済み

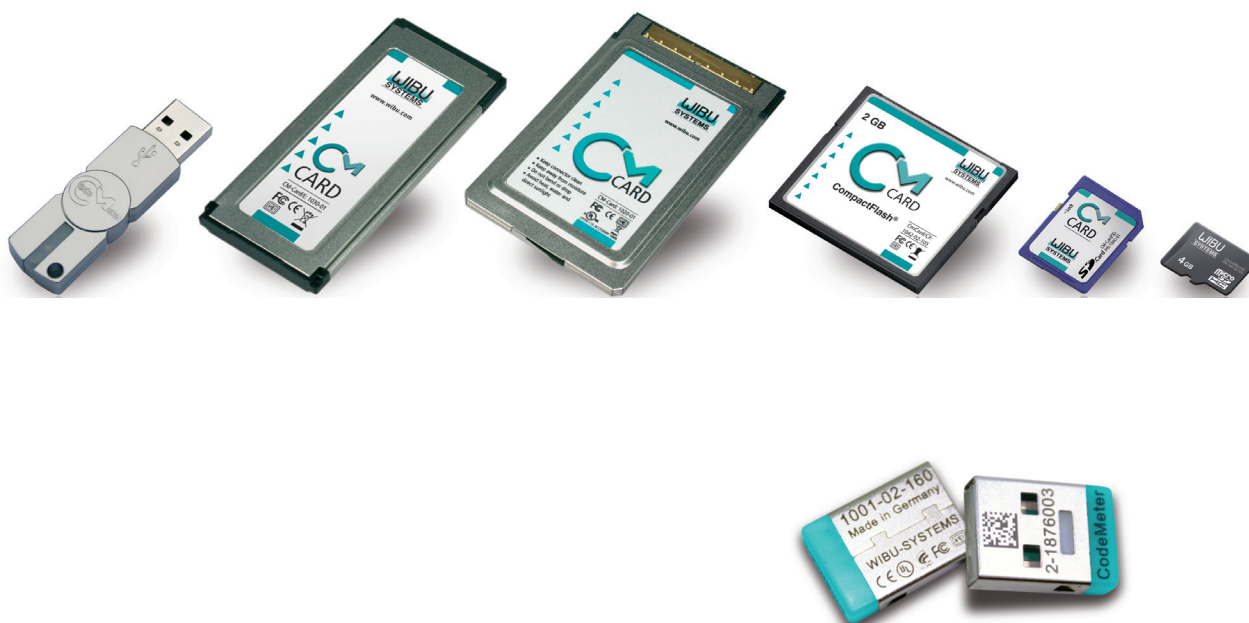
コードメータは、RoHS指令で規制されている鉛、水銀、カドミウム、六価クロム、ポリ臭化ビフェニール、ポリ臭化ジフェニルエーテルの使用基準を満たしているWEEE/RoHS指令適合製品です。

16. ISO9001:2008 認証取得

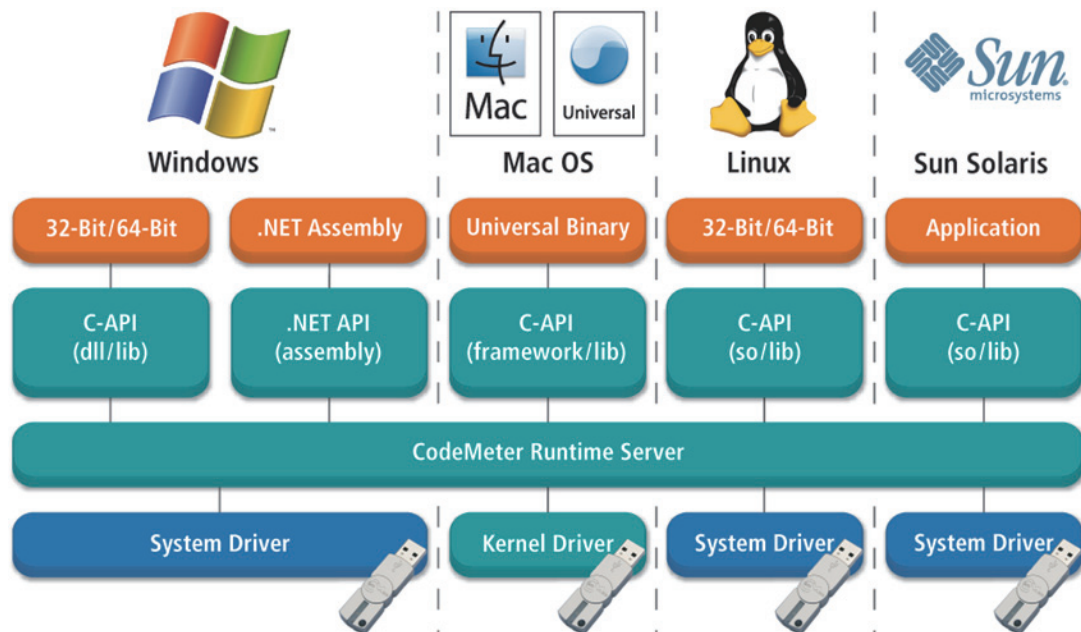
コードメータは、ISO9001:2008認証されたWIBU-SYSTEMS社(ドイツ)で開発・製造されており、高い品質と信頼性を実現した商品です。

17. 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。

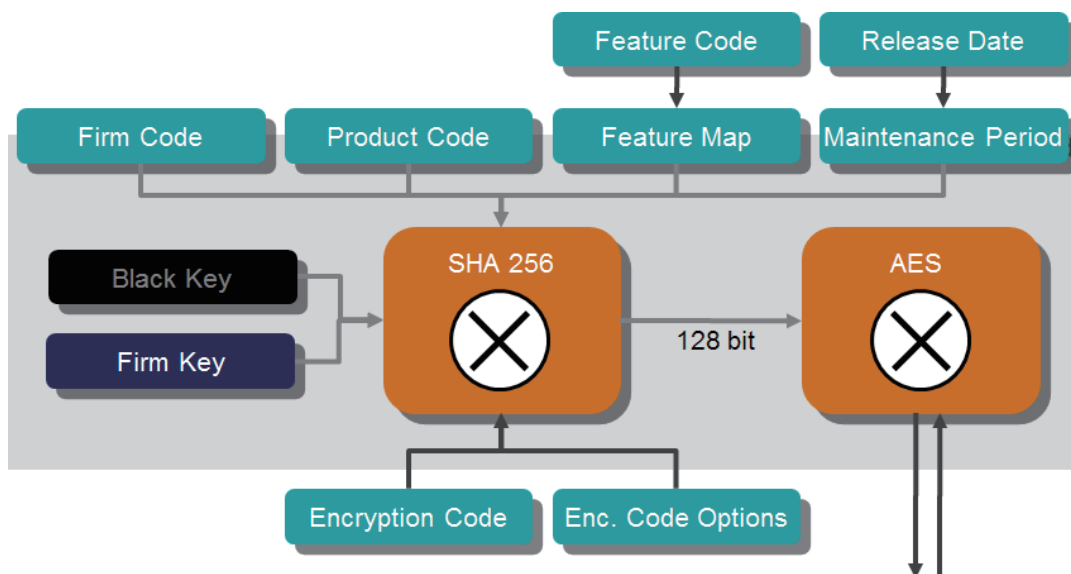


1-3. コードメータ・アーキテクチャ



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をコピーする

[NOTE]

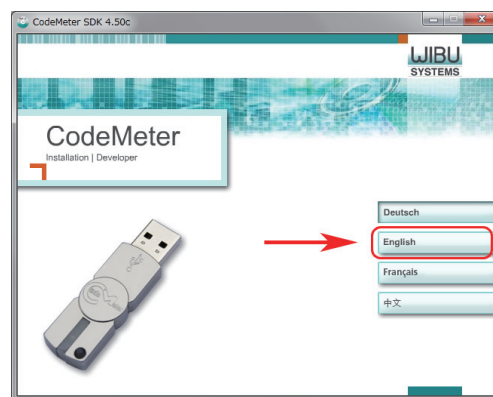
コードメータ開発キットをインストールするには、あらかじめMicrosoft .NET Framework 2.0以降がインストールされている必要があります。インストールされていない場合は、コードメータ開発キットをインストールする前に、Microsoft .NET Framework 2.0以降をインストールしてください。

2-2. CodeMeter SDK をインストールする

①コードメータ CD を起動する

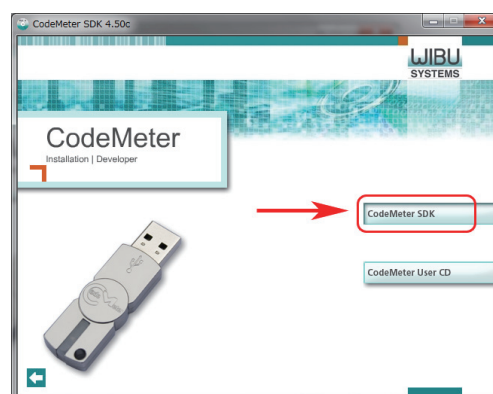
コードメータCDをCD/DVDドライブに挿入すると右の画面が表示されます。表示されない場合はコードメータCDのルートにある"CDStart.exe"をダブルクリックして起動します。

選択する言語が表示されますので、"English"をクリックします。



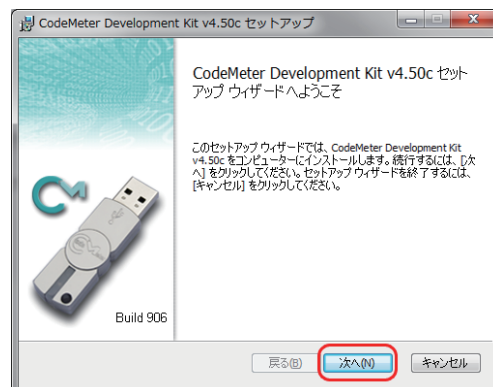
② CodeMeter SDK をクリックする

"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。



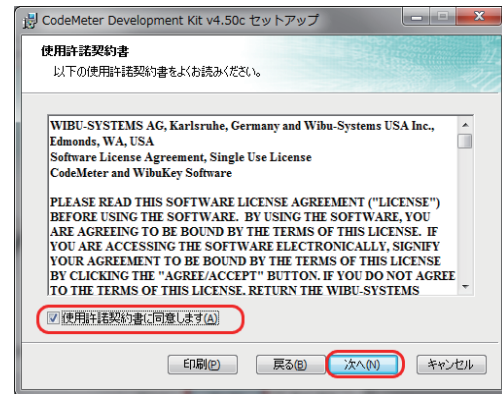
③ CodeMeter Development Kit v4.50c をインストールする

「CodeMeter Development Kit v4.50c 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。



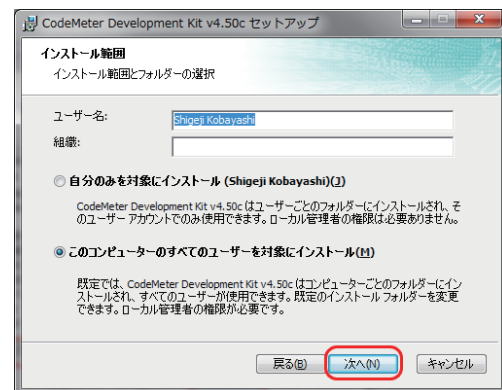
④ライセンス契約書

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。



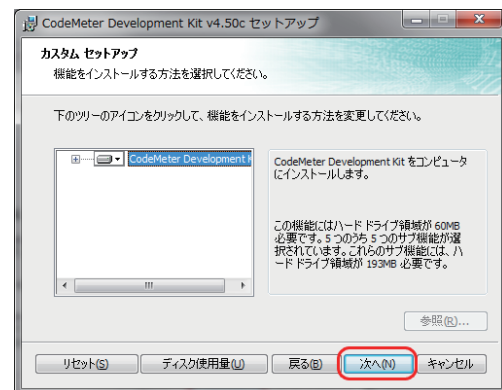
⑤ユーザー情報を入力する

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。



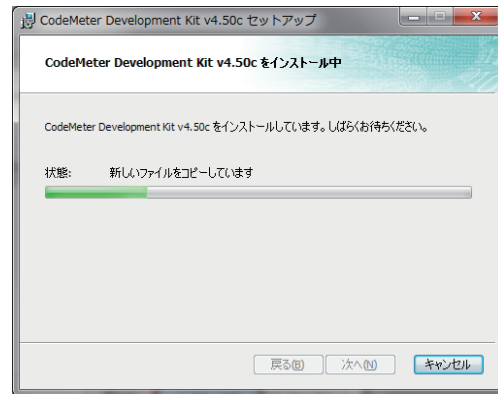
⑦インストールを開始する

インストールを開始する準備ができましたら、「次へ」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。



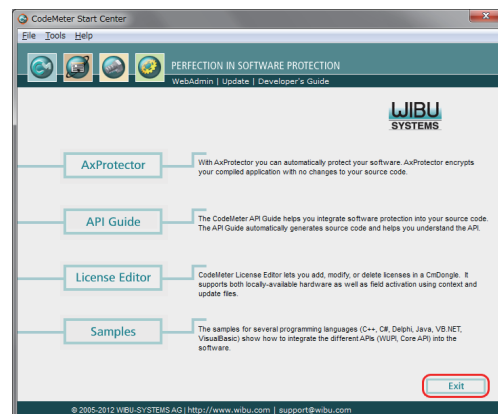
⑨正常にインストールされた

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。

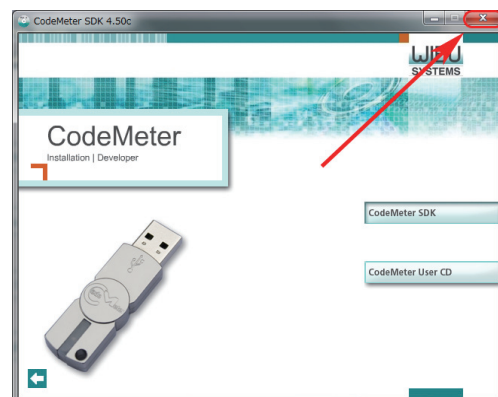


⑪最後に「閉じるボタン」をクリックする

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「2-4. 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がインポートされていますので、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。

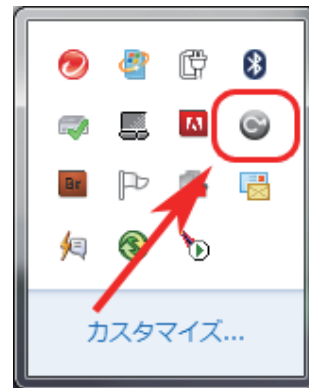
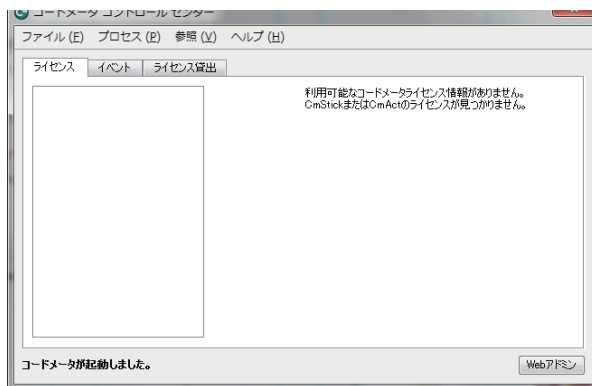


2-3. 貴社のライセンスファイル CmFirm.wbc をインポートする

コードメータFSB(CM-FSB)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。

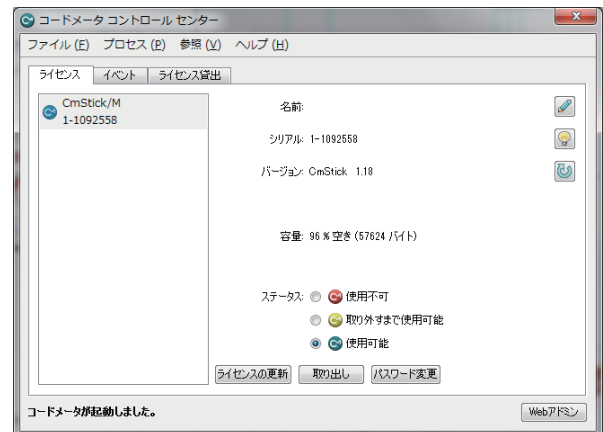
① コードメータコントロールセンターを開く

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。



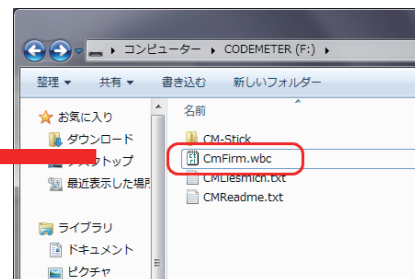
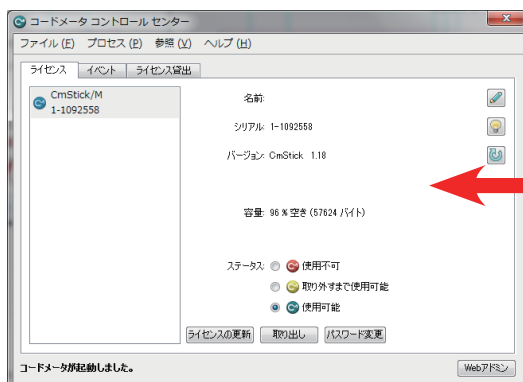
② コードメータ FSB を PC に装着する

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



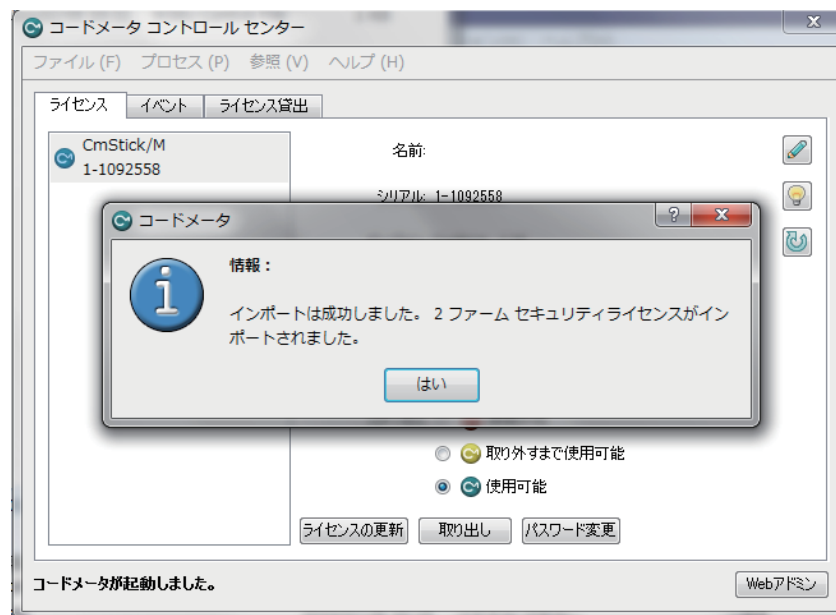
③ CmFirm.wbc をドラッグ&ドロップする

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. CM-Stick 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

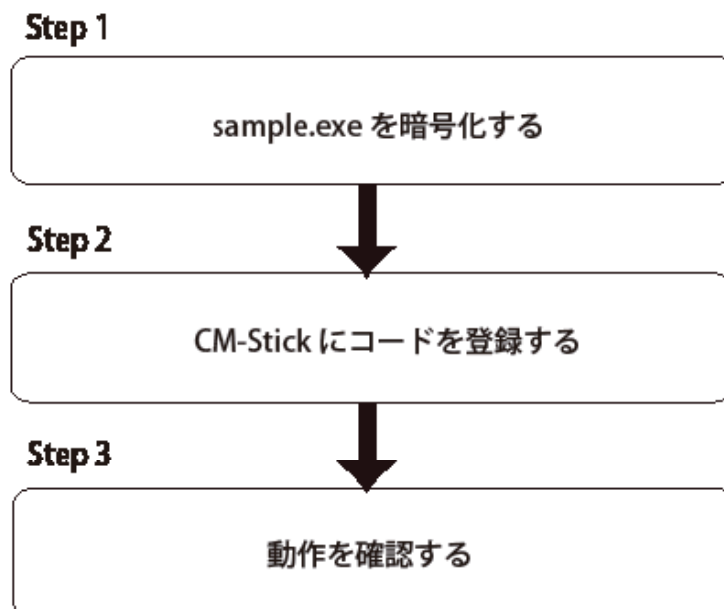
3-1. 実行形式プログラムにプロテクトをかける

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CD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CD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/v450c/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。

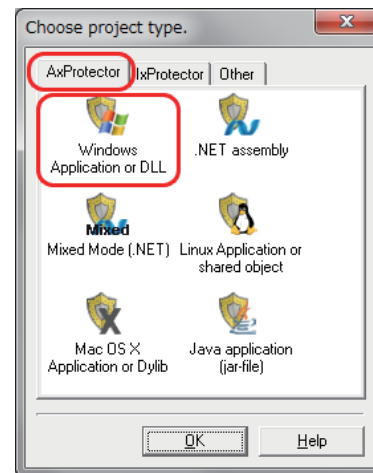
コードメータCDのToolsフォルダからsample.exeを任意のフォルダにコピーします。ここでは、testフォルダを作成し、そこにコピーします。

0. 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。

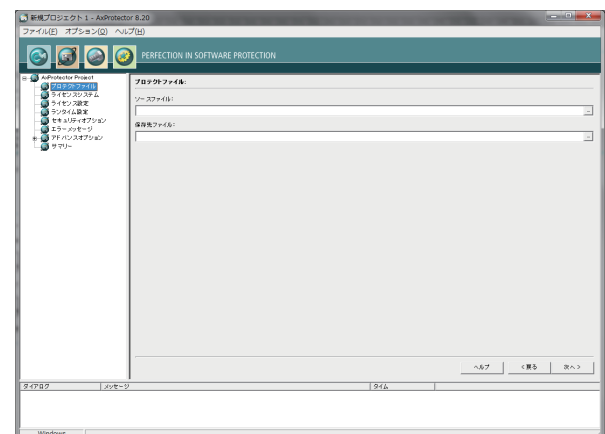
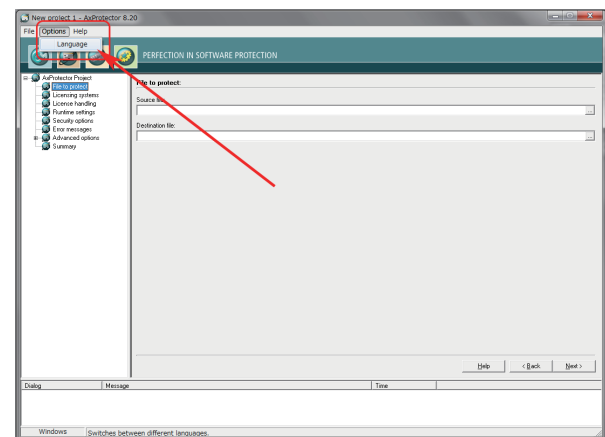
まず、コードメータFSB(CmFSB)をPCのUSBポートに装着し、【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】から「AxProtector」を起動し、「Windows 32-bit exe or dll」を選択し、OKボタンをクリックします。

"AxProtectorGui.exe"は、インストール先の
¥Program Files¥WIBU-SYSTEMS¥AxProtector
¥DevKit¥binフォルダの中にあります。



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。

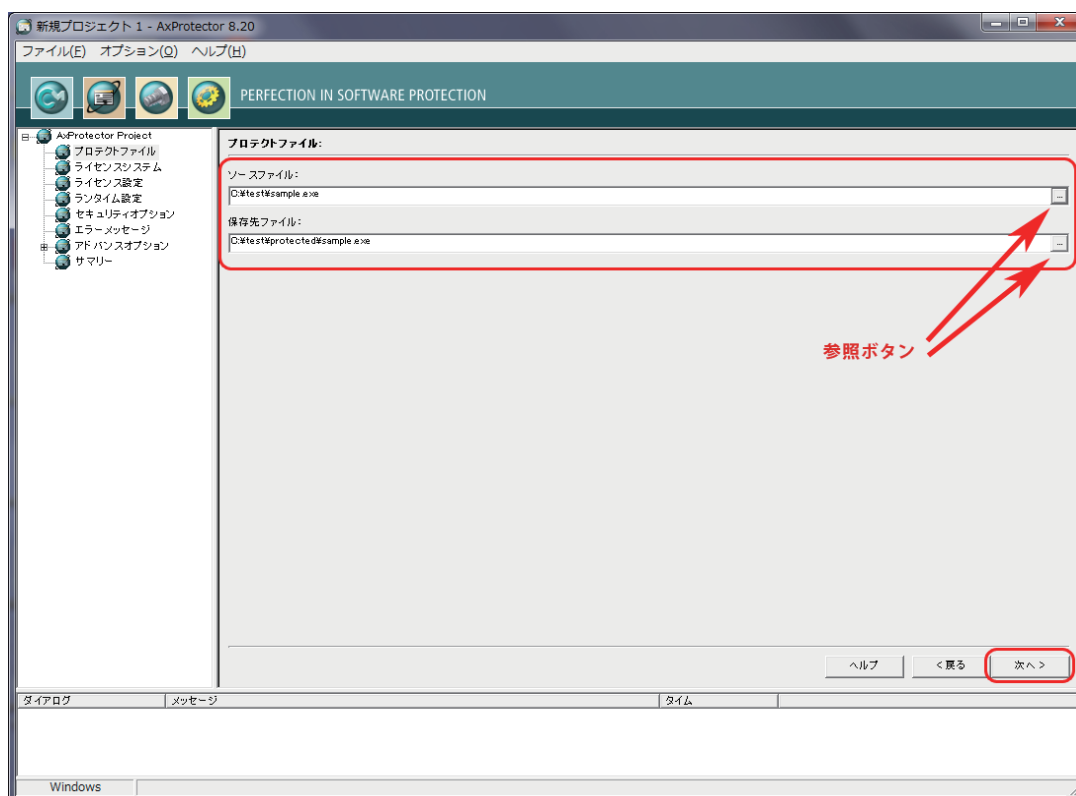


1. プロテクトファイル：

「1. 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」にプロテクト処理後に作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]

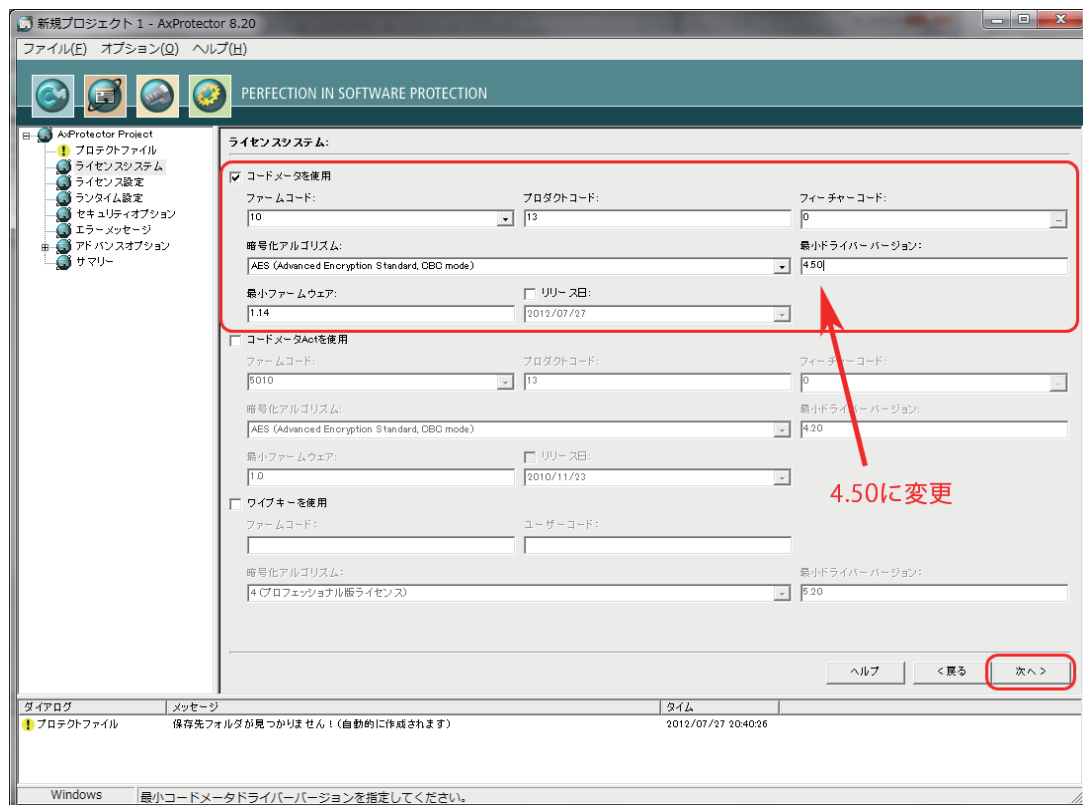
オリジナルファイルを上書きするのを防ぐために、必ず保存先ファイル名はソースファイル名と別名にするか、または異なるパス名（異なるフォルダ）にしてください。ソースファイルを指定すると、同一フォルダにprotectedフォルダが自動的に作成されます。



2. ライセンスシステム：

「2. ライセンスシステム」画面で、ハードウェアキーの指定を行います。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CmStick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とプロダクトコード等をベースに生成された独自の暗号キー（ランダム値）で暗号化されます。

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(CmDongle)だけを使用し、ファームコード=10、プロダクトコード=13 を設定します。最小ドライババージョンは"4.50"を設定します。(4.50に書き換える)それ以外の、フィーチャマップ、暗号化アルゴリズム、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]

コードメータ、コードメータAct、ワイブキーすべてを選択してプログラムを暗号化すると、コードメータ、コードメータAct、ワイブキーのどれかが存在するとプログラムが起動します。また、コードメータ、ワイブキーの2つを選択すると、コードメータかワイブキーのどちらかが存在するとプログラムが起動します。この機能をうまく利用することで、ワイブキーからコードメータへの移行をスムーズに行うことが可能になります。

[NOTE]

ファームコードは貴社専用の6桁コード(整数値)になり、コードメータ開発キット(CmFSB)を購入した時点で割り当てられ変更することはできません。貴社独自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。

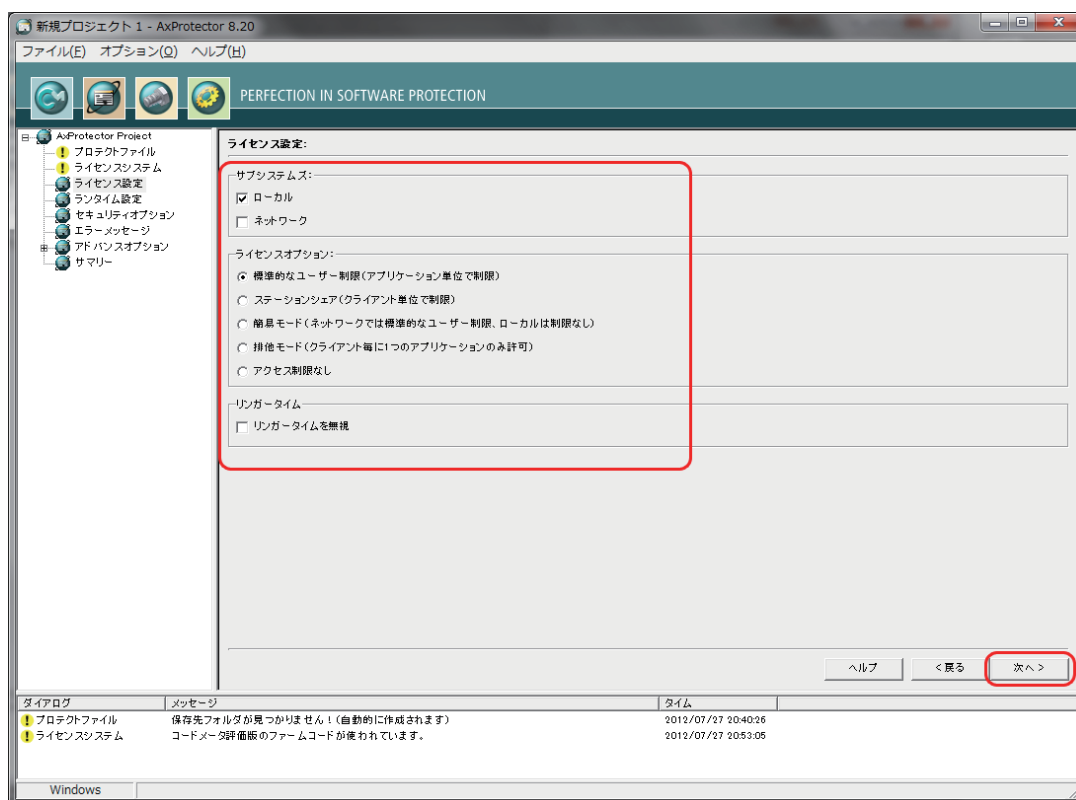
3. ライセンス設定：

「3. ライセンス設定」画面でライセンス体系を決めます。暗号化されたプログラムが、ローカルPC上のコードメータキーだけを検索するか、ネットワーク上(LAN上)のコードメータキーを検索するかの設定をします。ここでは、ローカルPC上のコードメータキーを検索しますので、「サブシステムズ」の「ローカル」にチェックを入れます。(デフォルトの状態)

ここで、「ネットワーク」にもチェックを入れると、暗号化されたプログラムがローカルPC上でコードメータキーを見つけられなかった場合、ネットワーク上(LAN上)にあるコードメータキーを検索し始めます。検索の結果、コードメータサーバーに指定したコードメータキーが見つかり暗号化されたプログラムが起動します。この「ネットワーク」オプションは、コードメータネットワーク機能を使ったライセンス管理(同時アクセス数など)に役立ちます。

「ライセンスオプション」は、デフォルトの「標準的なユーザー制限(アプリケーション単位で制限)」に設定します。

「リンガータイム」はチェックをはずしておきます。リンガータイム機能は、ネットワーク機能と一緒に使用する機能で、プロテクトされたプログラムが終了するとライセンス割り当てが解放されますが、解放を直ちに実行するのではなく、ある程度の時間(リンガータイム)を置いて解放したい場合に使用します。これは、プログラムの再起動などで再度ライセンス割り当てが必要な場合に、他ユーザーがそのライセンスを使用することを一時的にブロックするためのものです。(再起動による再割り当てを確保するため)



設定が終了したら、「次へ」ボタンをクリックして次に進みます。

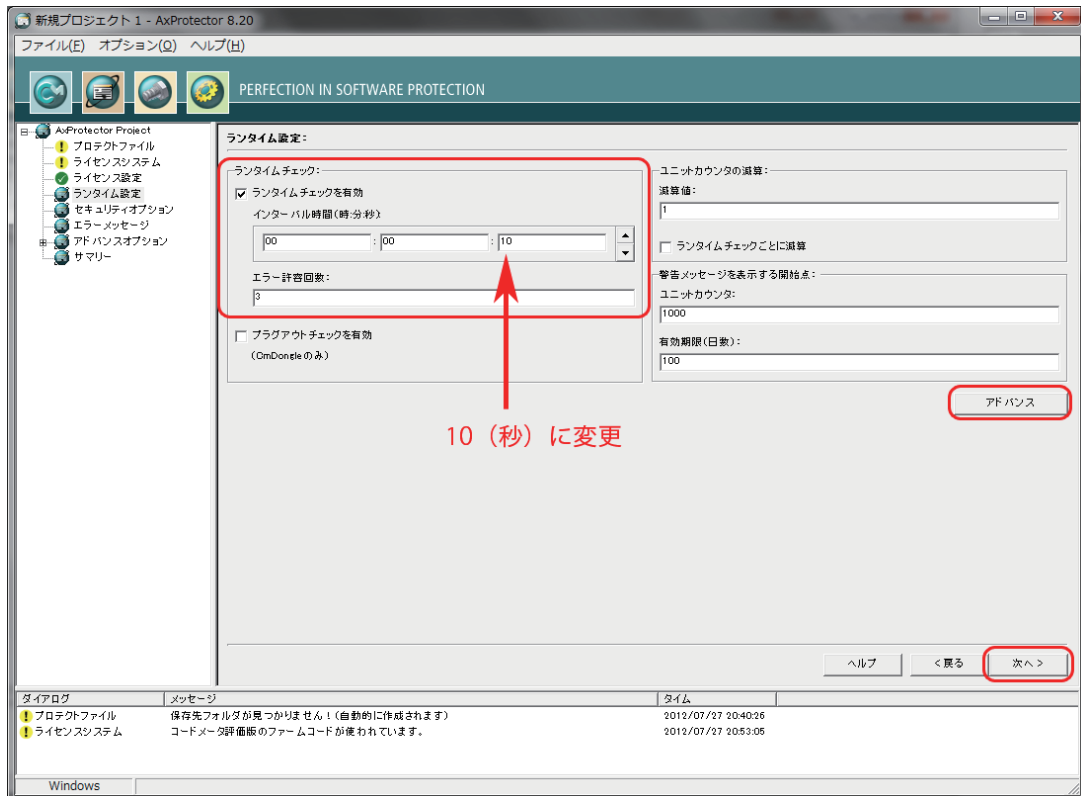
[NOTE]

ライセンスオプション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/ 4-4 . AxProtectorの各入力画面の説明/ 3. ライセンス設定」をご参照ください。

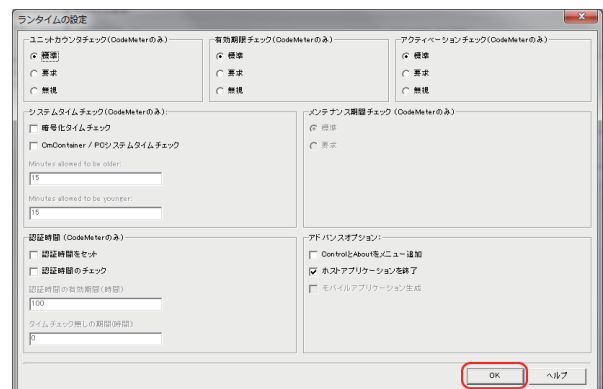
4. ランタイム設定：

「4. 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の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。



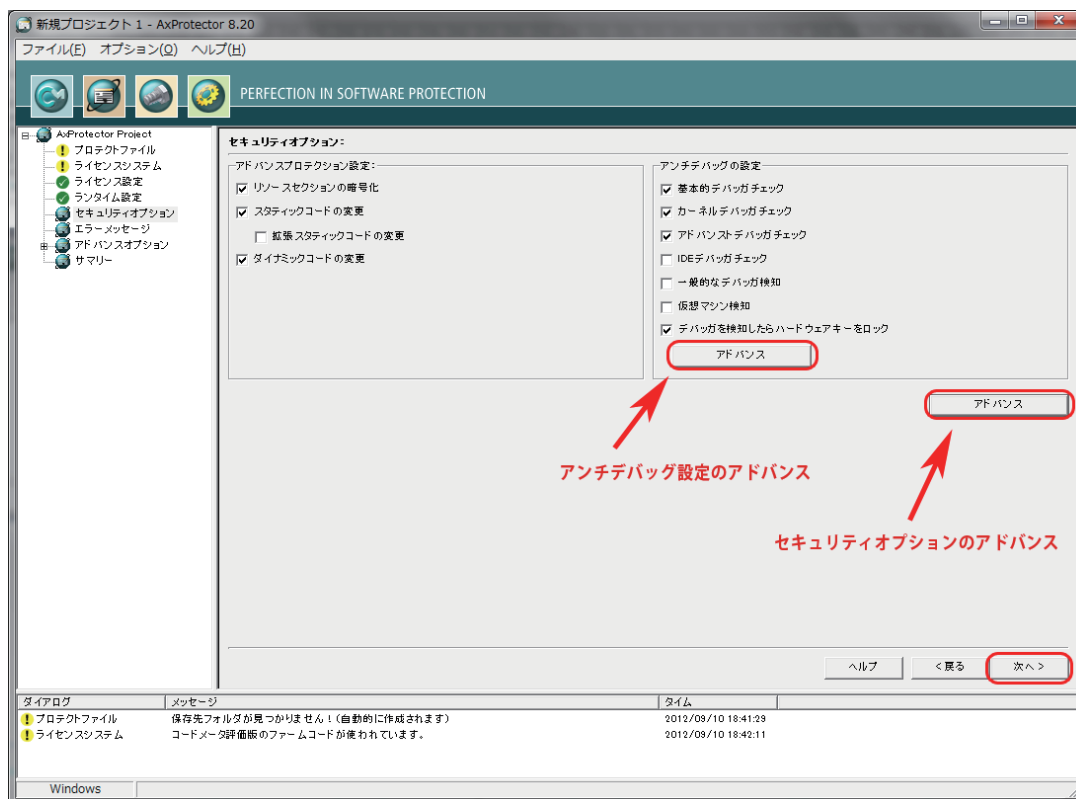
また、「アドバンス」ボタンをクリックすると、さらにきめの細かいランタイム設定を行うことができます。ここでは、特に設定しませんので、「OK」ボタンをクリックして閉じます。



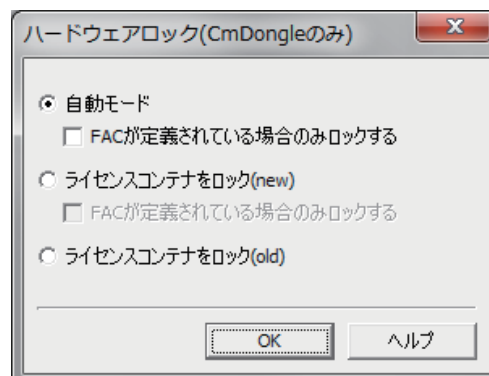
設定が終了したら、「次へ」ボタンをクリックして次に進みます。

5. セキュリティオプション：

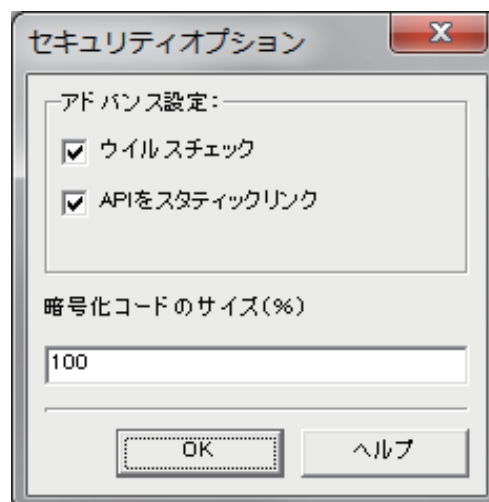
「5. セキュリティオプション」画面で、セキュリティ強度の設定を行います。暗号化の方法や、デバッグ解析に対するセキュリティポリシーを決めます。AxProtectorは、デフォルトの設定で十分強力な暗号化を実現できるため、ここではデフォルトの状態にします。



また、アンチデバッグ設定のアドバンスボタンをクリックすると、ハードウェアロック (CmDongle) のロックモードの選択画面が表示されます。これは、解析行為の中でデバッグを検知したらどのようにハードウェア (CmDongle) 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、ウイルス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。「5. 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。

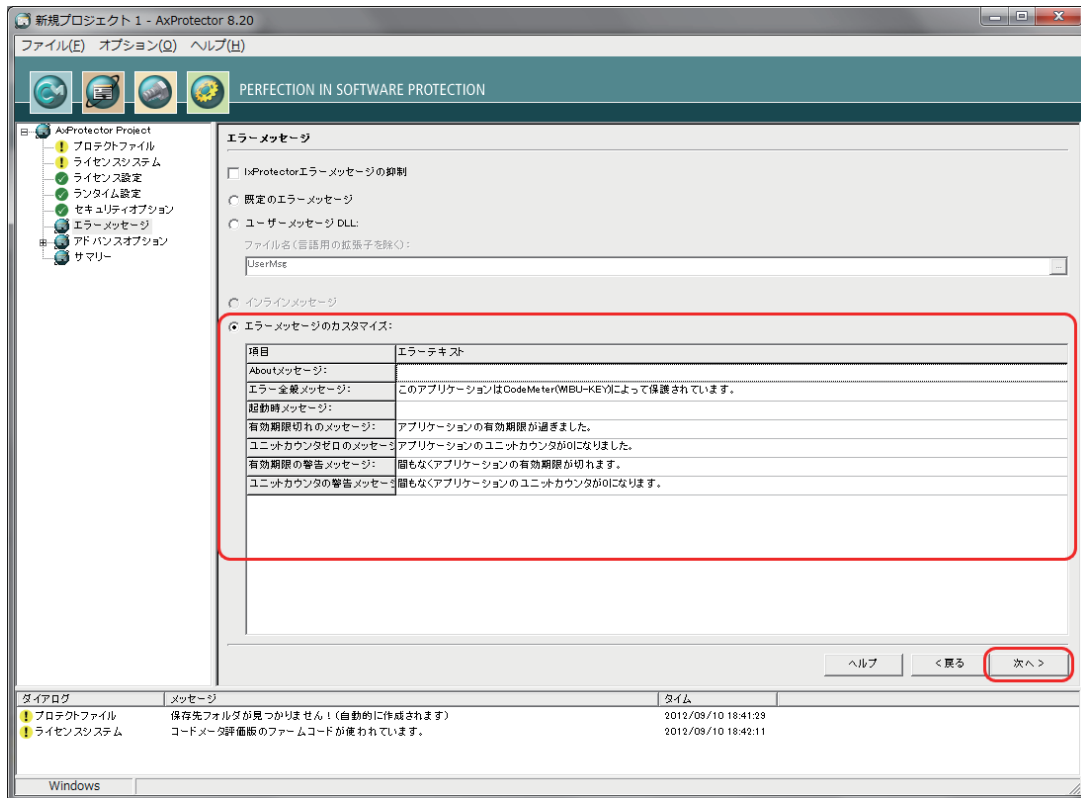


6. エラーメッセージ：

「6. 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。

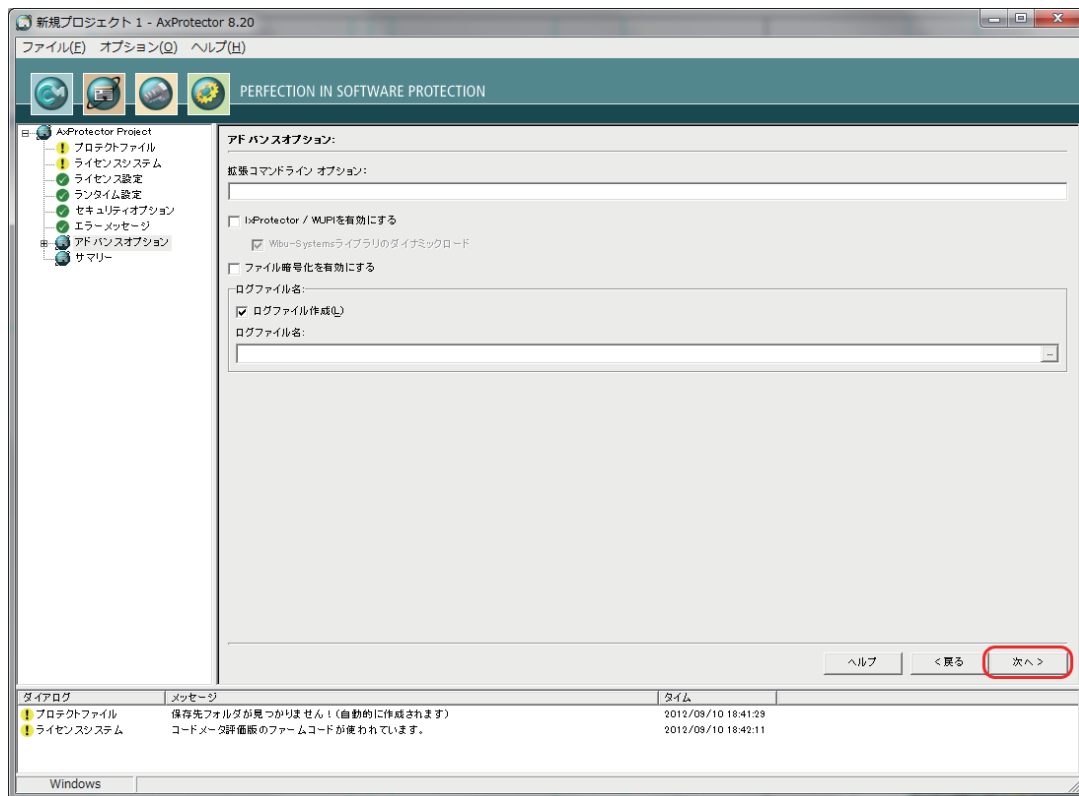
[NOTE]

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/ 4-4 . AxProtectorの各入力画面の説明/ 6. エラーメッセージ」をご参照ください。

7. アドバンスオプション：

lxProtector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化し、実行したあとは再び暗号化しておくという、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できます。

lxProtector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、lxProtectorは使用しませんのでチェックをはずしておきます。

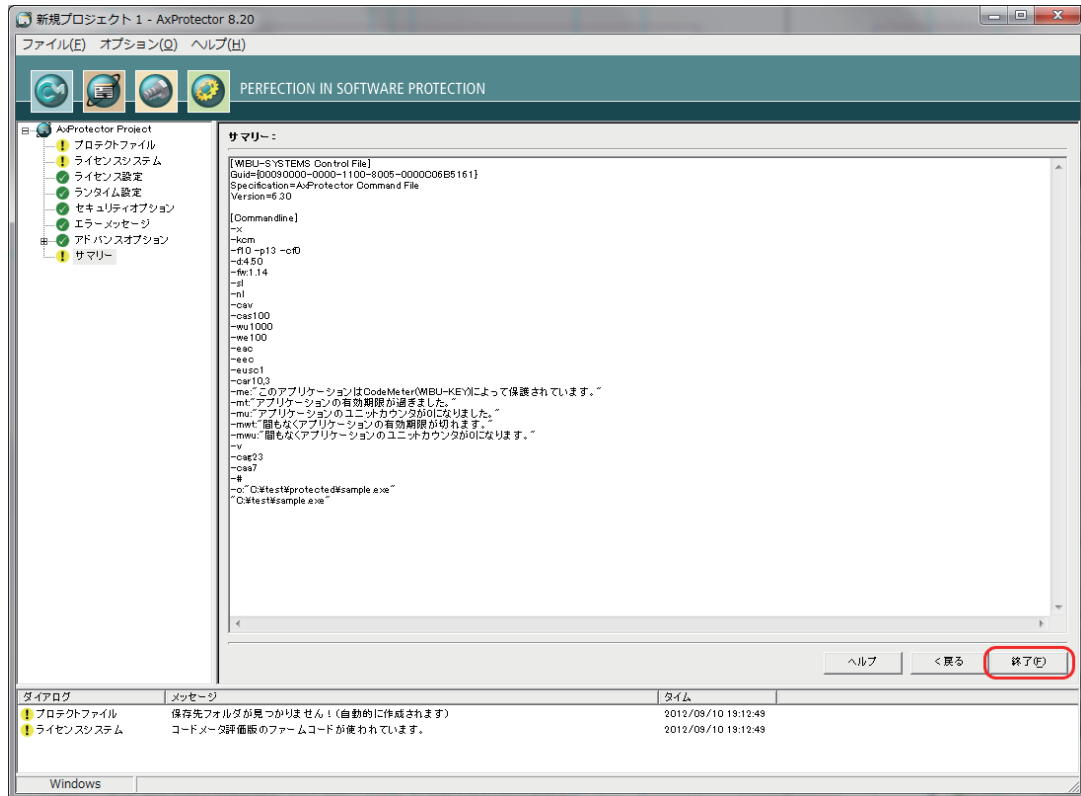


[NOTE]

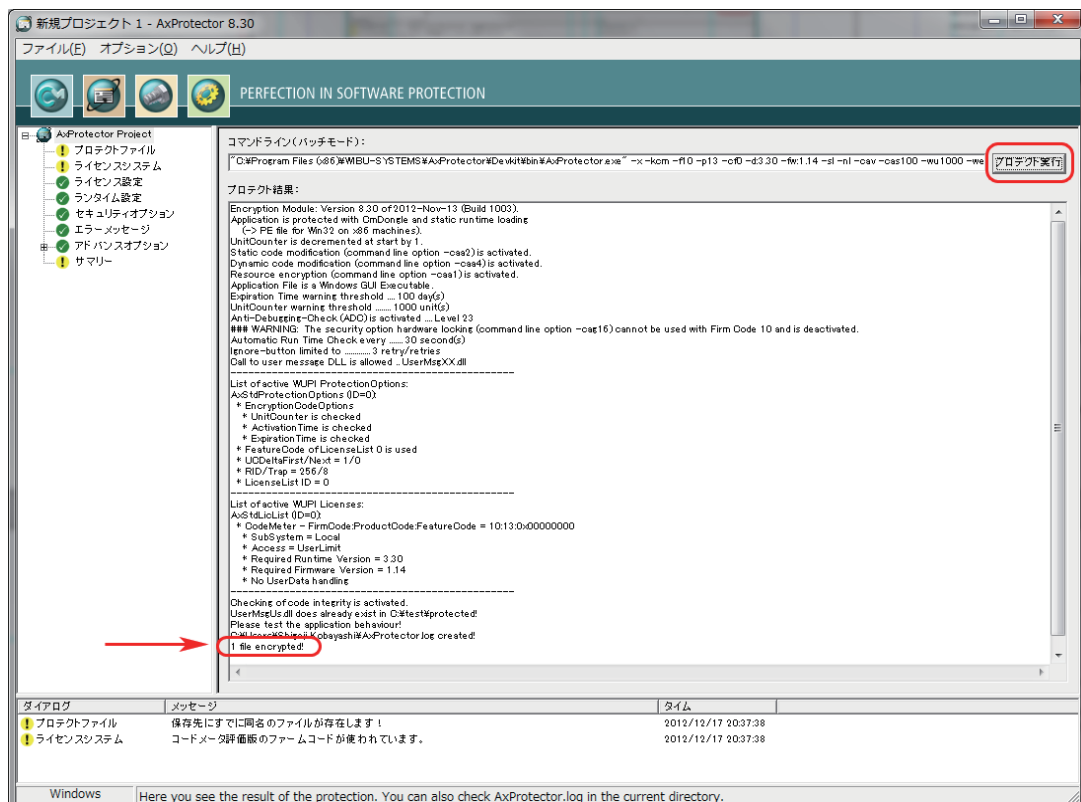
C#やVB.NETなどで作成したMicrosoft .NET Framework対応プログラム(マネージコード)の場合は、lxProtector/WUPIを使用しなくても、AxProtector (.NETアセンブリ)で暗号化することで自動的に「オンデマンド復号」機能が組み込まれます。「オンデマンド復号」を実現するために、あえてソースコードを修正する必要がありません。lxProtector/WUPIをソースコードに組み込む必要があるのは、VC++やVB6などのネイティブコード(アンマネージコード)の場合です。

8. サマリー（プロテクト内容を確認する）：

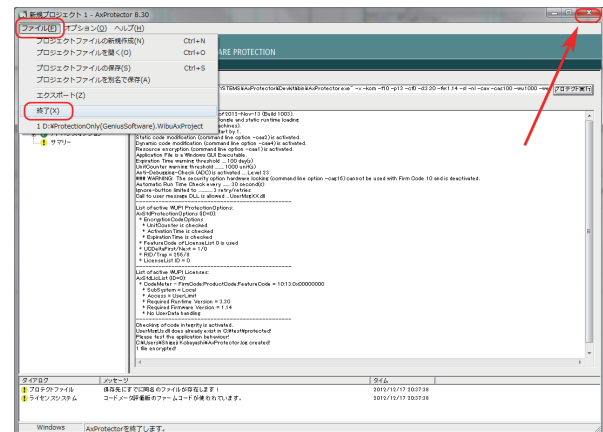
「サマリー」画面に、いままでに設定したプロテクト内容が表示されます。



"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。)

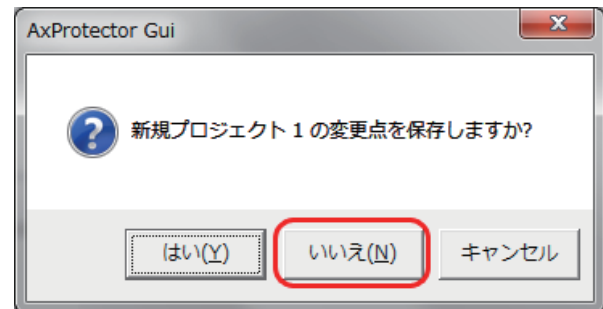


これで、sample.exeの暗号化処理は終了です。メニューバーの「ファイル(F)」-「終了(X)」,または画面右上の 閉ボタンでAxProtectorを終了します。



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。

¥test¥protected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/C 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/Cに登録します。CmStick/C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/Cにコードを登録します。

1. 用意するもの

CmStick/Cにコードを登録するには下記が必要です。

- (貴社の)コードメータFSB (CmFSB)
- CmStick/C

[NOTE]

ここでは、USBミニタイプのCmStick/Cを例にしますが、他のCmContainer(CmStick、CmSDCard、CmCFCardなど)の場合も操作は同じです。

2. コードメータライセンスエディタを起動する

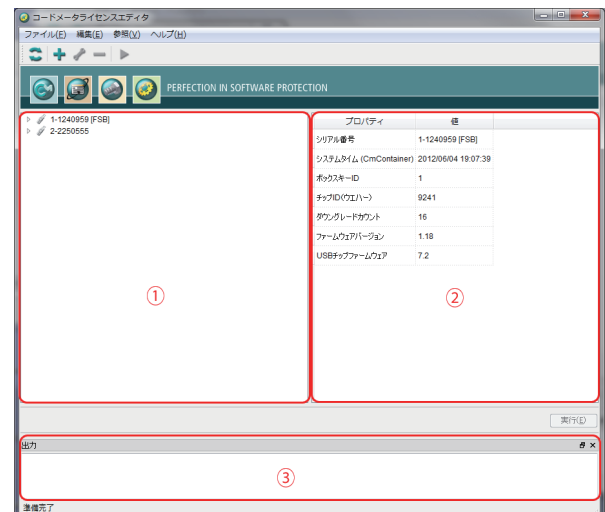
①コードメータ FSB と CmStick/C を装着する

まず、コードメータFSBとコードを登録するCmStick/C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。

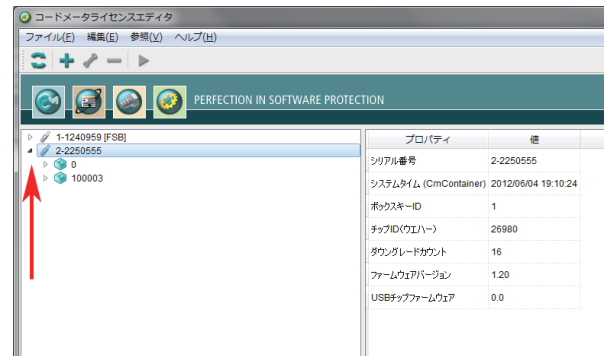
左ペイン①には、CmStick/Cのシリアル番号がツリー状で表示されます。コードメータFSBの場合は、シリアル番号の右側に[FSB]と表示されます。右ペイン②には、左ペイン①で選択されたCmStick/Cのプロパティが表示されます。また、下部の出力ペイン③には、「実行」ボタンで実行された操作結果のステータスが表示されます。



③ CmStick の▷マークをクリックする

CmStick/C (ここでは2-2250555) 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。◀マークをクリックすると元に戻ります。

正規版CmStick/Cをお持ちの方は、すでに貴社のファームコードが登録されています。その場合は、次の④と⑤の作業は不要ですので、⑥に進んでください。

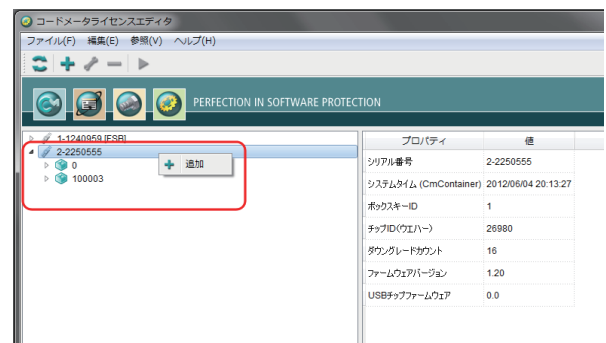


評価版CmStick/Cの場合は、既にファームコード= 0と100003が登録されています。ファームコード= 10を追加登録しますので、次の④「追加」ボタンを表示するに進んでください。

④ 「追加」ボタンを表示する

CmStick/C (ここでは2-2250555) を選択し、右マウスをクリックすると「追加」ボタンが表示されます。その「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。

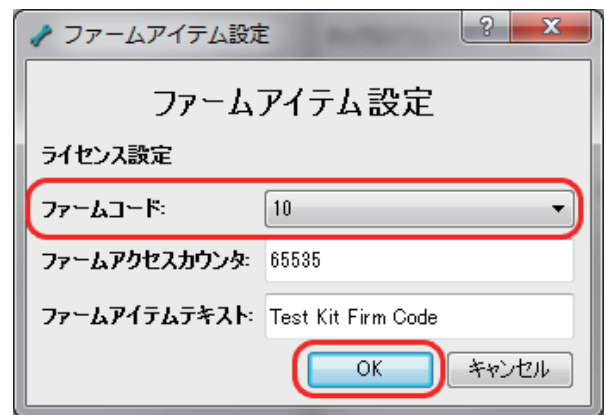


⑤ ファームコードを入力する

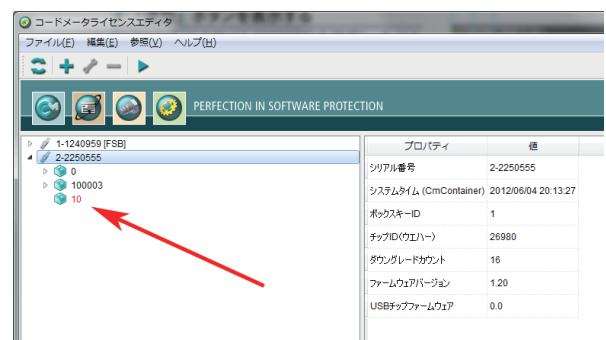
ファームアイテム設定画面で、ファームコードの入力欄にファームコード=10を選択し、「OK」ボタンをクリックします。

ファームアクセスカウンタは、暗号化・復号化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。



ファームアイテム設定画面で「OK」ボタンをクリックすると、コードメータライセンスエディタの左ペインのCmStick/C (ここでは2-2250555) に"10"が赤色で表示されます。



⑥ プロダクトコードを入力する

ファームコード"10"を選択し、右マウスをクリックすると「追加・編集・削除」メニューが表示されます。「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設定画面で、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設定画面では、プロダクトコード設定だけでなく、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。

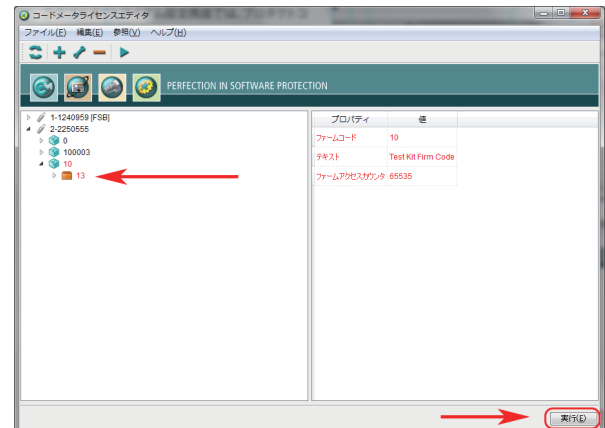
ここでは、プロダクトコードに"13"を入力して「OK」ボタンをクリックします。



⑦ CmStick/C に登録する

コードメータライセンスエディタの左ペインのCmStick/C 2-2250555にファームコード=10、プロダクトコード=13が赤色で表示されています。プロダクトコード=13が表示されていない場合は、ファームコード=10の左部の▷マークをクリックしてください。

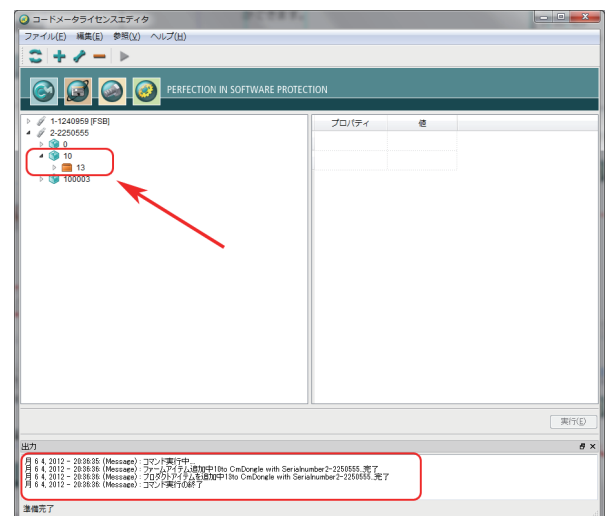
右下の「実行(E)」ボタンをクリックすると、ファームコード=10、プロダクトコード=13がCmStick/Cに書き込まれます。



⑧ 登録されたことを確認する

CmStick/C 2-2250555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。

下部の出力ペインには、実行結果のステータスが表示されます。



3. Web アドミン (WebAdmin) 上からも確認する

CmStick/C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、CmContainer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/C 2-2250555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 WebAdmin」が表示され、[内容]タブページには、CmStick/C 2-2250555の情報が表示されます。

「ライセンス」メニューをクリックすると、CmStick/C 2-2250555のライセンス内容が表示されます。

ファームコード=10 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。



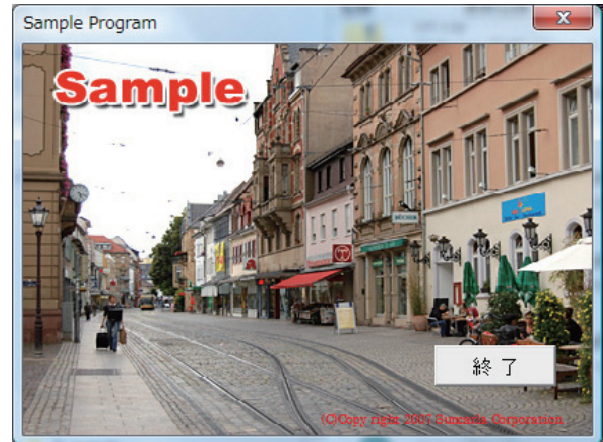
[NOTE]

大量のCmStick/C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 9 CmBoxPgmの使い方」をご参照ください。

3-4. 動作を確認する

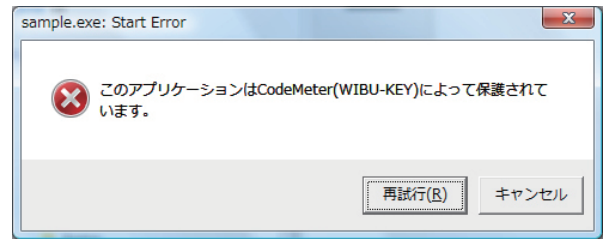
コードを登録したCmStick/CをPCに装着し、先ほど暗号化したsample.exeを起動します。正常に起動することを確認してください。次に、CmStick/Cを取り外した状態で、sample.exeを起動します。エラーメッセージが表示されsample.exeが起動しません。

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/C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 / 4-4 AxProtectorの各入力画面の説明 / 6. エラーメッセージ」をご参照ください。

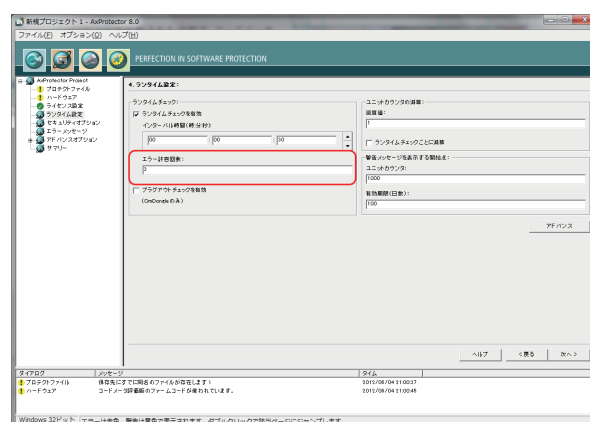
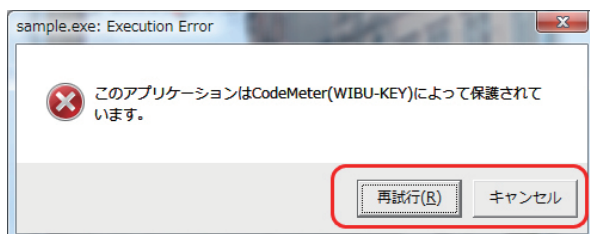
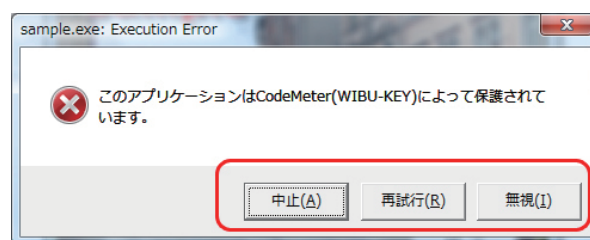
ランタイムチェックを確認する

sample.exeが起動された状態で、コードメータ CmStick/C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/C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「4. 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/C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「4. 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/Cにユニットカウンタを設定します。

現在のCmStick/Cのファームコード = 10、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とCmStick/C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。

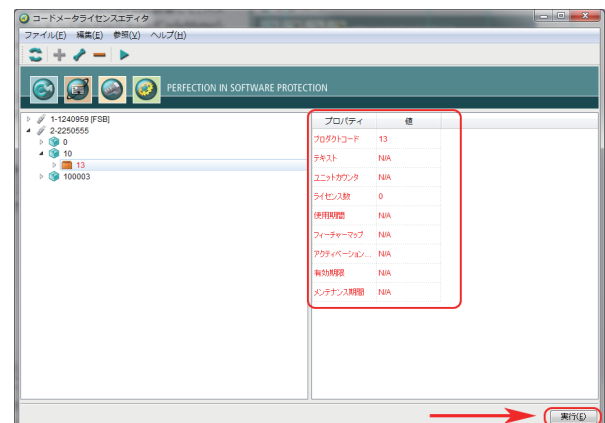
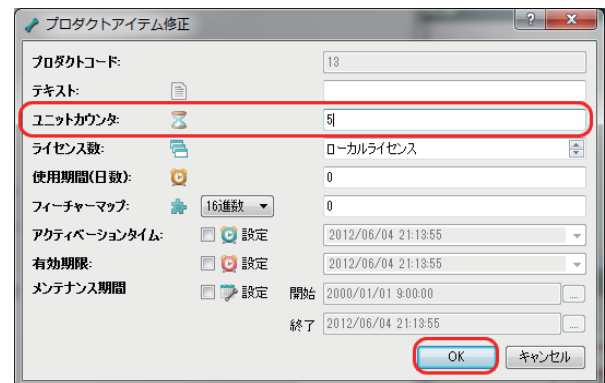
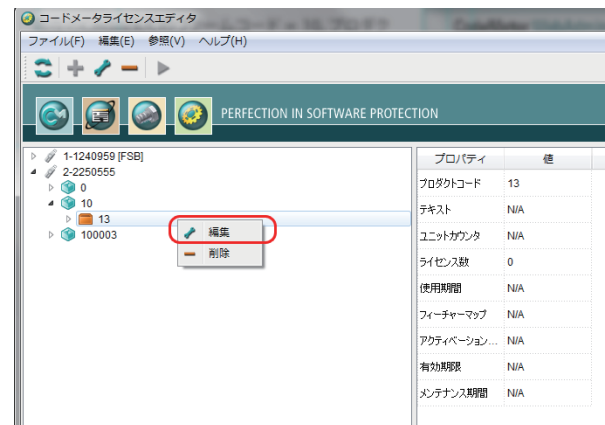
CmStick/Cのファームコード=10のプロダクトコード=13を選択し、右マウスをクリックして「編集」をクリックします。

②ユニットカウンタを設定する

プロダクトアイテムの変更画面で、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。

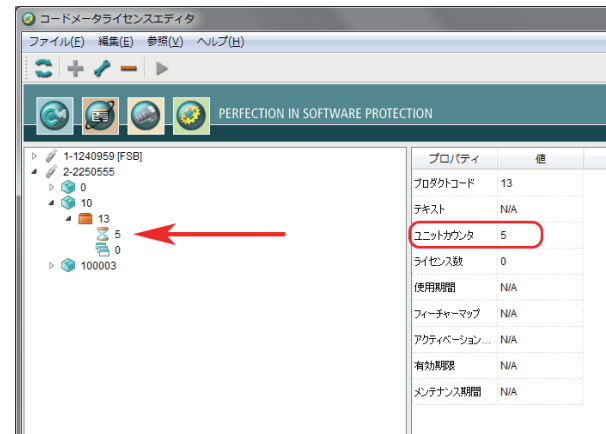
③「実行 (E)」ボタンをクリックする

コードメータライセンスエディタの右ペインのプロパティが赤色に変わっています。右下の「実行 (E)」ボタンをクリックすると、ユニットカウンタ=5がCmStick/Cに書き込まれます。



④ ユニットカウンタを確認する

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(WebAdmin)上からも確認できます。



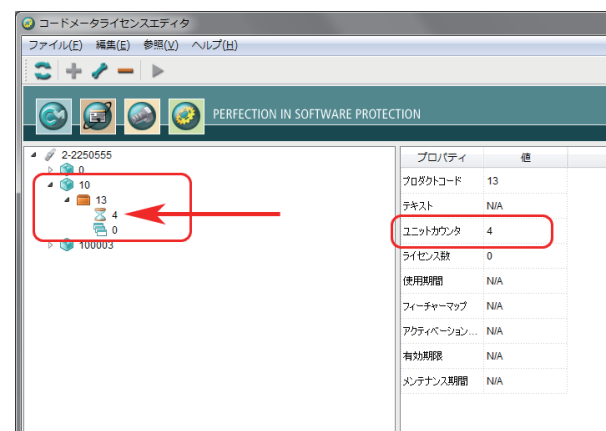
⑤ sample.exe を起動しユニットカウンタを確認

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5回起動すると、6回目は起動しない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、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/Cの「有効期限」に日時を設定します。

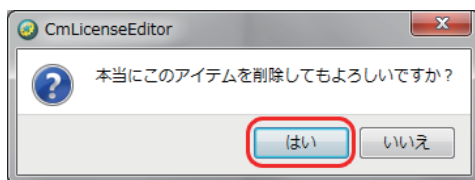
まず、コードメータライセンスエディタを起動し、前章「3-5.使用回数(ユニットカウンタ)を設定したプロテクトを行う」で設定したユニットカウンタを削除します。

① ユニットカウンタを削除する

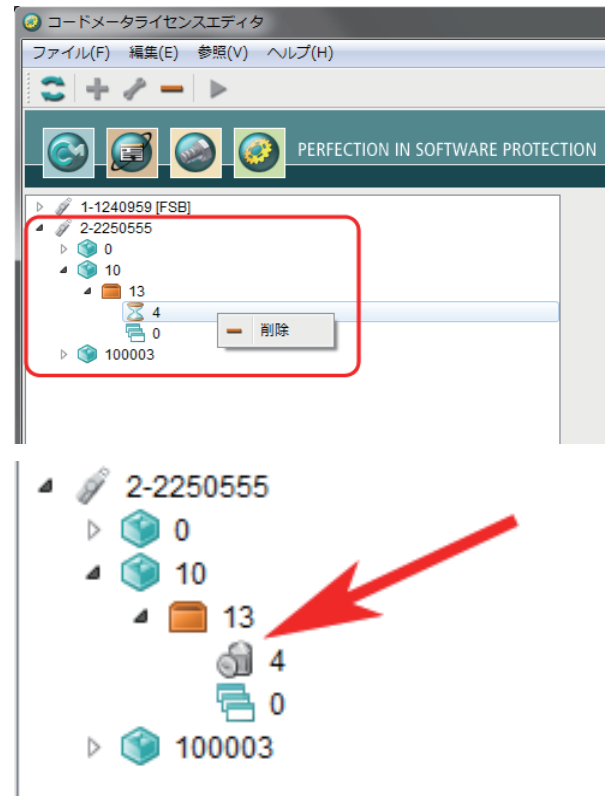
コードメータFSBとCmStick/CをPCに装着し、スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

ユニットカウンタ(ここでは4)を選択し、右マウスをクリックし、「削除」メニューをクリックします。

以下のメッセージが表示されますので、「Yes」をクリックします。



ユニットカウンタのアイコンが変わります。

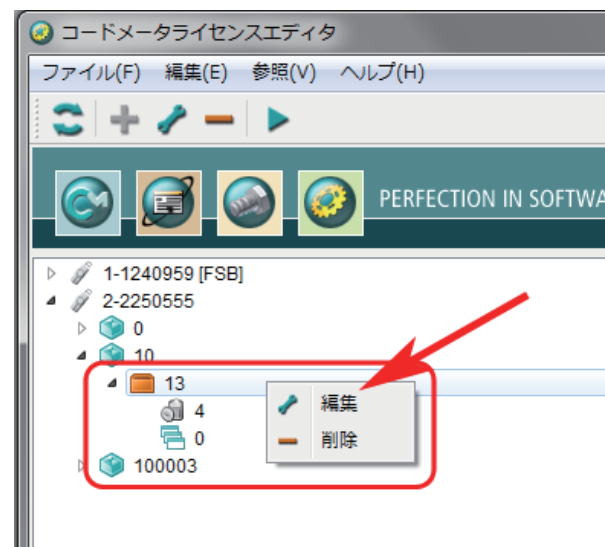


② 有効期限を設定する

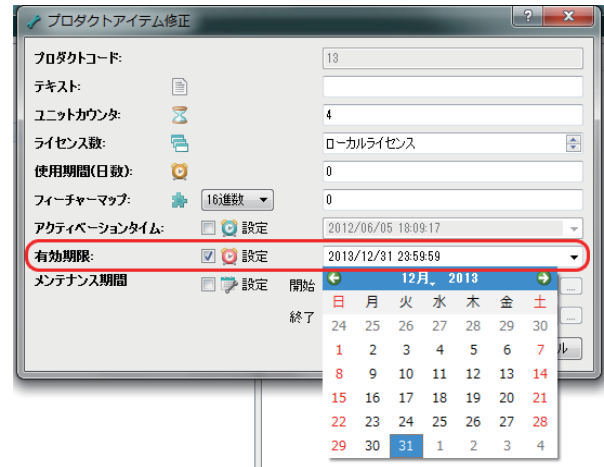
次に、プロダクトコード"13"を選択し、右マウスをクリックし、「編集」メニューをクリックします。

[NOTE]

"13"(プロダクトコード)を確実に選択してから右マウスをクリックしてください。

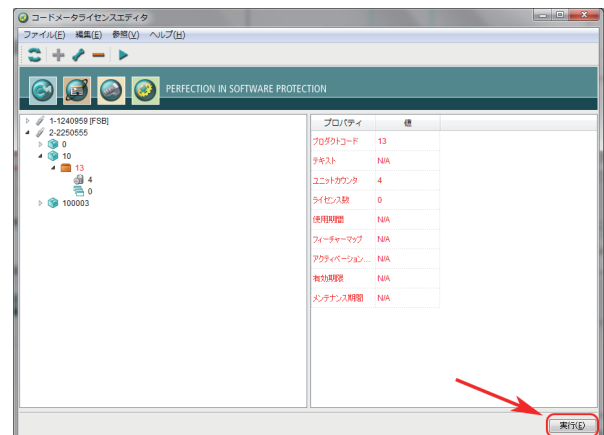


「プロダクトアイテムの変更」画面が表示されます。
「有効期限」にチェックを入れ、日付および時刻を設定し、「OK」ボタンをクリックします。



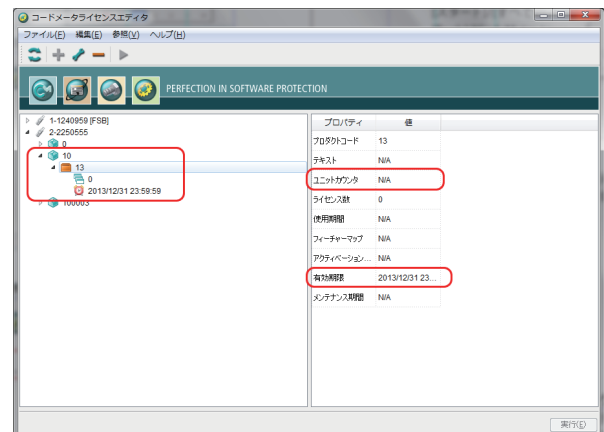
③「実行」ボタンをクリックして登録する

右ペインのプロパティが赤色表示になります。右下の「実行」ボタンをクリックして、CmStick/Cに編集内容を書き込みます。



④ 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、ユニットカウンタが削除され、有効期限が登録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、"ユニットカウンタ"が"N/A"になっており、"有効期限"に日付が設定されているのが確認できます。



Webアドミン(WebAdmin)上からも登録内容を確認することができます。

これで、使用有効期限2013年12月31日のCmStick/C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/C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/C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

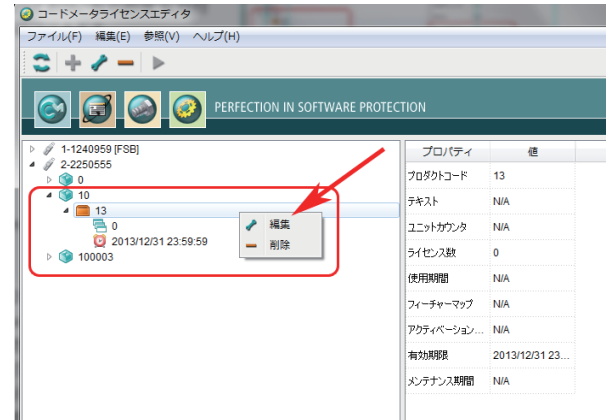
使用有効期限 (Expiration Time)と同じように、プログラムの使用開始日 (アクティベーションタイム = Activation Time)を設定することができます。指定した期日にならないとプログラムの起動ができません。使用有効期限と組み合わせて使用すると、確実に使用可能日数を設定することができます。

ファームコード=10/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)=2013年1月1日、使用有効期限=2013年12月31日を設定します。

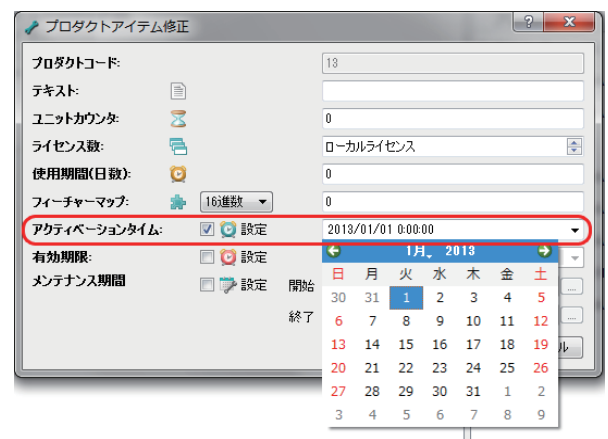
①アクティベーションタイムを設定する

コードメータFSBとCmStick/C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。

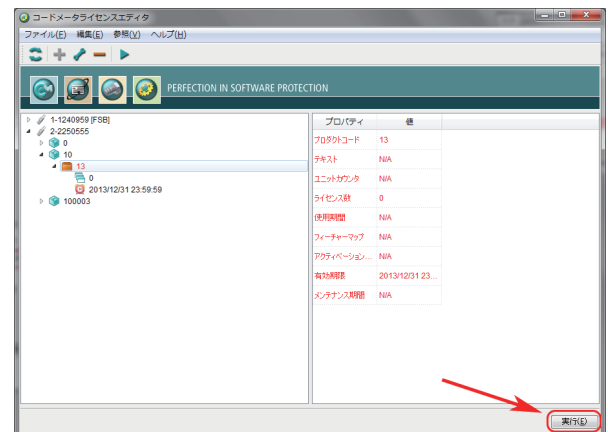


「プロダクトアイテムの変更」画面の「アクティベーションタイム」にチェックを入れ、「2013/01/01」を設定し、「OK」ボタンをクリックします。



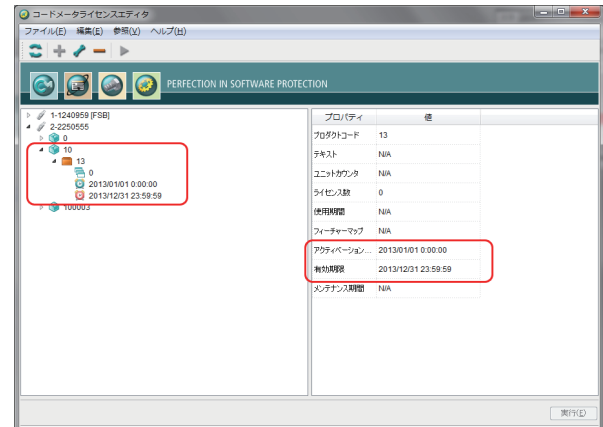
②「実行 (E)」 ボタンをクリックして登録する

右ペインのプロパティが赤色表示になります。右下の「実行(E)」ボタンをクリックして、CmStick/Cに編集内容を書き込みます。



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、新たにアクティベーションタイム(使用開始日)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、"アクティベーションタイム"と"有効期限"に日付が設定されているのが確認できます。



Webアドミン (WebAdmin) 上からも登録内容を確認することができます。

これで、使用開始日2013年1月1日、使用有効期限2013年12月31日のCmStick/C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とCmStick/C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。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、初めに削除しておいてください。

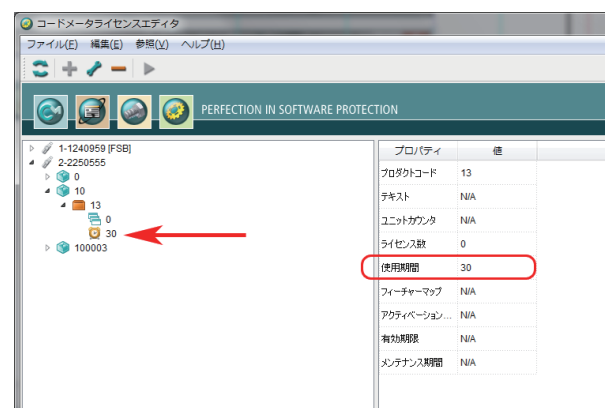
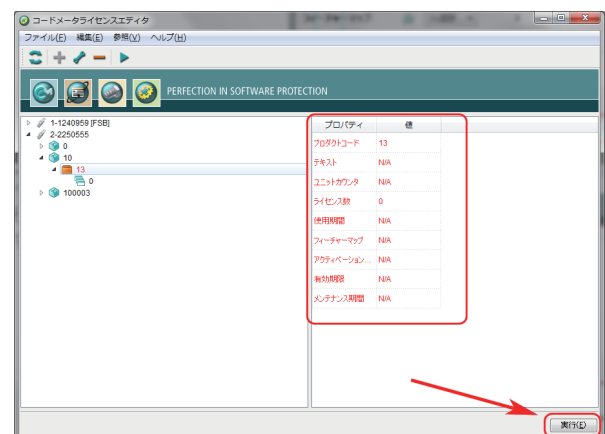
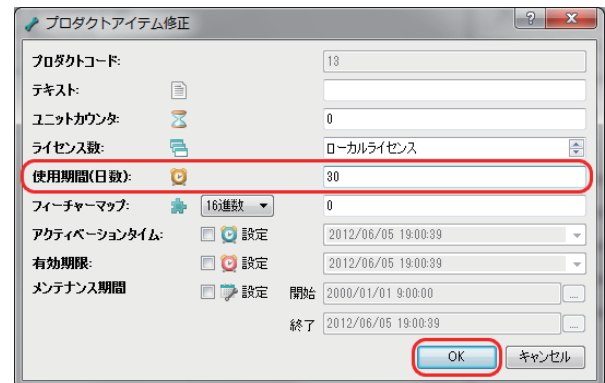
「プロダクトアイテムの変更」画面の「使用期間(日数)」に"30"を入力し、「OK」ボタンをクリックします。

② 「実行 (E)」 ボタンをクリックして登録する

右ペインのプロパティが赤色表示になります。右下の「実行(E)」ボタンをクリックして、CmStick/Cに編集内容を書き込みます。

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/Cが作成されます。WebAdmin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 - 使用開始時刻: n/a

と登録されています。



"使用開始時刻: 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、初めて暗号化されたプログラムを起動すると、その時の時刻が使用開始時刻としてCmStick/Cに自動的に登録されます。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。

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/C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削除を行うことも可能です。



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/Cの中に書かれており、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/Cのファームウェア Ver1.18以上が必要になります。

[NOTE]

CmStick/C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]

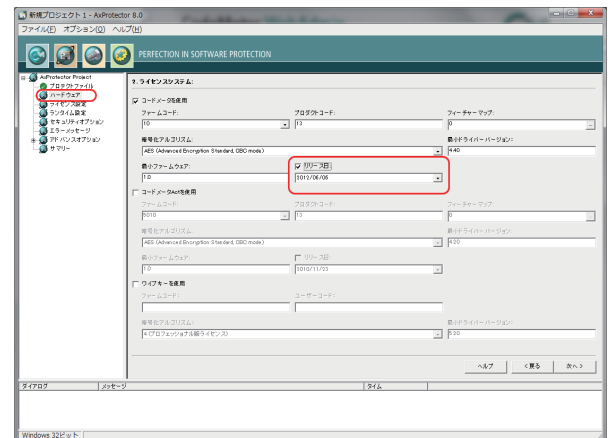
アプリケーションのリリース日を設定したり、CmStick/Cにメンテナンス期間を登録するには、コードメータFSBのファームウェアがVer1.18以上である必要があります。

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。
(2. ライセンスシステム (ハードウェア) 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/C内に保存されているメンテナンス期間と照合します。

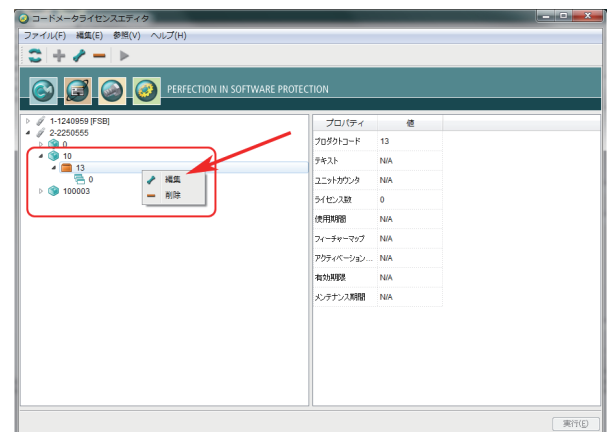
リリース日がメンテナンス期間内であればプログラムは起動し、メンテナンス期間外であれば起動しません。



② CmStick/C にメンテナンス期間を登録する

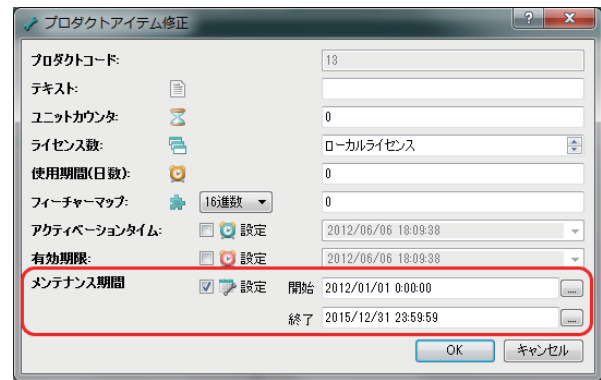
CmStick/Cにメンテナンス期間を登録するには、貴社のコードメータFSBが必要になります。メンテナンス期間を登録するCmStick/CとコードメータFSBをPCに同時に装着し、コードメータライセンスエディタを起動します。

ファームコード=10 / 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「編集」を選択します。



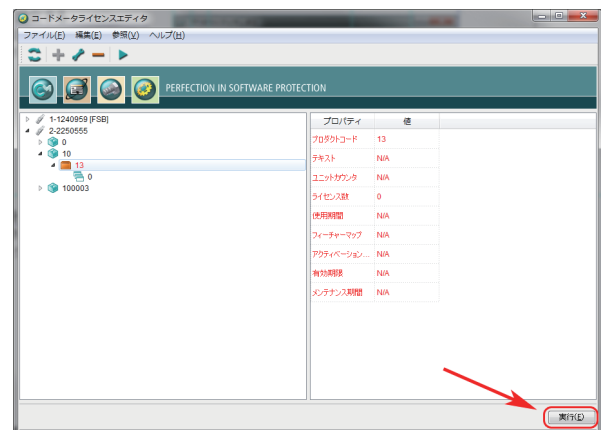
③ メンテナンス期間を設定する

メンテナンス期間の「設定」にチェックを入れ、開始日と終了日を設定し、OKボタンをクリックします。



④ 「実行 (E)」 ボタンをクリックして登録する

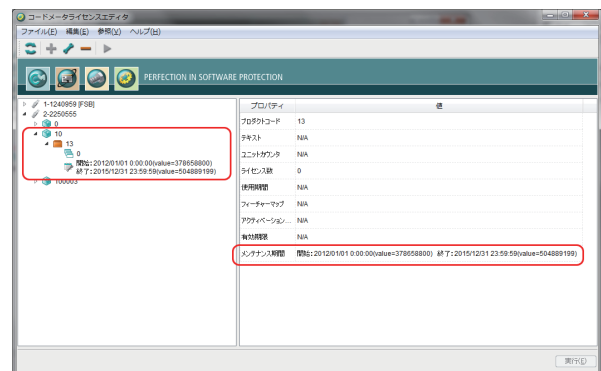
右ペインのプロパティが赤色表示になります。右下の「実行(E)」ボタンをクリックして、CmStick/Cに編集内容を書き込みます。



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2012年1月1日、終了2015年12月31日というCmStick/Cが作成されました。




⑥ WebAdmin から確認する

WebAdmin上から、ファームコード=10 / プロダクトコード=13 をクリックします。



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。



CodeMeter WebAdmin

ホーム 内容 サーバー 構成 診断 情報 ヘルプ

CMステック | ライセンス | ユーザーデータ | バックアップ/リストア

プロダクトアイテム詳細

CmContainer 2-2250555のプロダクトアイテム 10:13

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
ライセンス数		4	data, serial, counter	ローカル
Maintenance Period		4	data, serial, counter	Start: 2012-01-01 00:00:00 End: 2015-12-31 23:59:59

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はありませんでした。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 13 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. AxProtector の各入力画面の説明
- 4-5. コマンドラインでの使用方法

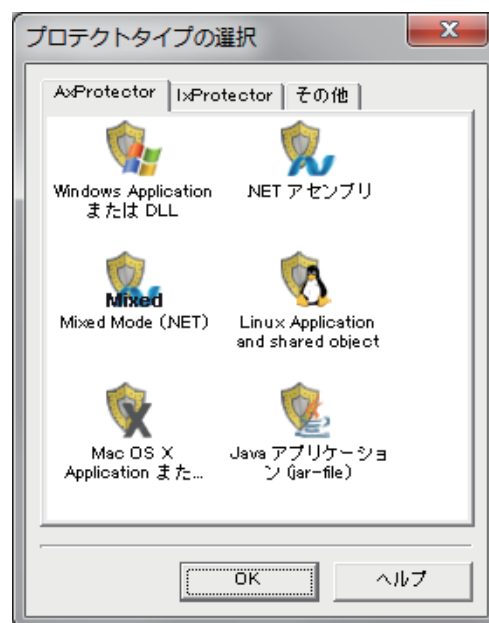
4-1. 自動暗号化ツール AxProtector について

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. Windows32bit/64bit Application (EXE, DLL)
2. .NET アセンブリ
3. Mixed Mode (.NET)
4. Linux Application/Shared Object
5. Mac OS X Application/Dylib
6. Javaアプリケーション (jar-file)



また、IxProtectorとして、

7. Windows 32bit/64bit Application (EXE, DLL)
8. .NETアセンブリ
9. Linux Application/Shared Object
10. Mac OS X Application/Dylib



その他、

11. ファイル暗号化

があります。

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

【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】から起動できます。



AxProtector 8.30 バージョン

AxProtectorGui.exe 8.30.1003.500

AxProtector.exe 8.30.1003.500

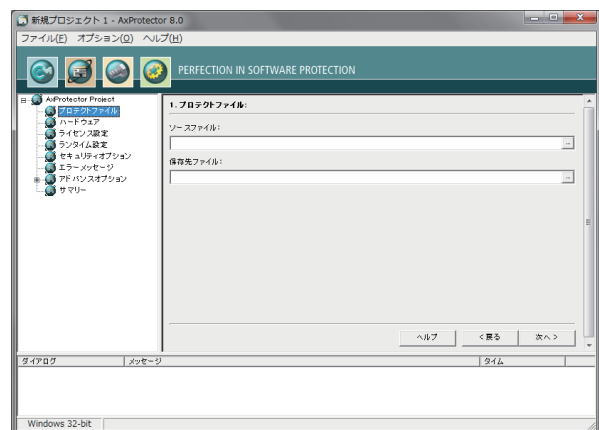
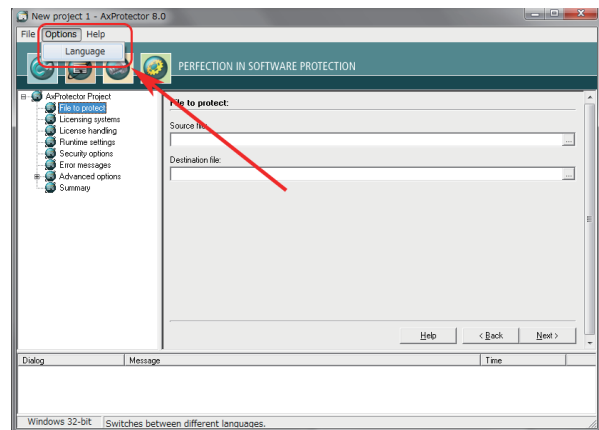
AxProtectorNet.exe 8.30.1003.500

WibuAxpJava32.dll 8.30.1003.500



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

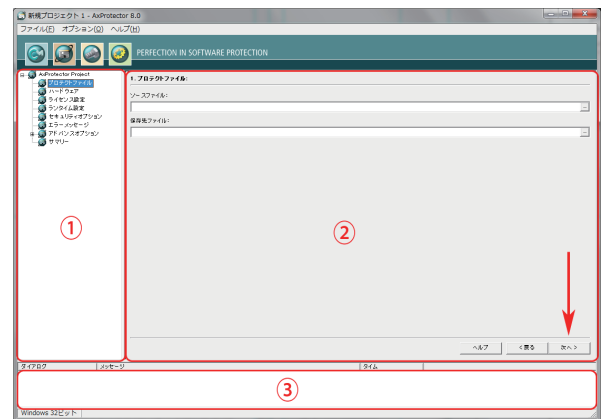
①のペインにはAxProtectorの操作項目が表示されます。各項目を上から順に実行すれば自動的に全てのパラメータを設定できるようになっています。ここで選択した項目に対するパラメータの値は②のペインで入力します。

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間はプロテクト操作が正常に行なわれません。

4-4. AxProtector の各入力画面の説明

1. プロテクトファイル

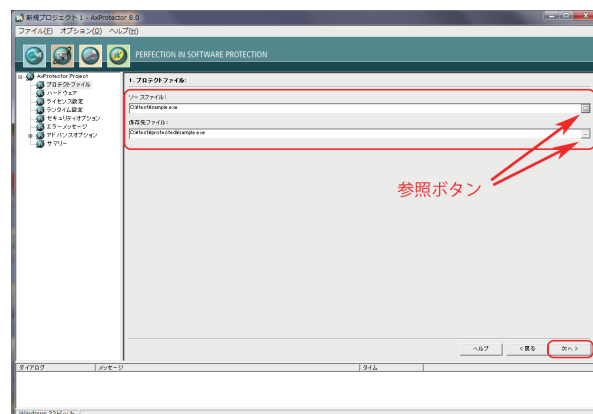
暗号化する前のオリジナルファイル名と、暗号化生成されるファイル名を指定します。AxProtectorを起動した直後はこの画面が表示されます。

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:

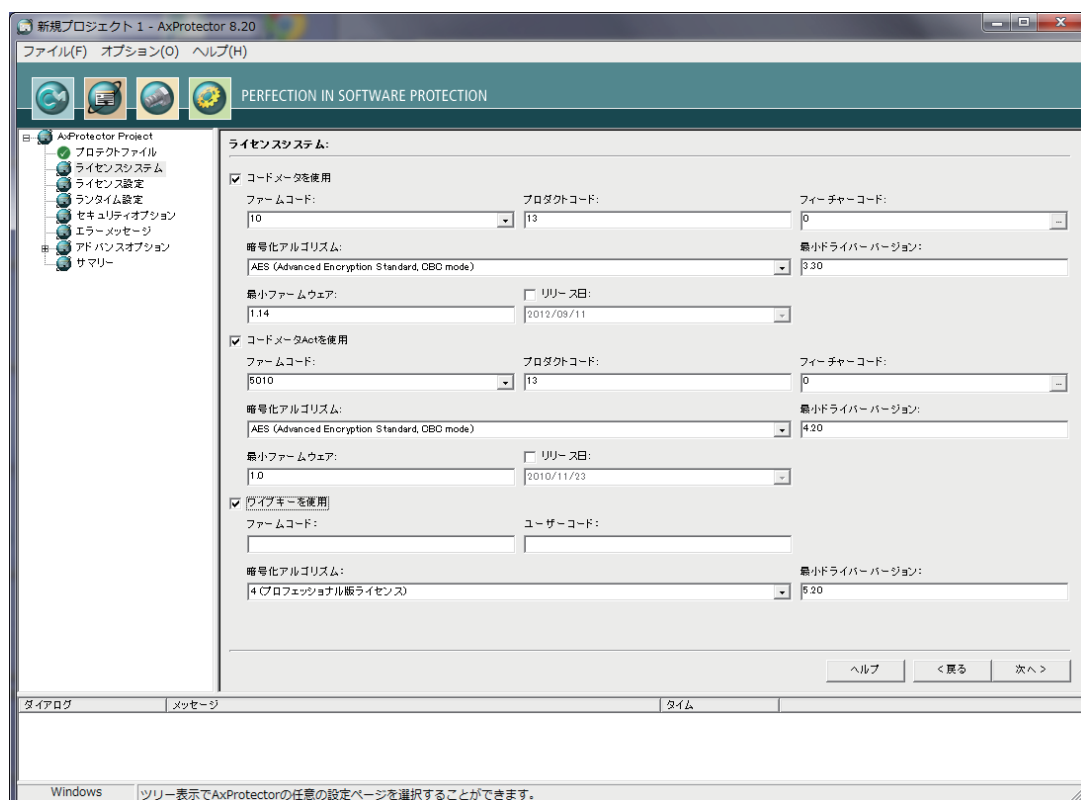
暗号化生成されるファイル名とその保存先のフォルダ名を指定します。右部の参照ボタンからファイル指定することもできます。



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. ハードウェア (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。



☐ コードメータを使用

ハードウェアベースのコードメータ (CmDongle) を使用する場合はここにチェックを入れます。

☐ コードメータActを使用

ソフトウェアベースのコードメータAct (CmActLicense)を使用する場合はここにチェックを入れます。

☐ ワイブキーを使用

ワイブキー (WibuKey) を使用する場合はここにチェックを入れます。

[NOTE]

複数を同時に指定することが可能です。その場合、暗号化されたプログラムは指定したキーのいずれかが見つかったと動作します。

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。

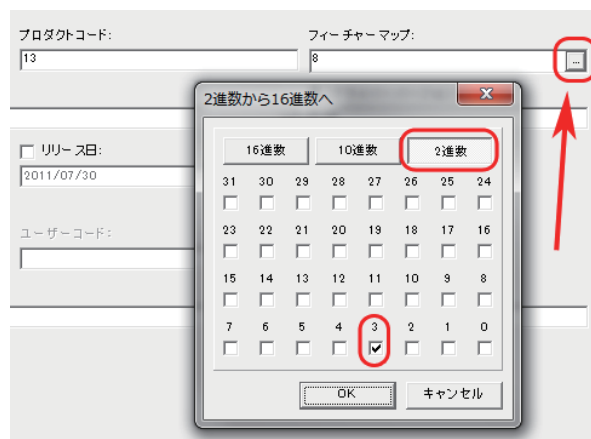
プロダクトコード:

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。

フィーチャーマップ(Feature Map):

フィーチャーマップを入力します。フィーチャーマップは、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、フィーチャーマップに"8"を指定することで実現します。フィーチャーマップの「Bin」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。フィーチャーマップは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。

[illegible]

ファームコード=10、プロダクトコード=13、フィーチャーマップ=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、フィーチャーマップ=8）を持つCmContainerが必要になります。

このフィーチャーマップ機能を利用すると、1 個のエントリで複数のバージョン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、フィーチャーマップ=1

バージョン1 ファームコード≒10、プロダクトコード≒13、フィーチャーマップ≒2

バージョン2 ファームコード=10、プロダクトコード=13、フィーチャーマップ=4

バージョン3 ファームコード=10、プロダクトコード=13、フィーチャーマップ=8

`0x00000000000000000000000000000000`**111** = 15 (Dec)

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 checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

OK キャンセル

`0x00000000000000000000000000000000`**1111** = 31 (Dec)

暗号化アルゴリズム:

最小ドライババージョン:

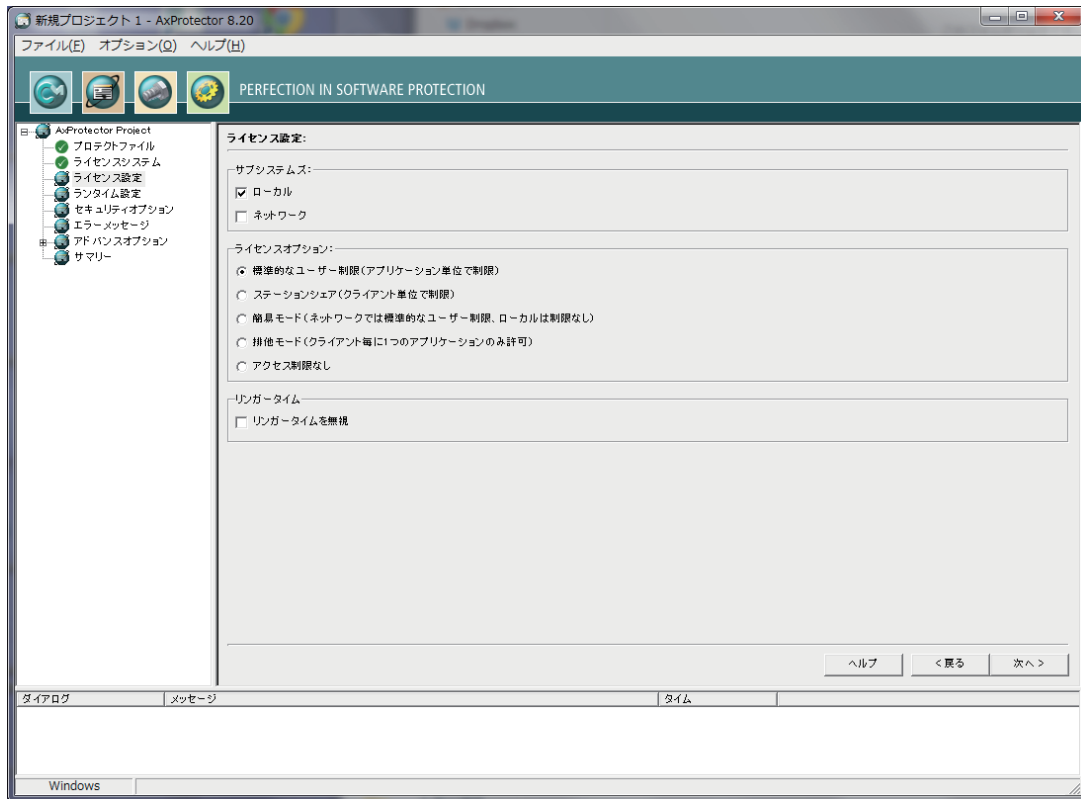
最小ファームウェア:

リリース日:

-54-

3. ライセンス設定

ネットワーク(LAN)上 または ローカルPC上に装着されたコードメータキーを自動検索して認識する設定を行ないます。



サブシステムズ：

☐ ローカル：

ローカルPCに装着されているコードメータキー (CmContainer) を検索します。

☐ ネットワーク：

ネットワーク(LAN)上のコードメータサーバーに装着されているコードメータキー (CmContainer) を検索します。

両方を選択した場合、最初にローカルPCをチェックし、そこでコードメータキー (CmContainer) が見つからなければネットワーク (LAN)上のコードメータサーバーを検索します。

ライセンスオプション:

○ 標準的なユーザー制限(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 (CmContainer) 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)

○ 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)

○ 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)上のコードメータキー (CmContainer) に対しては「標準的なユーザー制限 (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 (CmContainer) に対しては制限がありません。

○ 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。

○ アクセス制限なし(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 (CmContainer) 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。

リンガータイム:

リンガータイムとは、ライセンスの解放(リリース)を指定時間行わない機能です。プロテクトされたプログラムが終了するとライセンスが解放(リリース)されますが、ライセンス解放を直ちに実行するのではなく、ある程度の時間(リンガータイム)を置いてから解放する場合に使用します。これは、プログラムの再起動などで再度ライセンス割り当てが必要な場合に、他ユーザーがそのライセンスを使用する(横取りする)ことを一時的にブロックするためのものです。(再起動による再割り当てを確保するため)このリンガータイムは、ネットワーク機能と一緒に利用します。

[リンガータイムの設定方法]

CmBoxPgmでCmContainer側にコマンドラインで設定します。CodeMeter License Editor (GUI)にはまだ対応していません。パラメータは、
/plt - | [<seconds>] [, <TVB dep.>]

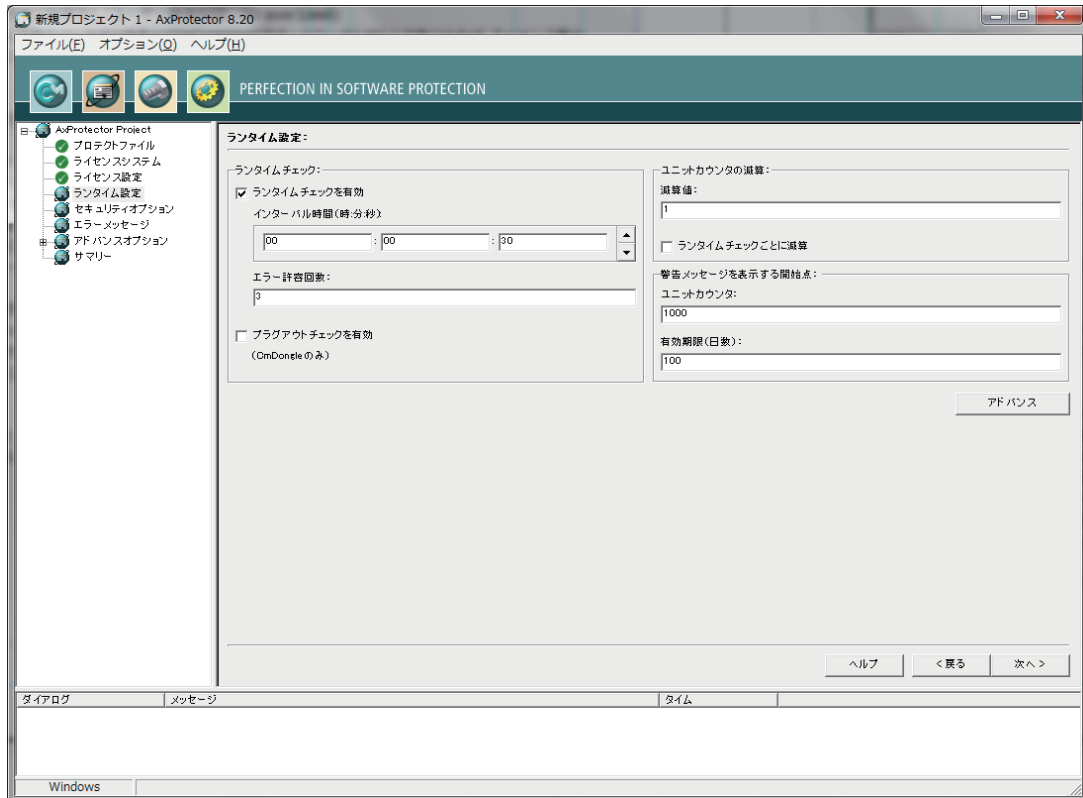
(例)

/plt15

* PIOにリンガータイムを15秒設定する。

4. 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)、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(CM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:

コードメー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、コードメー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ないます。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)

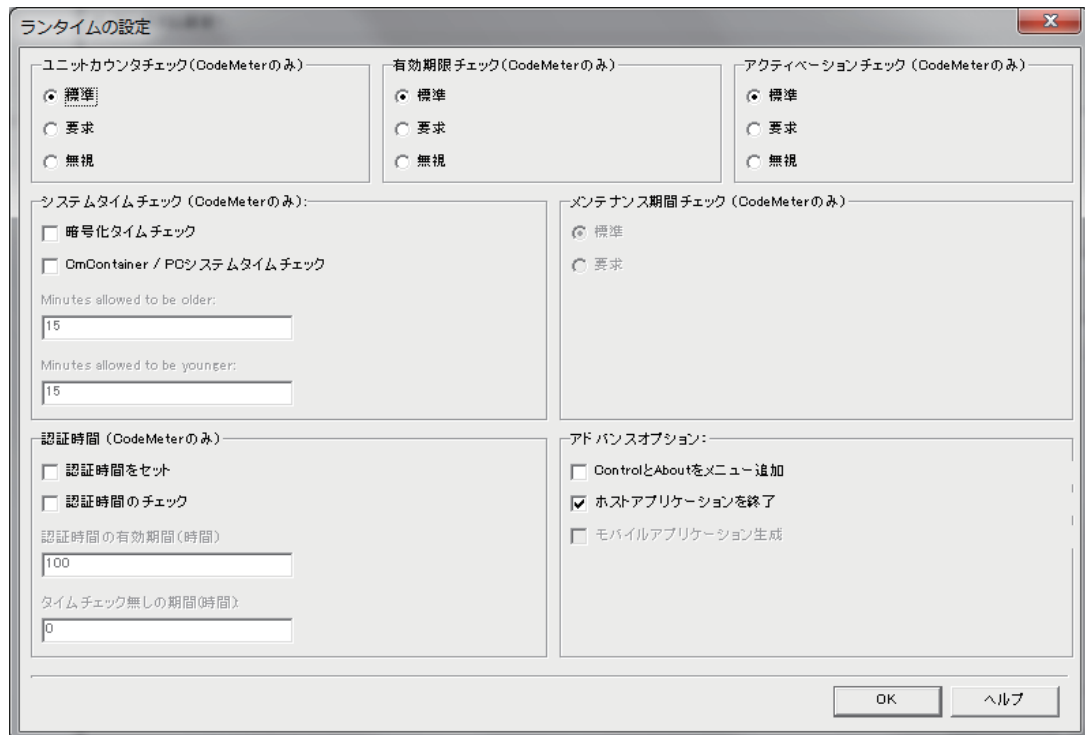
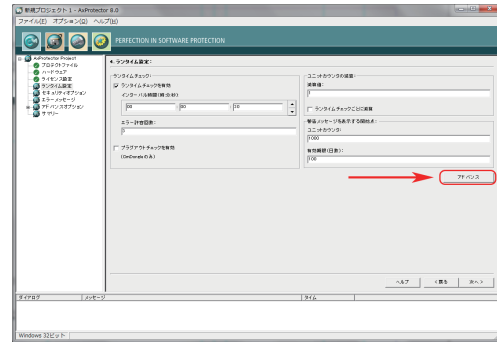
☐ **有効期限(日数):**

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。

**ユニットカウンタチェック (CodeMeterのみ)：****○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

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

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。

有効制限チェック (CodeMeterのみ) :

○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

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

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。

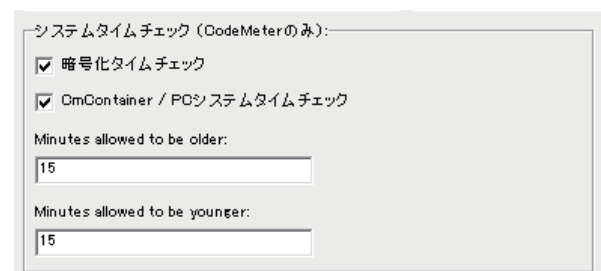
システムタイムチェック (CodeMeterのみ) :

☐ 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、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。



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

Minutes allowed to be older:

15

Minutes allowed to be younger:

15

Minutes allowed to be older

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。

Minutes allowed to be younger

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。

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**

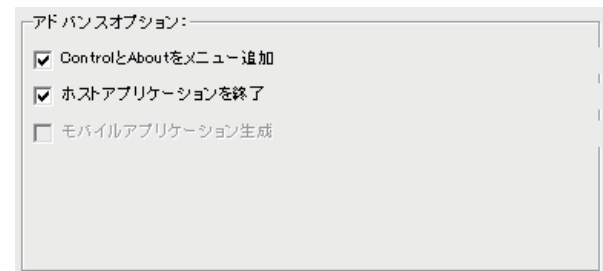
システムメニューとAboutメニューをアプリケーションに追加します。このオプションは、ランタイムチェックが実行されているときに有効です。

☐ **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)

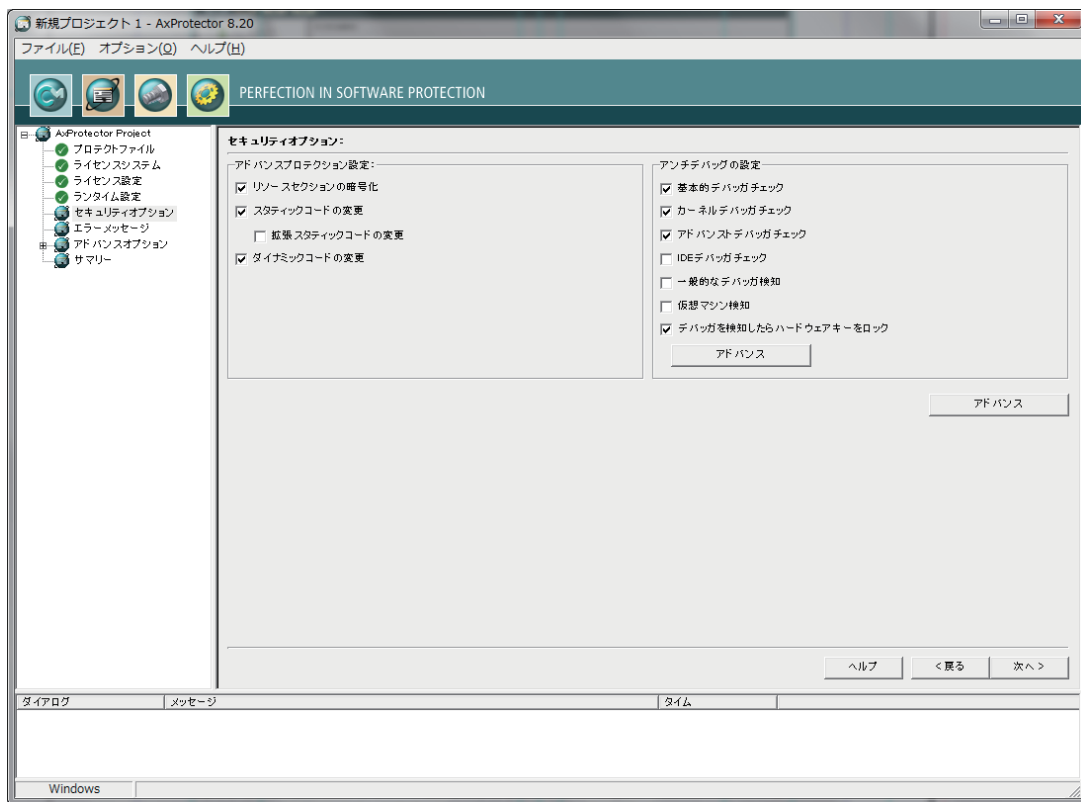
☐ **モバイルアプリケーション生成**

現在未使用。



5. 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。

☐ 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な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

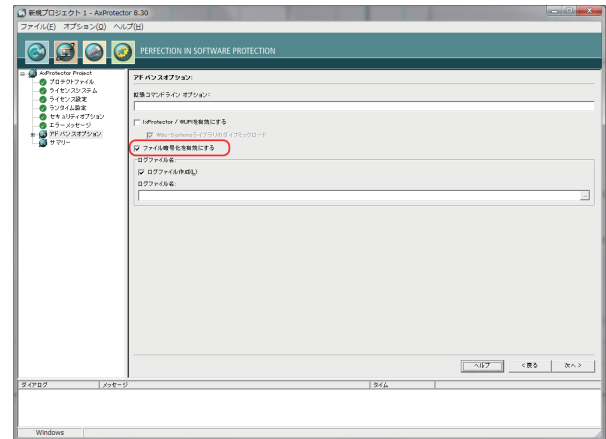
アプリケーションが実行されている間、コードを変更します。

[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。

[NOTE]

アドバンスオプションで、「ファイル暗号化を有効にする」にチェックを入れる場合、「スタティックコードの変更」はチェックをはずす必要があります。



アンチデバッグの設定

アンチデバッグの設定です。解析強度を重視する場合は可能な限りチェックを入れてご利用ください。アプリケーションによって動作に影響が生じる場合は、以下を参考にチェックをはずしてください。

□基本的デバッグチェック

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。

□カーネルデバッグチェック

「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グチェック

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 デバッグチェック

Visual StudioやDelphiなどの統合開発環境(IDE)のデバッグを禁止します。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。

□一般的なデバッグ検知

デバッグ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。

□デバッグを検知したらハードウェアキーをロック

デバッグ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。

「アドバンス」ボタンをクリックして、ロックモードを選択することができます。

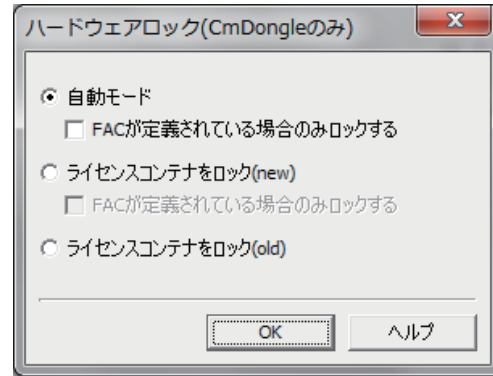


○自動モード

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：

☐ ウイルスチェック

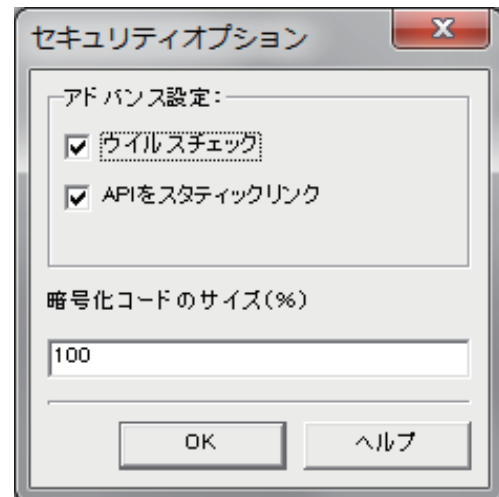
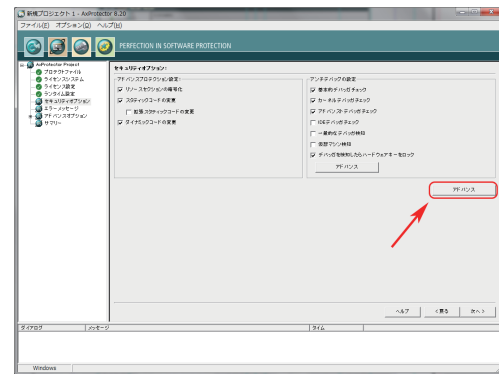
実行ファイル感染型のウイルスに対するチェック機能を追加します。チェックサムを利用します。ウイルス感染の疑いのある場合、プログラムを起動する前に警告メッセージを出します。

☐ 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(%)：

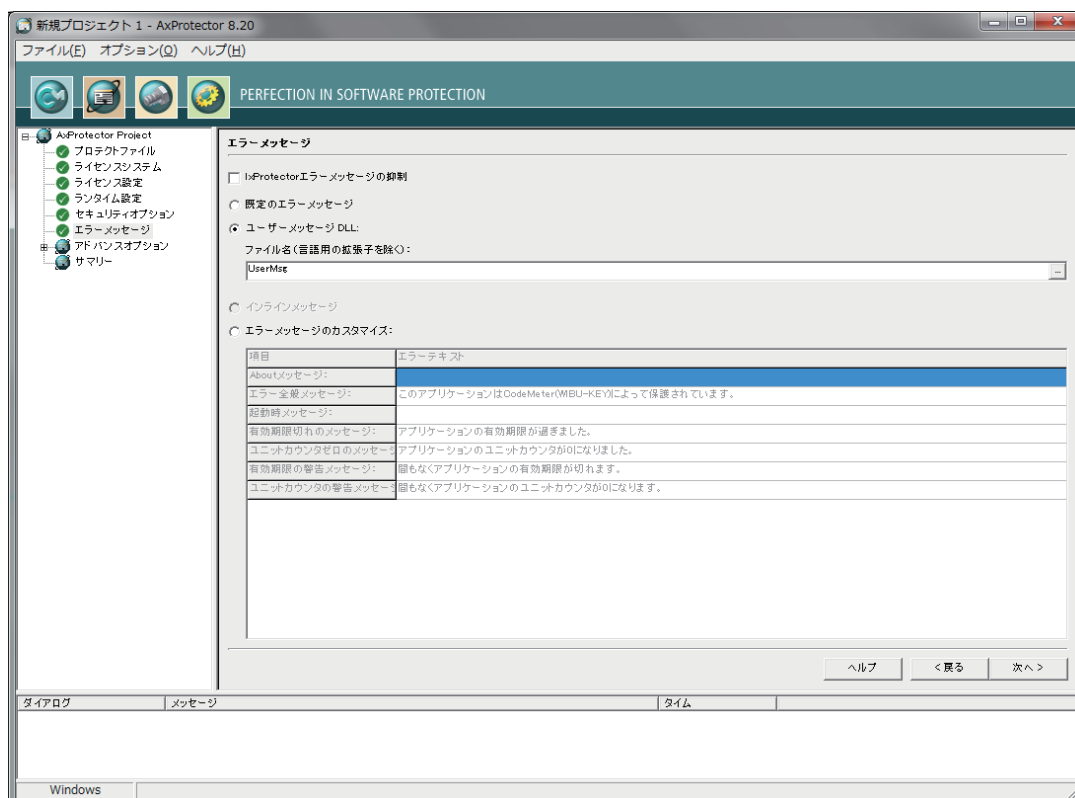
暗号化するコードサイズを%にて設定します。
(100%推奨)



6. 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 (UserMsgUs.dllとUserMsgJp.iniファイルを使用)
- インラインメッセージ (.NET Assembly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。

【UserMsgJa.ini ファイルの説明】

[Main]

BuyUrl: WebサイトのURLを設定します。

Logo: 左部のメッセージ画面に表示されるBMP画像ファイルを指定します。(BMPのみ有効)

MainText: エラーメッセージ本文を入力します。改行は¥nで行います。

BuyText: 購入サポート窓口情報などを入力します。

OkButton: メッセージ画面の[OK] ボタンの名前を設定します。

CancelButton: メッセージ画面の[キャンセル] ボタンの名前を設定します。

RetryButton: メッセージ画面の[再試行] ボタンの名前を設定します。

IgnoreButton: メッセージ画面の[無視] ボタンの名前を設定します。

BuyNowButton: メッセージ画面の[購入] (HPへのリンク) ボタンの名前を設定します。

BuyHint:

BuyHint=on BuyTextメッセージを表示

Buyhint=off BuyTextメッセージを非表示

UnitCounterMax = プログレスバーに表示するユニットカウンタ最大数 (例:1000)

DaysMax = プログレスバーに表示する有効日数最大数 (例:100)

UnitCounterText = ユニットカウンタのタイトル (例:回数:)

ExpirationDateText = 有効期限のタイトル (例:日数:)

;***** CodeMeter ERROR CODES ***** (BEGIN)

[CM32]

Caption=CodeMeter - 使用可能なクレジットがありません!

MainText=このソフトウェアはペイパーユース版で、すべてのクレジットが使用されました。\\n\\n"キャンセル" ボタンをクリックしてソフトウェアを終了してください。

HeadLine=使用可能なクレジットがありません!

[CM35]

Caption=CodeMeter - ライセンスの有効期限が終了しました!

MainText=このソフトウェアは使用有効期限が設定されたバージョンです。使用有効期限が終了しました。\\n\\n"キャンセル" ボタンをクリックしてソフトウェアを終了してください。

HeadLine=ライセンスの有効期限が終了しました!

[CM37]

Caption=CodeMeter - ソフトウェアはまだアクティベートされていません!

MainText=このソフトウェアは使用有効期限付きバージョンで、まだアクティベーションが行われていません。\\n\\n"キャンセル" ボタンをクリックし、ソフトウェアを終了してください。

HeadLine=まだアクティベートされていません!

[CM71]

Caption=CodeMeter - 使用期間が過ぎました!

MainText=使用期間が過ぎたためアプリケーションを起動することができません。

HeadLine=使用期間が過ぎました!

[CM200]

Caption=CodeMeter - ライセンスが見つかりません! [CM200]

MainText=このソフトウェアに必要な #FirmProductCode# を持つCM-Stickが見つかりません。\\n\\n正しいCM-StickをPCに装着してください。

HeadLine=ライセンスが見つかりません!

[CM52]

Caption=CodeMeter - CM-Stick使用不可!

MainText=接続されているCM-Stickは使用不可の状態です。\\n\\n使用可能な状態にして、"再試行"ボタンをクリックしてください。

HeadLine=CM-Stick使用不可!

[CM106]

Caption=CodeMeter - ライセンスが見つかりません![CM106]

MainText=このソフトウェアに必要な #FirmProductCode# を持つCM-Stickが見つかりません。\\n\\n正しいCM-StickをPCに装着してください。

HeadLine=ライセンスが見つかりません!

[CM212]

Caption=CodeMeter - 最大ユーザー数に達しました!

MainText=利用できる #FirmProductCode# ライセンスがありません。\\n\\n"再試行" をクリックして利用可能なライセンスをスキャンしてください。

HeadLine=最大ユーザー数に達しました!

[CM255]

Caption=CodeMeter - 不明なエラー!

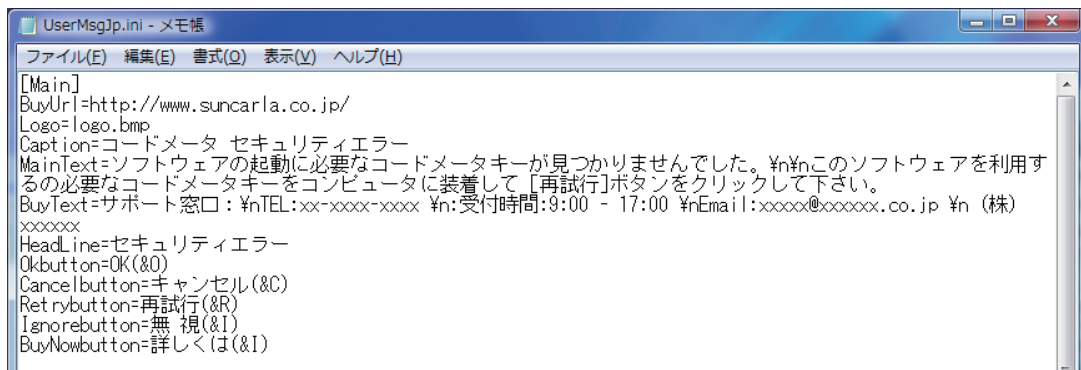
MainText= 不明なCM-Stickエラー、エラー(255)。

HeadLine=ライセンスが見つかりません!

,***** CodeMeter ERROR CODES ***** (END)

【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。



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。



3. 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、コードメータのデフォルトメッセージが表示されます。

○ 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:

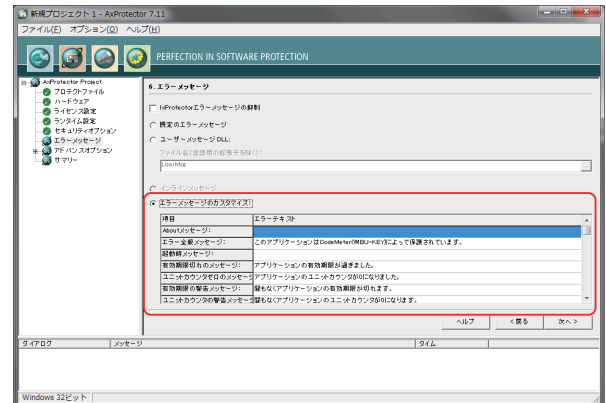
画面には表示されません。

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)または使用期限 (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。

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。

○ インラインメッセージ：

インラインメッセージは、.NETアセンブリを暗号化する際に使用します。インラインメッセージを選択して暗号化ファイルを作成すると、暗号化ファイルと同じフォルダに次のファイルが作成されます。

UserMessage.ini (英語)
 UserMessageDe.ini (ドイツ語)
 UserMessageJa.ini (日本語)
 UserMessageRu.ini (ロシア語)
 UserMessageZh.ini (中国語)
 UserMsg.bmp

日本語メッセージを作成するには、UserMessageJa.iniを編集します。

[Main]

BuyUrl: WebサイトのURLを設定します。
 Logo: 左部のメッセージ画面に表示されるBMP画像ファイルを指定します。(BMPのみ有効)
 Caption: メッセージ画面のタイトルを設定します。
 MainText: エラーメッセージ本文を入力します。改行は¥nで行います。
 BuyText: 購入サポート窓口情報などを入力します。
 HeadLine: エラーメッセージのヘッドラインを設定します。
 Okbutton: メッセージ画面の[OK] ボタンの名前を設定します。
 Cancelbutton: メッセージ画面の[キャンセル] ボタンの名前を設定します。
 Retrybutton: メッセージ画面の[再試行] ボタンの名前を設定します。
 Ignorebutton: メッセージ画面の[無視] ボタンの名前を設定します。
 BuyNowbutton: メッセージ画面の[購入] (HPへのリンク) ボタンの名前を設定します。
 BuyHint:
 BuyHint=on BuyTextメッセージを表示
 Buyhint=off BuyTextメッセージを非表示
 UnitCounterMax = プログレスバーに表示するユニットカウンタ最大数 (例:1000)
 DaysMax = プログレスバーに表示する有効日数最大数 (例:100)
 UnitCounterText = ユニットカウンタのタイトル (例:回数:)
 ExpirationDateText = 有効期限のタイトル (例:日数:)

[NOTE]

.NETアセンブリのユーザーメッセージに使用するファイル名は、"**UserMessageJa.ini**"である点にご注意ください。Windows32/64 アプリケーション (アンマネージドコード) に使用するユーザーメッセージのファイル名は"UserMsgJa.ini"です。

.NET アセンブリの場合 -->UserMessageJa.ini
Windows32/64アプリケーションの場合-->UserMsgJa.ini

[NOTE]

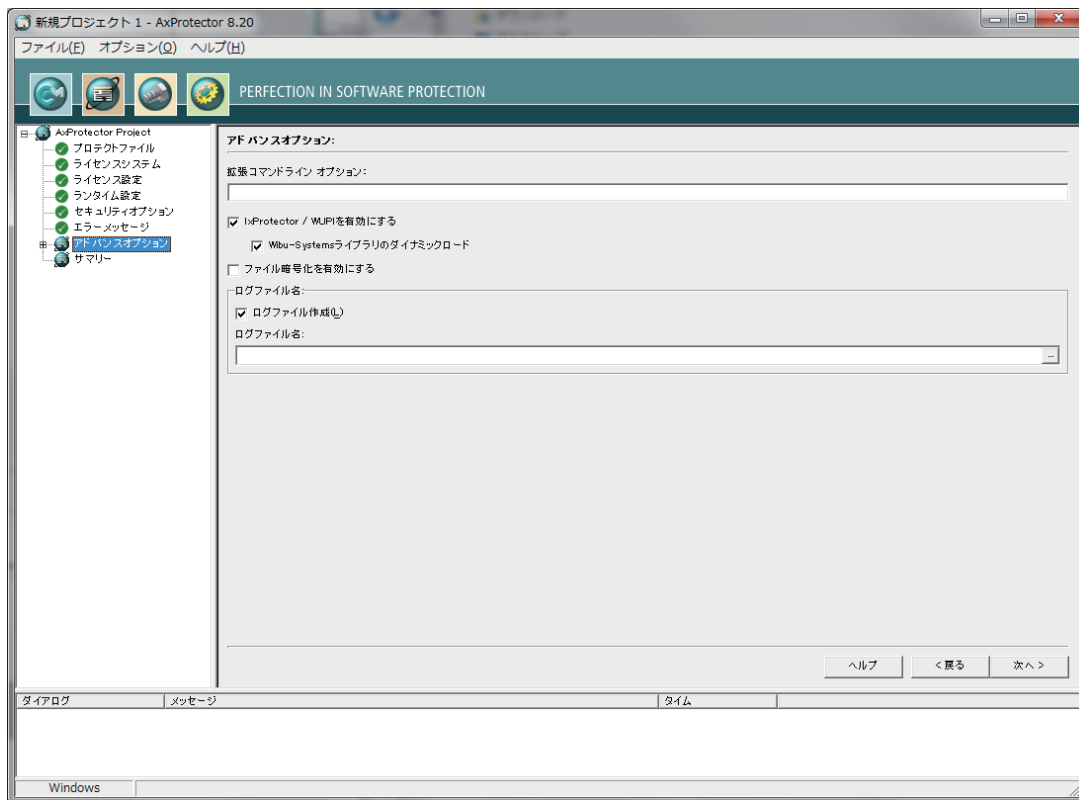
.NETアセンブリの場合、UserMsgUs.dllは使用しません。(フォルダに存在する必要がありません。)

オリジナルのメッセージを作成する

コードメータにはデフォルトのメッセージとは別に、VC++を使ってオリジナルのメッセージを作成することができます。下記にソースファイル(UserMsgUs08.sln)があります。

¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥Samples¥UserMessage¥Win

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。

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し、実行したあとは再び暗号化しておくという、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 6 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。

☐ **ファイル暗号化を有効にする**

データファイルを暗号化します。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。
(復号化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

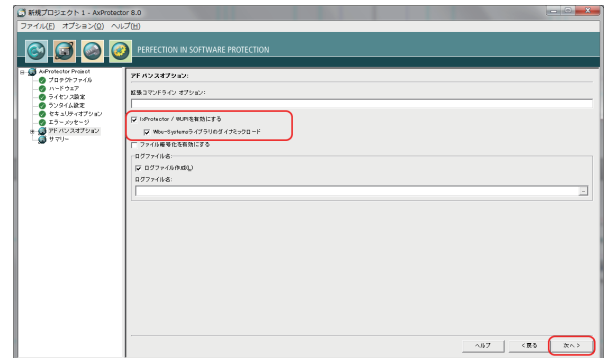
ログファイルが作成されるパスとファイル名を指定します。ファイル名だけを指定した場合は、
%¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin
に作成されます。

7-1. lxProtector/WUPIの使い方

① lxProtector/WUPIを有効にする

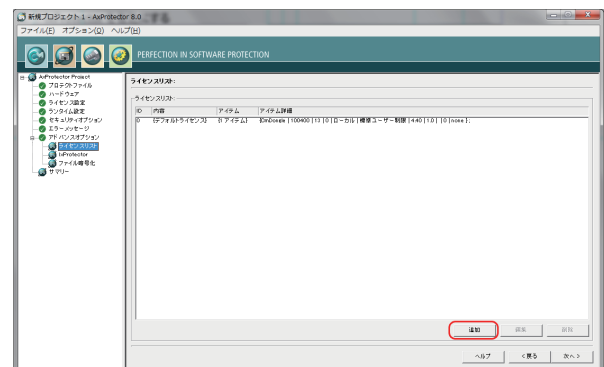
アドバンスオプション画面で、「lxProtector/WUPIを有効にする」にチェックを入れ、「次へ」をクリックする。

☐ **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。lxProtector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

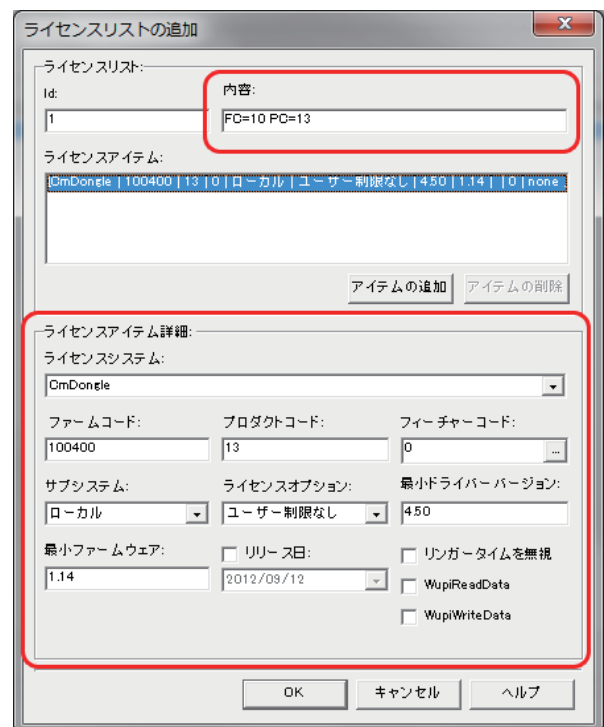
Idに対してライセンス内容を登録します。「ライセンスリストの追加」画面で行います。

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:**

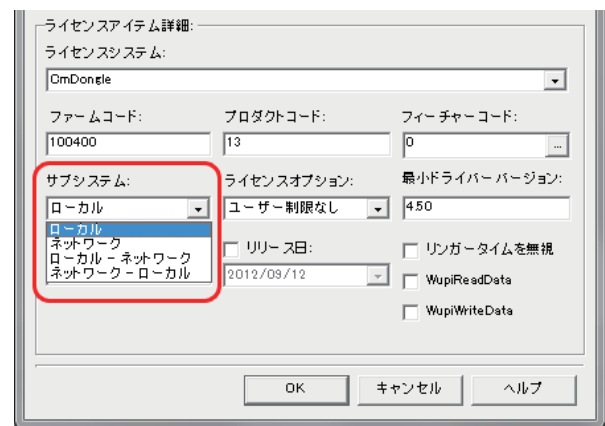
サブシステムを割り当てます。

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

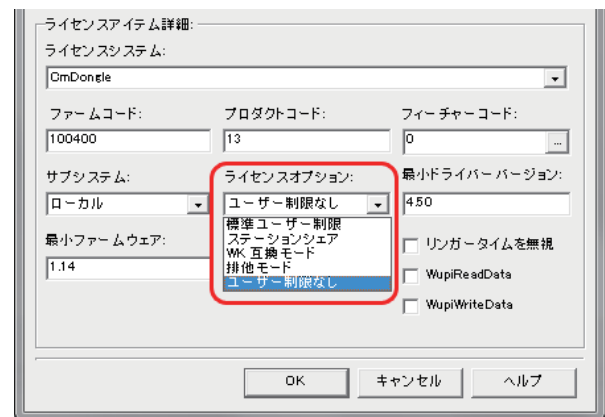
WK 互換モード:

ネットワークでは標準制限、ローカルは制限なし

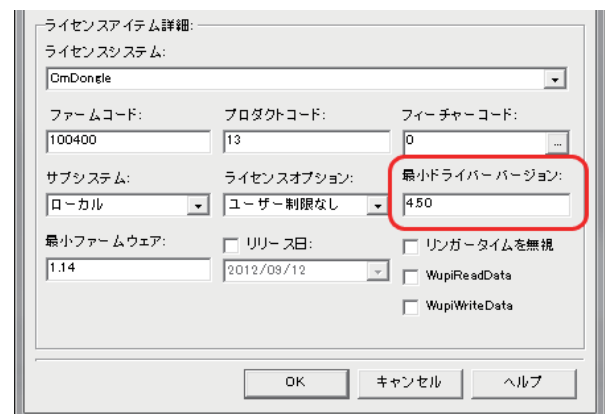
排他モード:

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:**

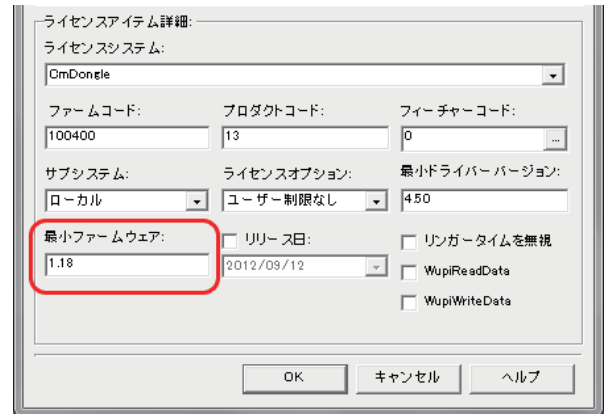
コードメータランタイムキットの最小バージョンを指定します。ここで指定したバージョンより古いバージョンがインストールされていると、暗号化されたプログラムは起動しません。コードメータランタイムキットは、OS環境に応じて定期的にバージョンアップしており、貴社プログラムが古いバージョンで動作し、誤動作するのを防ぐことができます。



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。



ライセンスアイテム詳細:
ライセンスシステム:
CmDongle

ファームコード:	プロダクトコード:	フィーチャーコード:
100400	13	0
サブシステム:	ライセンスオプション:	最小ドライババージョン:
ローカル	ユーザー制限なし	4.50
最小ファームウェア:	リリース日:	リンガータイムを無視
1.18	2012/09/12	<input type="checkbox"/> WupiReadData
		<input type="checkbox"/> WupiWriteData

OK キャンセル ヘルプ

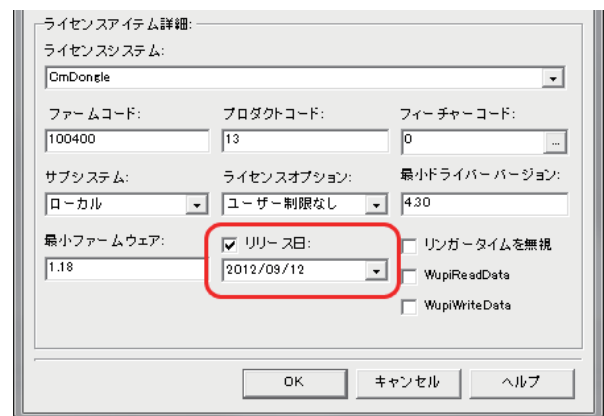
リリース日:

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に立つ機能です。

[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。



ライセンスアイテム詳細:
ライセンスシステム:
CmDongle

ファームコード:	プロダクトコード:	フィーチャーコード:
100400	13	0
サブシステム:	ライセンスオプション:	最小ドライババージョン:
ローカル	ユーザー制限なし	4.30
最小ファームウェア:	リリース日:	リンガータイムを無視
1.18	2012/09/12	<input type="checkbox"/> WupiReadData
		<input type="checkbox"/> WupiWriteData

OK キャンセル ヘルプ

☐ リンガータイムを無視

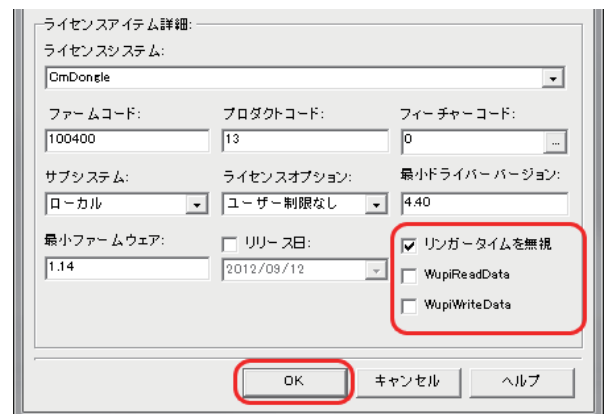
リンガータイムを無視する場合にチェックを入れます。

☐ 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にデータをライト(Write)する場合はチェックを入れます。

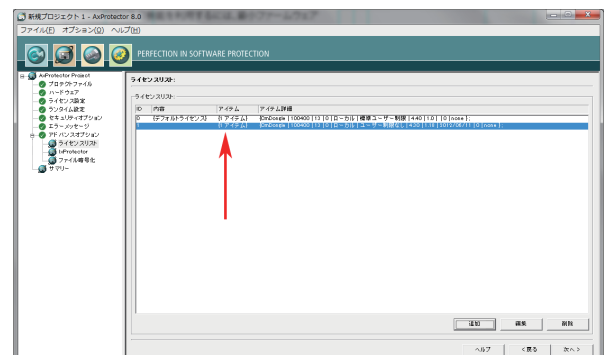


ライセンスアイテム詳細:
ライセンスシステム:
CmDongle

ファームコード:	プロダクトコード:	フィーチャーコード:
100400	13	0
サブシステム:	ライセンスオプション:	最小ドライババージョン:
ローカル	ユーザー制限なし	4.40
最小ファームウェア:	リリース日:	リンガータイムを無視
1.14	2012/09/12	<input checked="" type="checkbox"/> WupiReadData
		<input type="checkbox"/> WupiWriteData

OK キャンセル ヘルプ

「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。



PERFECTION IN SOFTWARE PROTECTION

ライセンスリスト:

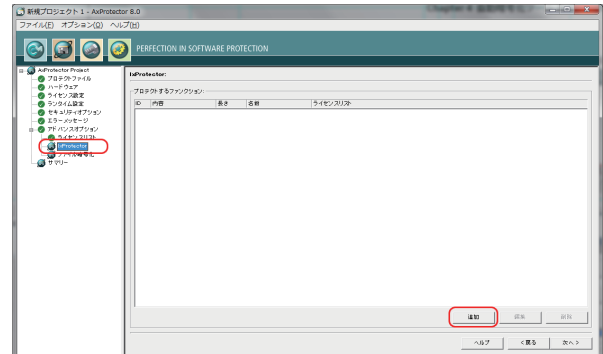
ID	内容	アイテム	アイテム詳細
10	ライセンスリスト	アイテム	アイテム詳細
11	ライセンスリスト	アイテム	アイテム詳細

通知 検索 削除

戻る 一覧 次へ

④ lxProtector を登録する

lxProtector画面で、「追加」ボタンをクリックして、lxProtector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**

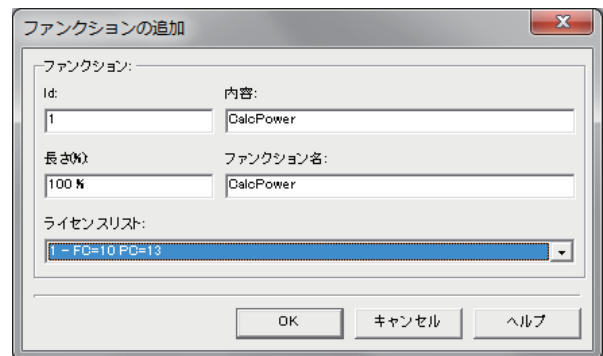
「ファンクションの追加」画面でファンクションを登録します。

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 (%):**

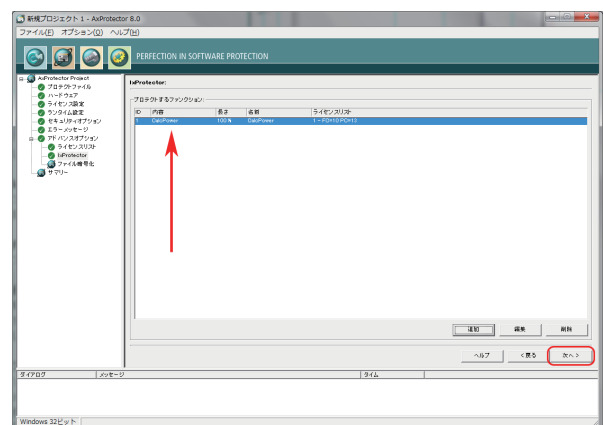
暗号化する範囲をパーセンテージ(%)で指定します。%を使用しないで、直接数値を設定すると、設定したバイト数分が暗号化されます。

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、lxProtector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

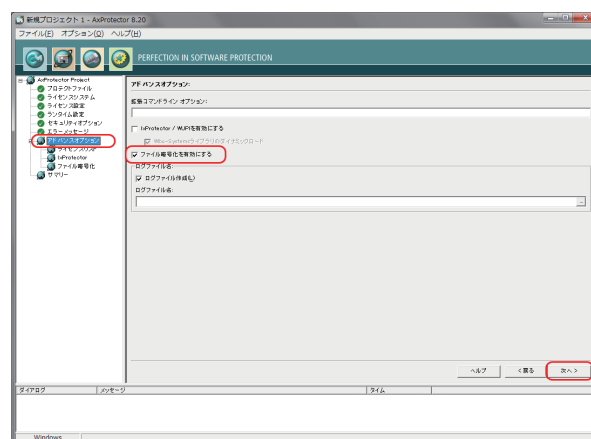
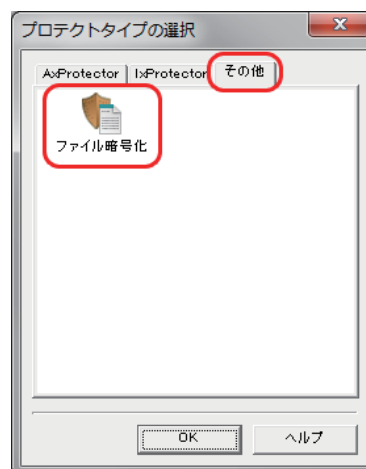
この「ファイル暗号化を有効にする」オプションは、プログラムが使用（または制御）するデータファイルをメモリ上で復号化・暗号化する機能です。復号化・暗号化はあくまでメモリ上で行われ、ディスク上のデータファイルは常に暗号化されているため、データファイル自身のコピープロテクトが可能です。

なお、データファイル自身はAxProtectorの「ファイル暗号化」で最初に暗号化しておきます。

この「ファイル暗号化を有効にする」オプションは、ここで暗号化するプログラムに「データファイルの復号化・暗号化を行う機能」を持たせるかどうかの選択を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化・暗号化は行われません。

① ファイル暗号化を有効にする

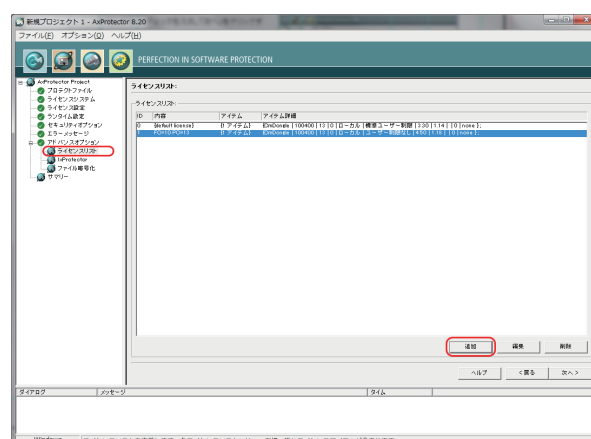
「アドバンスオプション」画面で、「ファイル暗号化を有効にする」にチェックを入れ、「次へ」をクリックする。



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画面がグレイアウトしますので、「次へ」ボタンをクリックして次に進めます。



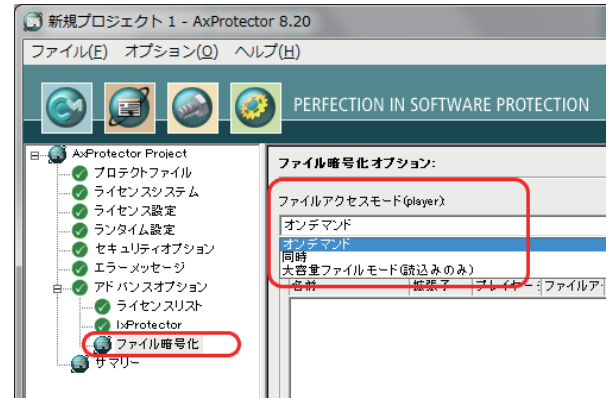
③ ファイルアクセスモードを指定する

「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。読み込みと復号化は 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化します。一度復号化されたブロックは、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、一度復号化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化します。復号化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、復号化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化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 (読込みのみ):

この「大容量ファイルモード (読込みのみ)」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化します。復号化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。

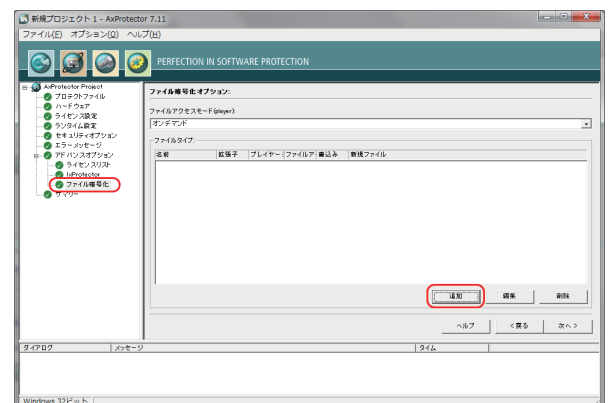
名前:

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

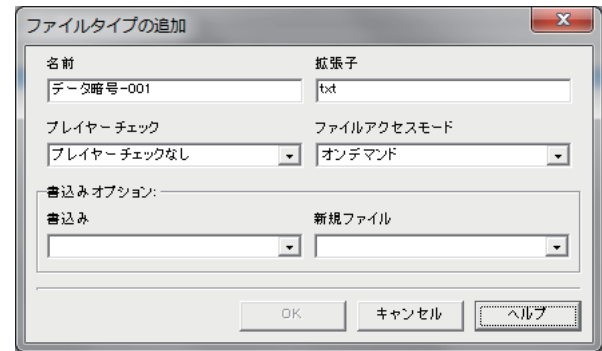


プレイヤーチェック:

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:

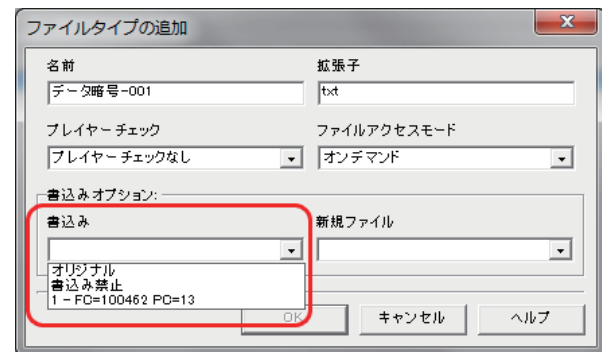
暗号化されていたファイルをプレイヤーで開いた後に編集した場合、どのようなセキュリティ属性で保存するかを指定できます。セキュリティ属性には、「オリジナル」、「書き込み禁止」、「（ライセンスリスト）」の3通りがあります。

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:

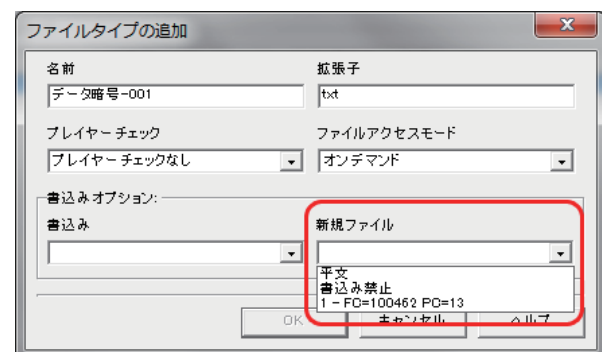
プレイヤーで新しく作成する新規ファイルのセキュリティ属性を指定します。

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

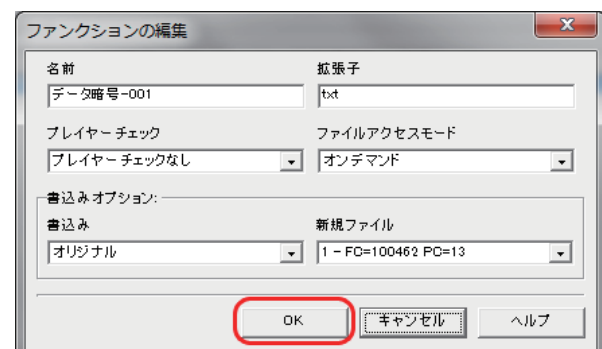
新規ファイルは作成しない。新規保存できない。



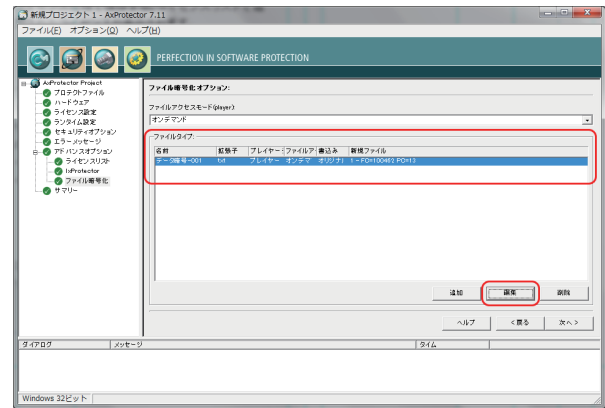
（ライセンスリスト）

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（復号化・暗号化）されます。

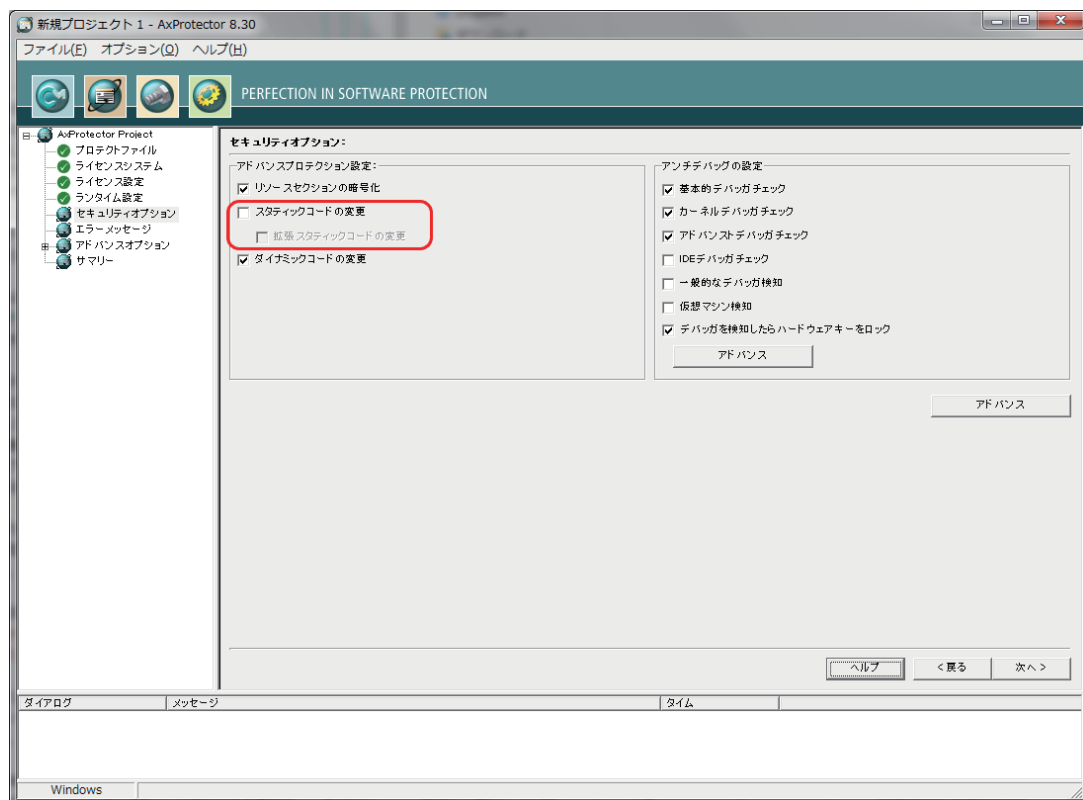


なお、登録したファイルタイプは「編集」ボタンをクリックすることで編集が可能です。



[注意]

制御するプログラムをAxProtectorで暗号化する際、「5. セキュリティオプション」で、「スタティックコードの変更」のオプションを必ずする必要があります。デフォルトではチェックが入っています。「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。



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。

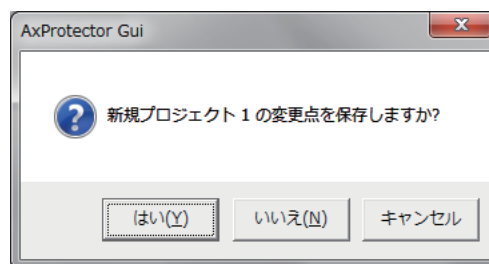
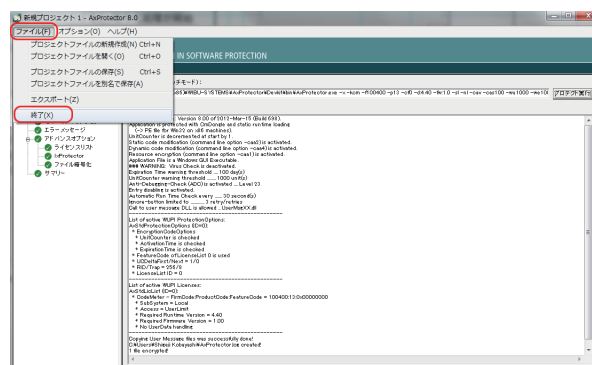
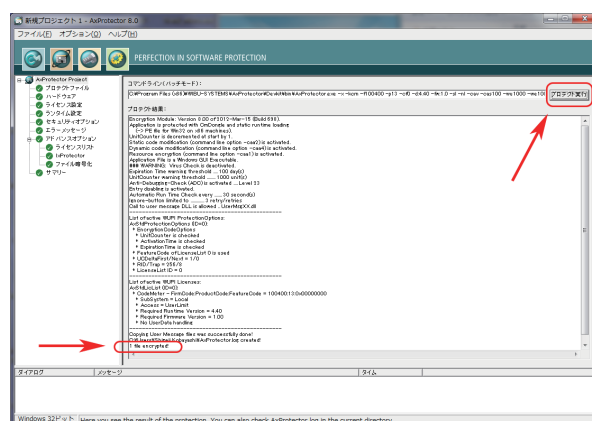
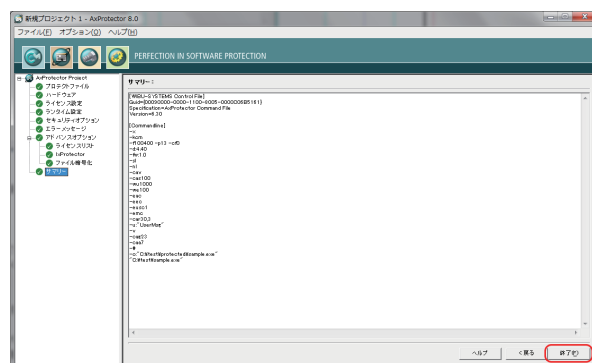
設定に問題がなければ、右下の「終了」ボタンがアクティブになります。もし、設定に問題がある場合は、「終了」ボタンが非アクティブの状態になります。

「終了」ボタンをクリックすると、暗号化処理が始まります。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。右上部の「プロテクト実行」ボタンをクリックすると暗号化処理が始まります。

[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。

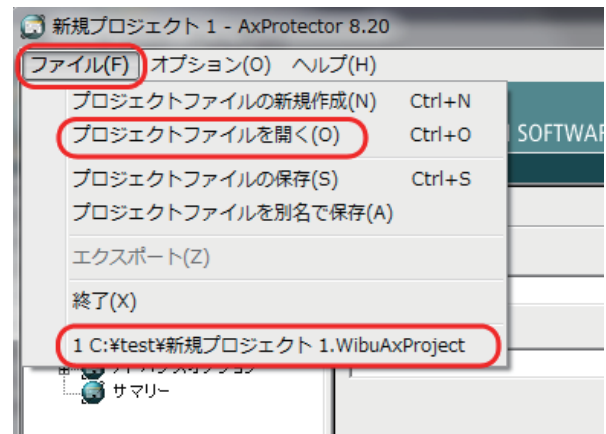


9. プロジェクトファイルを開く

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。



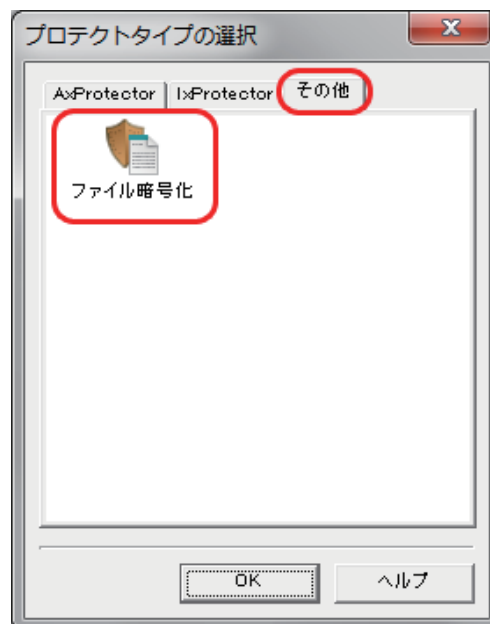
4-5. データファイルを暗号化する

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。

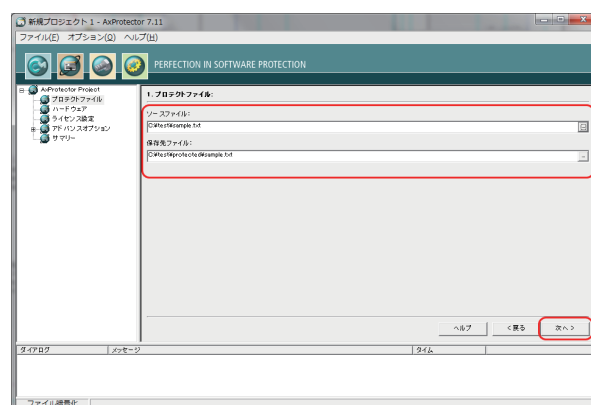
ただし、暗号化されたファイルを開くプログラム(exe等)にあらかじめAxProtectorで暗号化処理する必要がありますので、Excel.exeやMedia Player等、ライセンス上暗号化処理ができない場合は例外になります。

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。



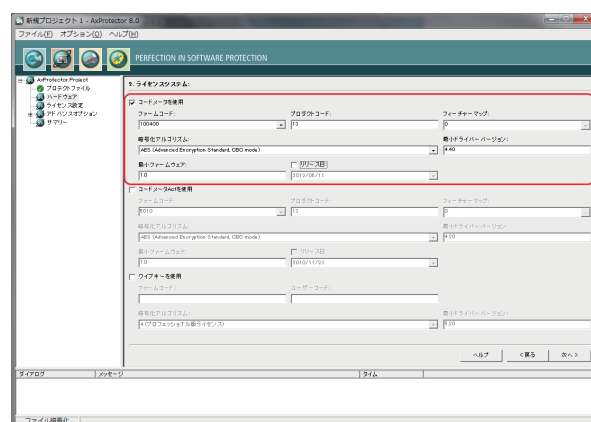
① ファイル名を入力する

「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、フィーチャーマップ、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)、最小ファームウェアバージョン、リリース日を設定します。リリース日の設定は任意です。



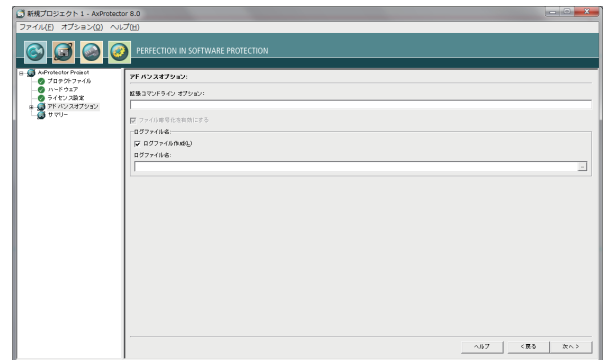
③ ライセンス設定を行う

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。



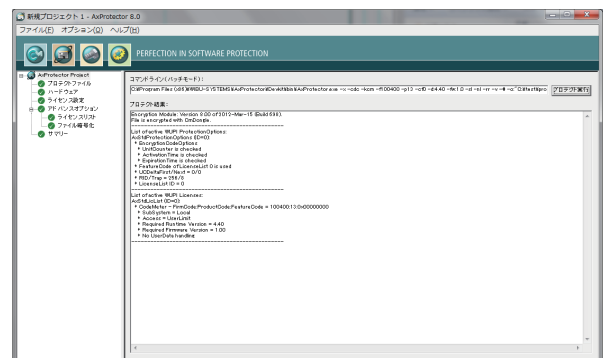
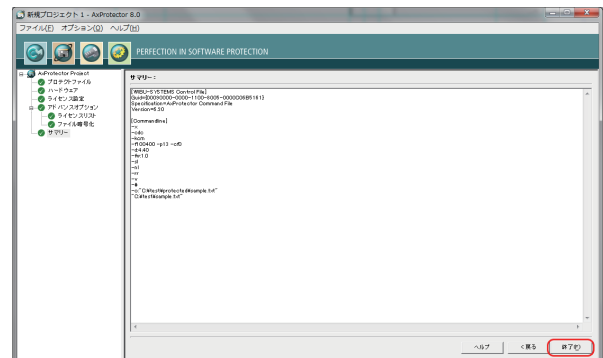
④ アドバンスオプション

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。



4-6. コマンドラインでの使用方法

AxProtectorをコマンドラインで使い、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、ワイブキーによる暗号化[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: ワイブキーを使用

-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)

<I> 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

<I> 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CM-Stick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CM-Stick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|I|R)],[E(C|I|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: CM-Stick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

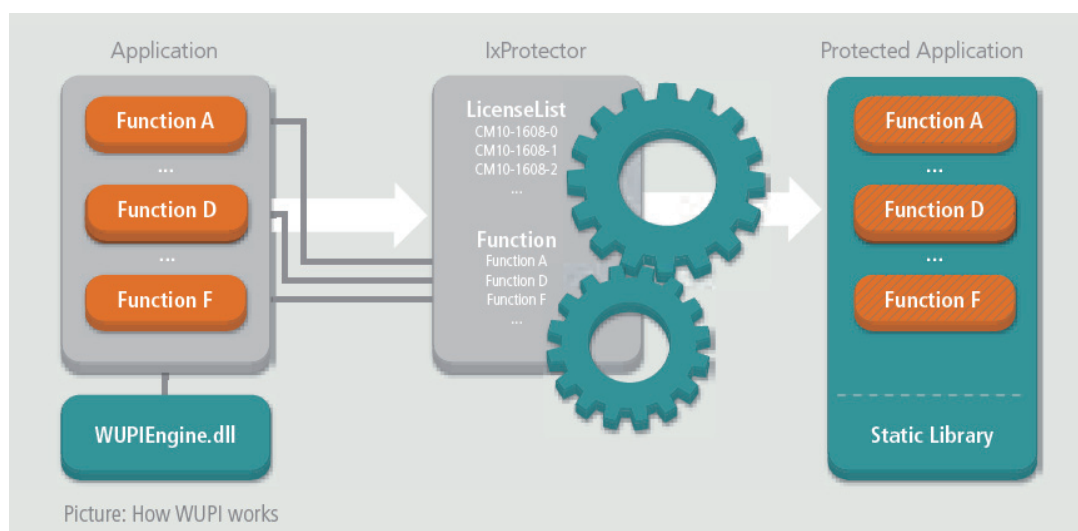
5-1. IxProtector とは

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化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、復号化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化されていること自体がセキュリティホール の 1 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を復号化し、実行したあとは再び暗号化しておくという、メモリー上での「オンデマンド復号」を実現する新機能です。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・復号化を行います。

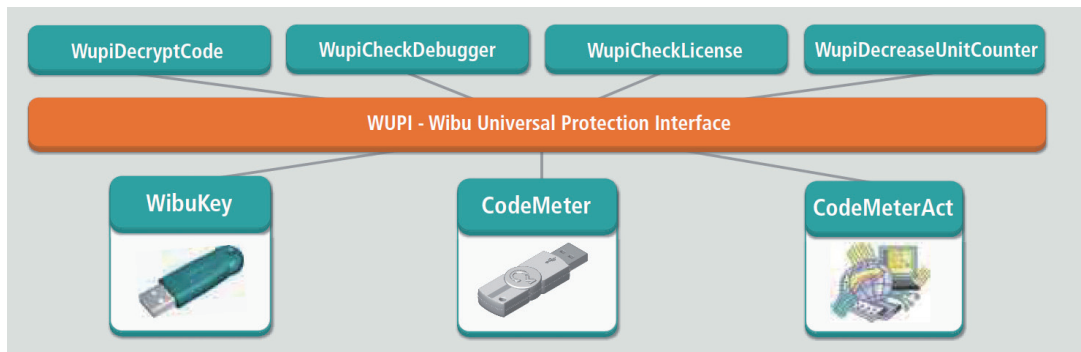


また、.NETアセンブリおよびJavaアプリケーションの場合、AxProtectorで自動暗号化することにより、このIxProtector/「オンデマンド復号」機能が自動的に組み込まれます。従い、WUPIファンクションを使用するためにソースコードを修正する必要がありません。

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CmActLicenseI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CmActLicenseI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

WUWupiGetLastError

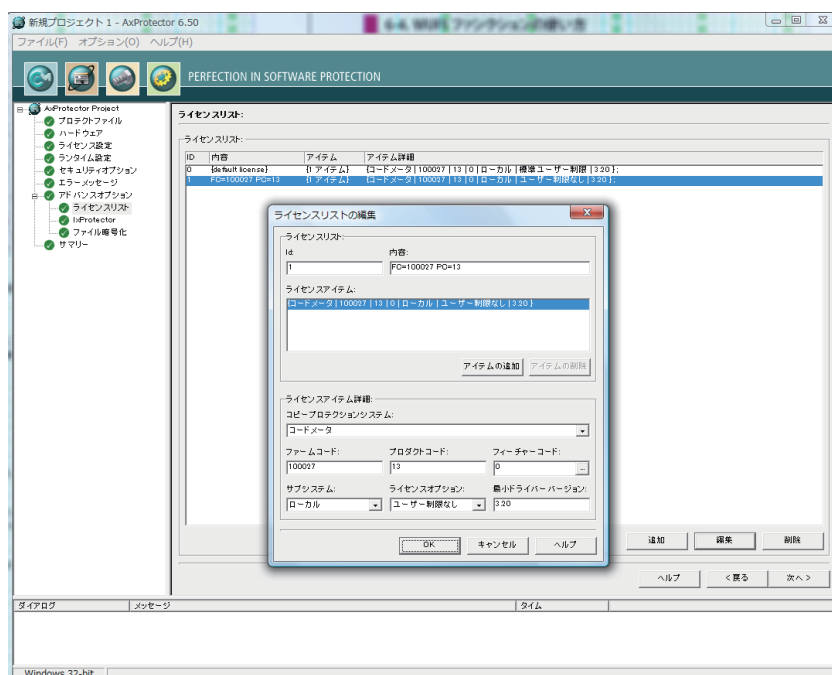
直前に実行したWUPIファンクションのエラーコードを返します。

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。

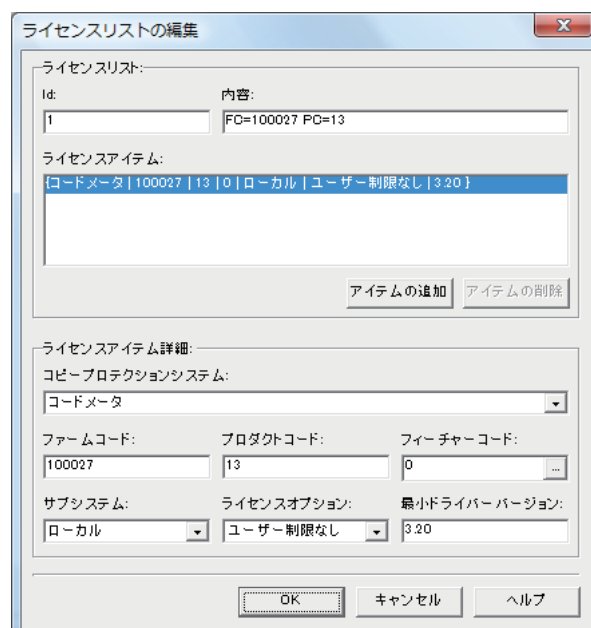


① ライセンスリストを作成する

「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。

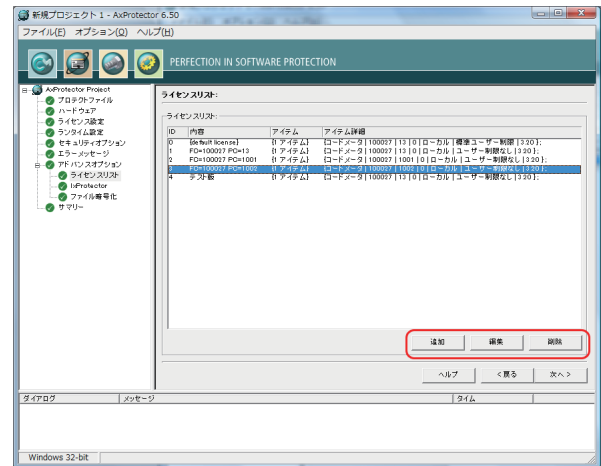
ここで作成されたライセンスリストが、WUPIファンクションから参照されます。

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、

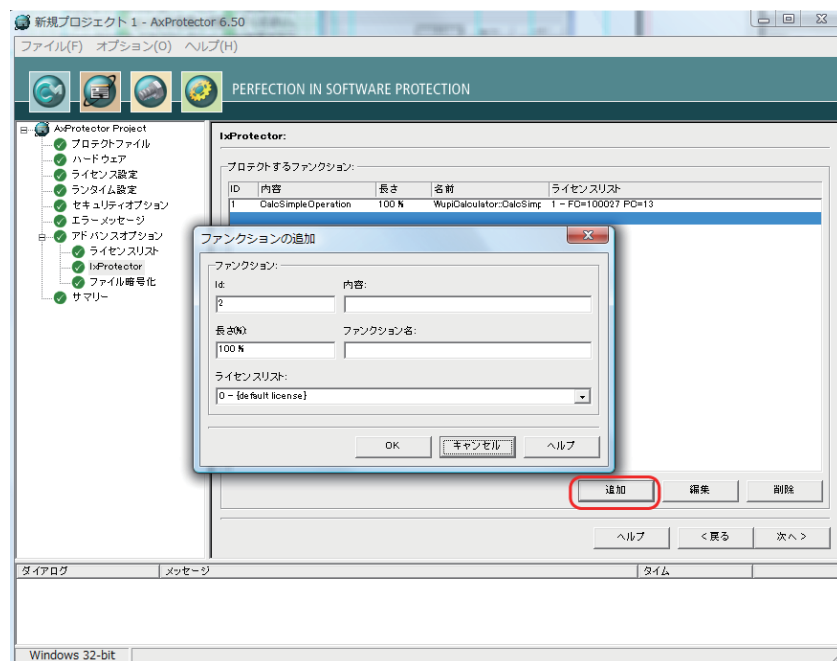
WupiCheckLicense(1)

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。



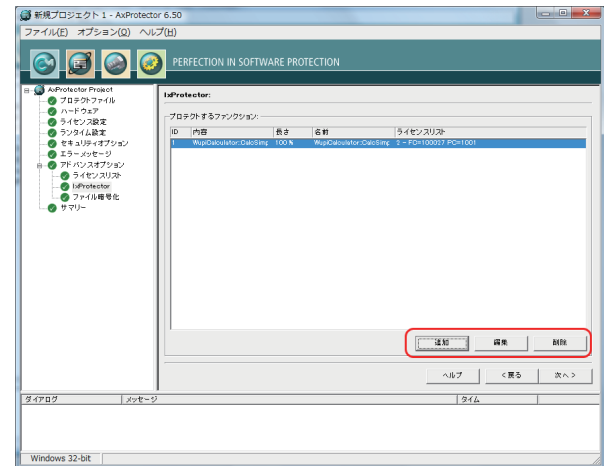
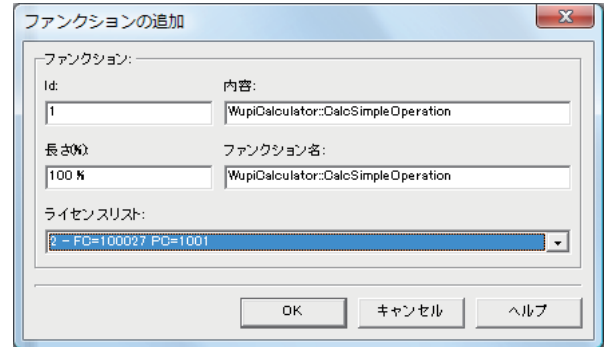
① ファンクション定義を行う

「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化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化します。(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。(¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥Samples¥IxProtectorの中にあります。(C/C++, Delphi)

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。(¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥Samples¥IxProtectorの中にあります。(C/C++, Delphi))

[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

[WUPI Accessing API]

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。

[WUPI Encryption / Decryption API]**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。

[WUPI Management API]

WupiDecreaseUnitCounter

[内容]

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。

[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。

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。

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 (リミットカウンター) の値、CodeMeterの場合はUnit Counter (ユニットカウンター) 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。

WupiReadData

[内容]

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。

[Syntax]

```
int WupiReadData (int iLicenseList, int iOffset, void* pvData, unsigned int cdData);
```

[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。

iOffset

リードするバイト数

pvData

データアレイ

cbData

pvDataのバイト数

[リターン値]

pvDataに格納されたバイト数を返します。

リターン値が0の場合、WupiGetLastErrorでエラー内容を取得してください。

WupiReadDataInteger

[内容]

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、前述のWupiReadDataを利用することを推奨します。

[Syntax]

```
int WupiReadDataInteger (int iLicenseList, int iOffset);
```

[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。

iOffset

リードするバイト数

[リターン値]

リターン値 4 バイトは、上位 2 バイトと下位 2 バイトに分かれます。上位 2 バイトはエラーとメッセージハンドリングで、下位 2 バイトはデータになります。上位 2 バイトは以下の値を持ちます。

```
#define WupiRDError (0x80000000)
```

```
#define WupiRDMoreDataAvail (0x40000000)
```

WupiWriteData

[内容]

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CmActLicenseには未対応です。

[Syntax]

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

[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。

iOffset

リードするバイト数

pvData

データアレイ

cbData

pvDataのバイト数

[リターン値]

エラーの場合、FALSE (0)が返ります。エラー以外の場合はTRUE (1)が返ります。

WupiWriteDataInteger

[内容]

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。2 バイト単位でデータをライト (Write) します。ポインタが使用できる言語の場合、前述の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CmActLicenseには未対応です。

[Syntax]

```
int WupiWriteDataInteger (int iLicenseList, int iOffset, int iData);
```

[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。

iOffset

リードするバイト数

iData

ライト (Write) するデータ

[リターン値]

エラーの場合、FALSE (0)が返ります。エラー以外の場合はTRUE (1)が返ります。

[WUPI Security API]

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。

[WUPI Error API]

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、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。

[スタート]-[すべてのプログラム]-[CodeMeter]-[Samples]-[Software Protection]

でアクセスできます。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, VBasic6.w32

[NOTE]

1. VBasic.NETの場合、"WibuCmNET.dll"をプロジェクトの参照設定で追加してください。"WibuCmNET.dll"は、¥Program Files¥CodeMeter¥DevKit¥assembliesフォルダにあります。

2. Javaプログラムの場合：

コードメータAPIを組み込んだJavaプログラムを、自動暗号化ツールAxProtectorで暗号化する際、以下の点にご注意ください。

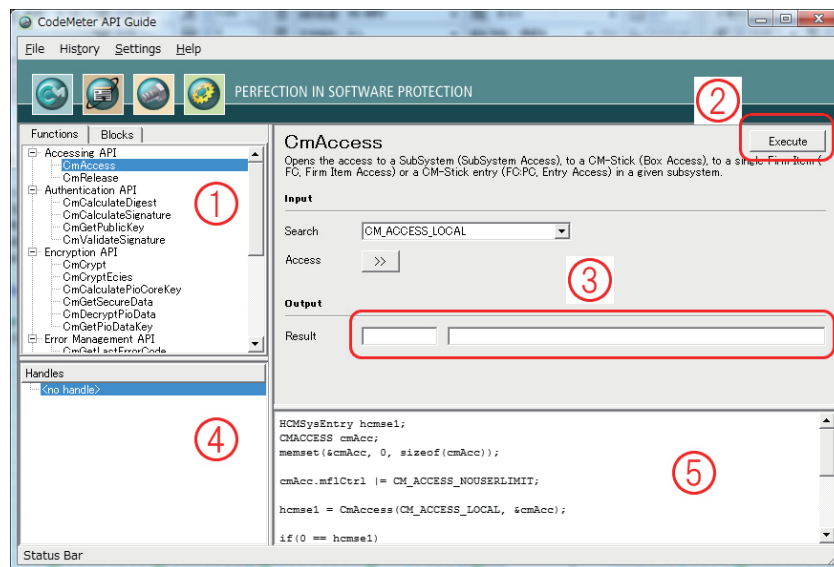
- jarファイルには必ずマニフェストファイルを含めてください。
- マニフェストでMain-Classを指定しない場合、「Java Options」ページのMain Classを必ず指定してください。
- AxProtectorで暗号化すると、jarファイルに含まれるすべてのクラスファイルの内容が暗号化されます。そのため、jarファイルに含まれるクラスを個別に呼び出すことができなくなります。
- AxProtectorで暗号化したjarファイルにEXEファイル化するラッパーツール(exewrap、launch4jなど)を適用することはできません。EXEファイル化する場合は、まずjarファイルをEXEファイル化してから、AxProtectorで暗号化してください。

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をPCに装着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

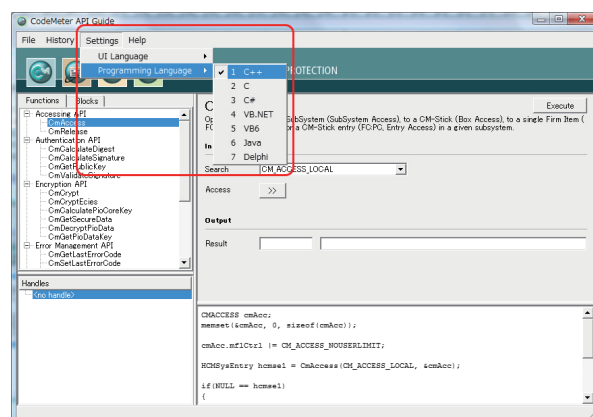
【スタート】→【すべてのプログラム】→【CodeMeter】→【Tools】→【CodeMeterApiGuide】をクリックして起動します。



- 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。

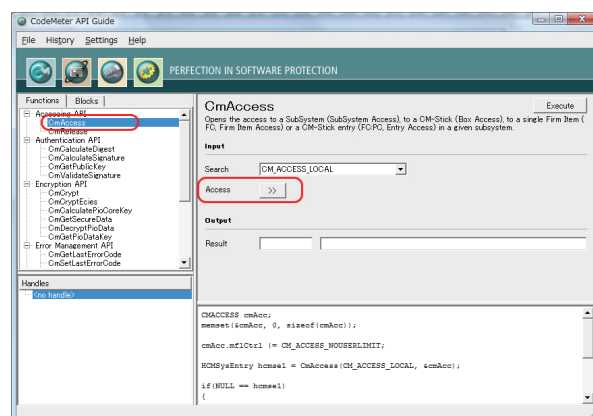
6-4. CodeMeter API ガイドの使用例

実際に ファームコード=10、プロダクトコード=13 のCmContainer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。

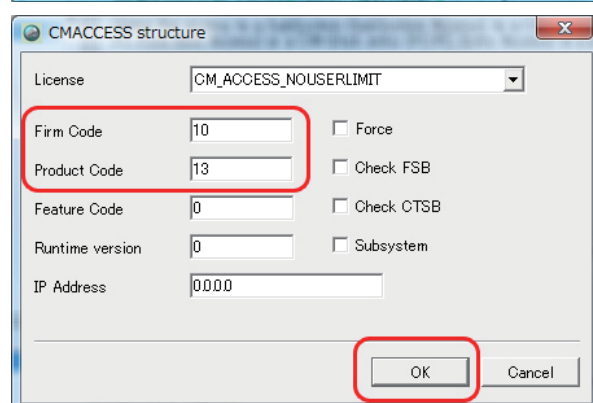


特定のファームコードとプロダクトコードが設定されたCM-Stickがあるかどうかをチェックするだけであれば、CmAccess() → CmGetLasetErrorCoe() → CmRelease() を行うだけで実現できます。CodeMeter API ガイドで実際に実行してみましょう。

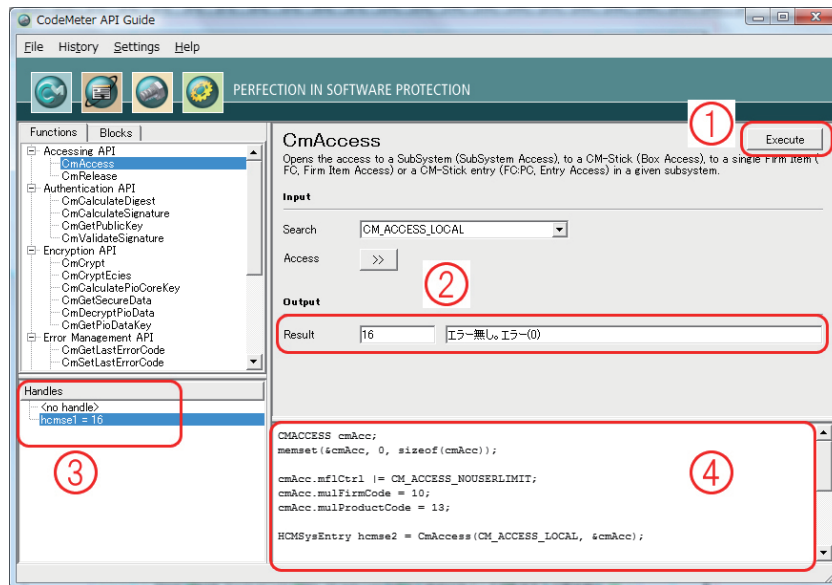
①FirmCode=10、ProductCode=13 が登録されたCmContainerをPCに装着し、CmAccess を選択します。Access項目の[>>]をクリックし、CMACCESS ストラクチャ画面を開きます。



②CMACCESS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。

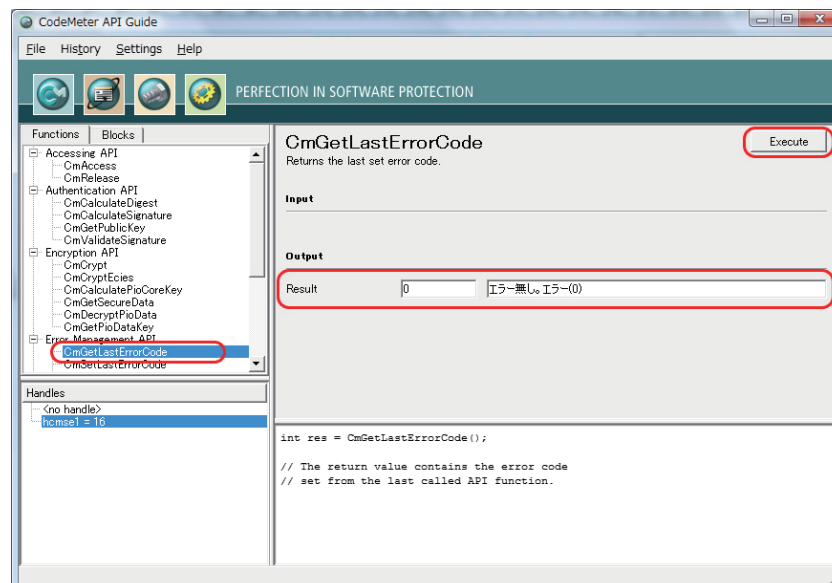


③ CodeMeter APIガイド画面に戻り、「Execute」ボタンをクリックすると、CMACCESSストラクチャで設定した内容でCmAccessファンクションが実行され、その結果がResult欄に表示されます。(②) CmAccessファンクションが返したハンドルは、「Handles」欄に表示されます。(③) 実行されたソースコードは(④)に表示されます。

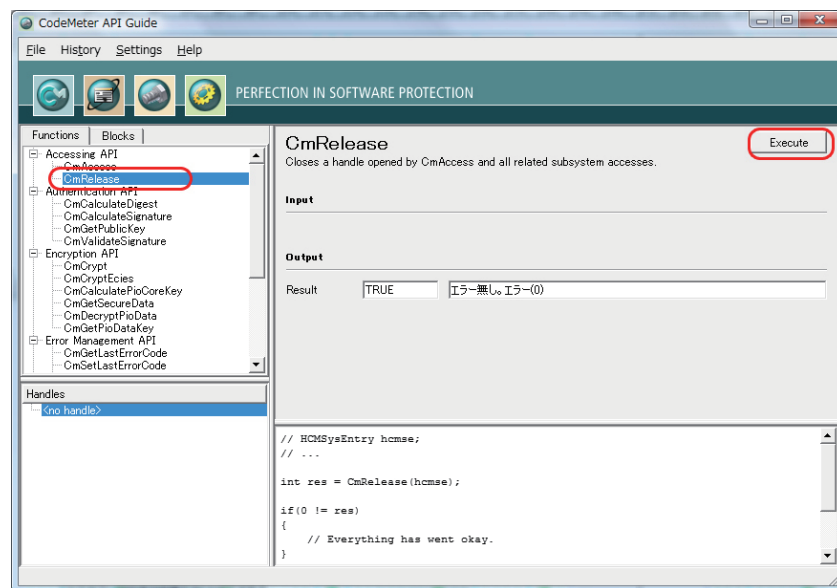


実行に成功すると、リターン値に0以外の整数値が返り、実行エラーの場合は、リターン値に0が返ります。(②) リターン値が0であるとエラーが発生したことはわかりますが、どのようなエラーが発生したかまではわかりません。どのようなエラーが発生したかを知るために、CmAccessファンクションに続けてCmGetLastErrorCodeファンクションを実行します。このCmGetLastErrorCodeファンクションは、直前に実行したファンクションのエラーコードを取得するファンクションです。

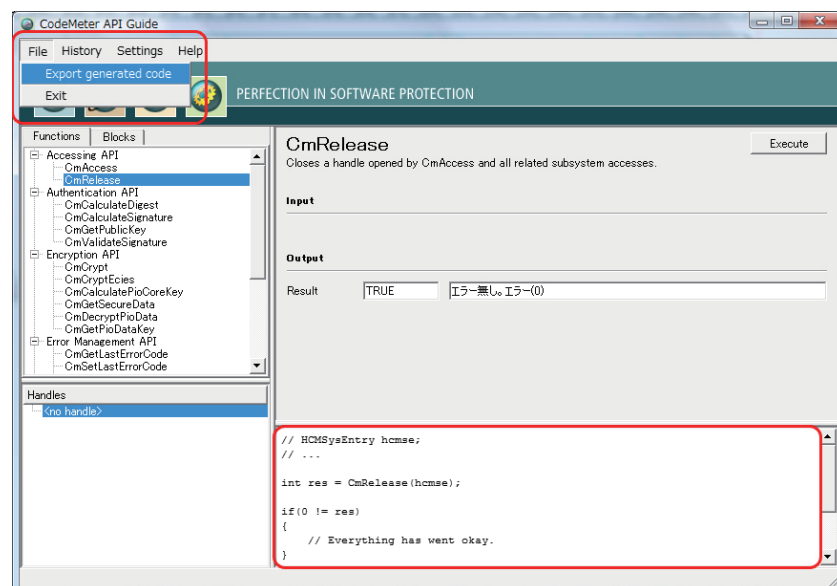
④ Functions一覧から、CmGetLastErrorCodeを選択し、「Execute」ボタンをクリックします。エラーが無い場合、リターン値に0が返ります。エラーの場合は、そのエラーコードが返ります。エラーコードを調べることでエラーの内容を把握することができます。



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、CmGetLastError、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。

Enabling API

CmEnablingWriteApplicationKey

Access CodeをEnableする。

CmEnablingGetApplicationContext

アクティベート/非アクティベートするアプリケーションの内容をリード (Read) する。

CmEnablingGetChallenge

アクティベート/非アクティベートするCmContainerからセッションIDをリード(Read)する

CmEnablingSendResponse

CmContainerまたはCmContainerエントリをアクティベートする。

CmEnablingWithdrawAccessRights

CmContainerまたはCmContainerエントリを非アクティベートする。

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**

直前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

直前のエラーテキストを取得する。

CmConvertString

入力されたストリングを指定したcharacter setに変換します。

CmGetLastErrorText2

直前のエラーテキストを取得する。取得するテキストの言語は使用中のシステムOSに依存し、対応していない場合は英語(English)で取得する。CM_GLET_DIALOGフラグを立てた場合、エラーテキストと参考情報がメッセージボックスに表示される。(Windows / Windows CE のみ)

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。

CmGetLicenseInfo

ライセンサーサイトにあるCmFirm.wbcファイルに格納されているライセンスを取得する。

CmRevalidateBox

使用中のCmContainerの検証を行う。このAPIは、ランタイムチェックのようにライセンス自体をチェックするのではなく、CmContainerが利用可能か、有効かどうかをチェックする。実行には、対称暗号作業がともなうため、多少時間がかかる。CmActLicenseの場合、CmActLicenseが壊れているかどうかをチェックする。CmRevalidateBox()をCallすると、CmActLicenseの"replug"が発生するため、検証が再度行われることになる。CmDonglesの場合、複数の内部キーを利用して検証が行われる。

Programming API

CmProgram

CmDongleのエントリをプログラムする。下記の行為が可能である。

- 新しいファームアイテムを追加する。
- 既存のファームアイテムを更新する。
- 既存のファームアイテムを削除する。
- 既存のファームアイテムに対して新しいプロダクトアイテムを追加する。
- 既存のプロダクトアイテムを更新する。
- 既存のプロダクトアイテムを削除する。
- CmDongleのBox Control レジスターに新しい値を設定する。
- CmDongleをロックまたは解除する。
- FSBエントリに新しいファームキー(Firm Key)をプログラムする。
- CmDongleに新しいユーザーキー(UserKey)をプログラムする。

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。

CmGetRemoteContextBuffer

このファンクションはCmGetRemoteContext()と同じように動作しますが、ファイルポインタの代わりにプリアロケートメモリバッファを使用します。

CmListRemoteUpdateBuffer

このファンクションはCmListRemoteUpdate()と同じように動作しますが、ファイルポインタの代わりにプリアロケートメモリバッファを使用します。

CmSetRemoteUpdateBuffer

このファンクションはCmSetRemoteUpdate()と同じように動作しますが、ファイルポインタの代わりにプリアロケートメモリバッファを使用します。

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。

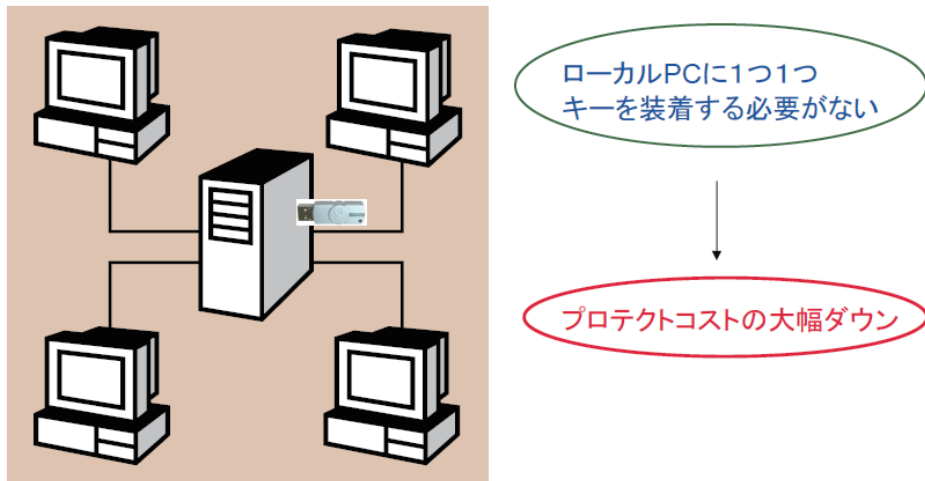
Chapter 7

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ーの登録方法
- 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(CM-Stick)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。CM-Stickのネットワークカウンター(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4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(CM-Stick)を装着するサーバーは、必ずしも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CM-Stickに登録できるネットワークカウンター(Network Counter)値は最大値65,536までです。理論的には、1つのCM-Stick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、CM-Stick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CM-Stickを置くことも可能です。

7-2. ネットワークカウンターの登録方法

コードメータキー (CM-Stick) にネットワークカウンターを登録するには、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(CM-FSB)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 CM-Stick と CM-FSB を PC に装着する

ネットワークカウンタを登録するCM-Stickと、貴社のコードメータFSB(CM-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、「編集」メニューを選択します。

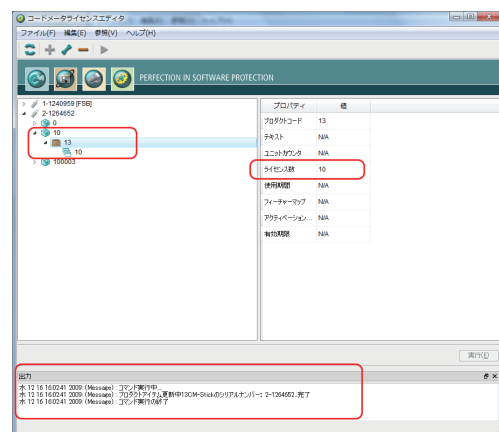
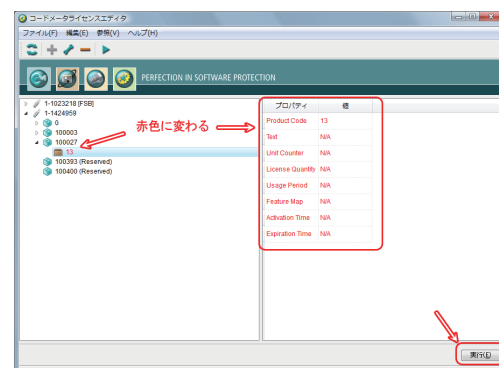
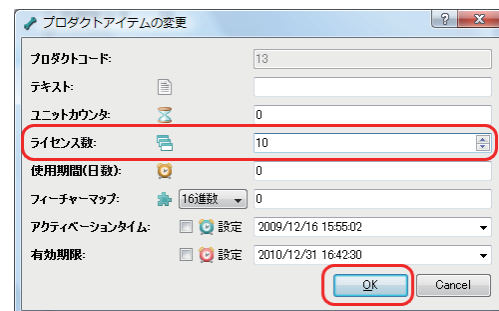
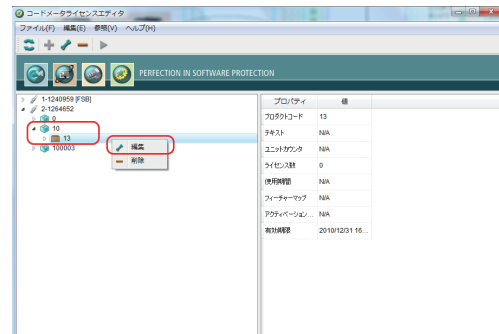
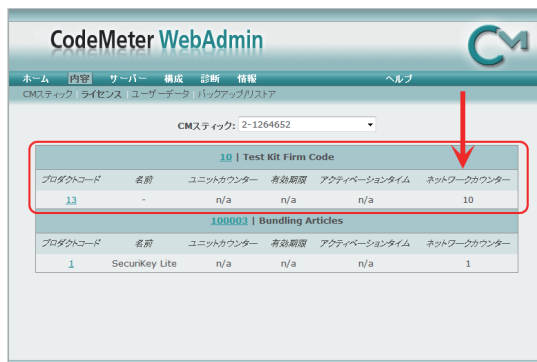
③ 「ライセンス数」を設定

「プロダクトアイテムの変更」画面で、「ライセンス数」に許可する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

ライセンスエディタ画面で、編集を行うプロダクトコードおよび右部のプロパティが赤色に変わっています。右下の「実行」ボタンをクリックすると、CM-Stick にネットワークカウンターが追記されます。

WebAdmin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 CM-Stick と CM-FSB を PC に装着する

ネットワークカウンタを登録するCM-Stickと、貴社のコードメータFSB(CM-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】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。

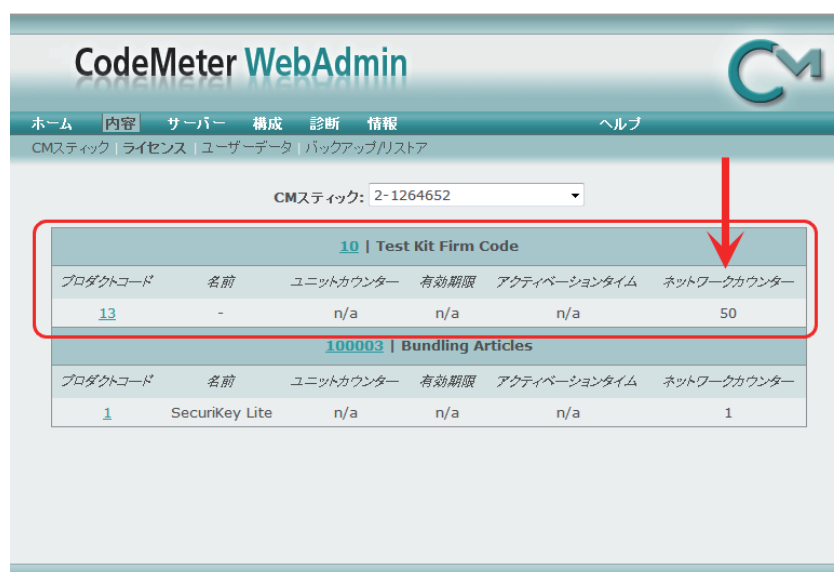
¥Program Files¥CodeMeter¥DevKit¥bin

【例】 コマンドプロンプトを開いて、
>CD ¥Program Files¥CodeMeter¥DevKit¥bin ↓

④ ネットワークカウンターを登録する

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ー= 50 を新規登録します。ネットワークカウンター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で追加されます。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。

また、既存のプロダクトアイテムのネットワークカウンターを変更する場合は、

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ーが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」を参照してください。

7-3. コードメータサーバーの起動方法

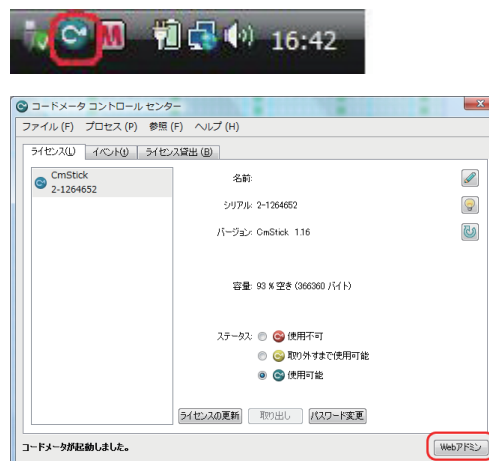
コードメータサーバーは、WebAdminのネットワーク設定から行います。コードメータランタイムキットをインストールするとこの機能も自動的にインストールされます。コードメータサーバーは、必ずしもネットワークを実際に管理するサーバーを指定する必要がなく、クライアントPCの1台をコードメータサーバーにすることが可能です。どれをサーバーにするかは、アクセス負荷を考慮の上、指定してください。

① コードメータサーバーに CM-Stick 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 WebAdmin 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 WebAdminを起動します。

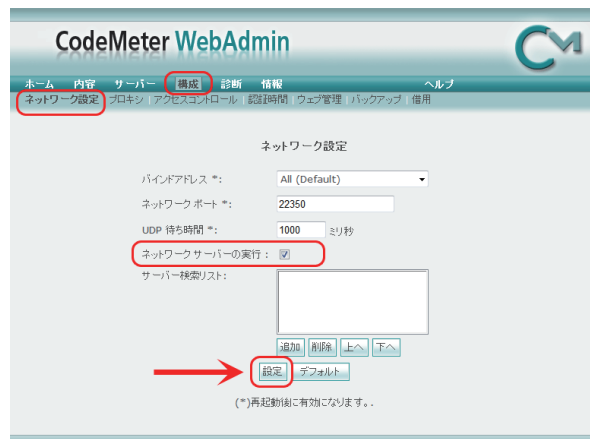


「構成」メニューをクリックし、「ネットワーク設定」画面を開きます。



③ ネットワークサーバーを設定する。

「ネットワーク設定」画面で、「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。

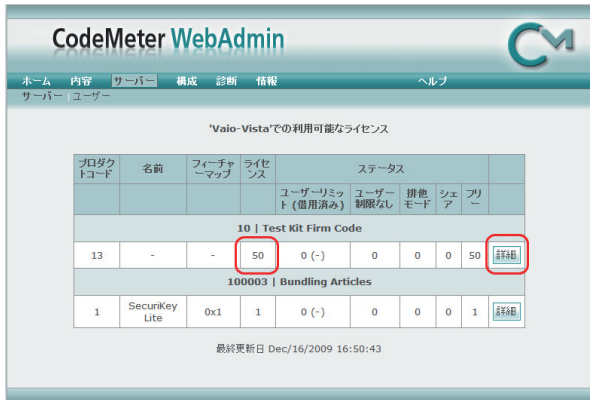


「設定」ボタンをクリックすると、右の画面が表示されネットワークサーバーが設定されます。



④ライセンス数を確認する。

WebAdminの「サーバー」/「サーバー」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。



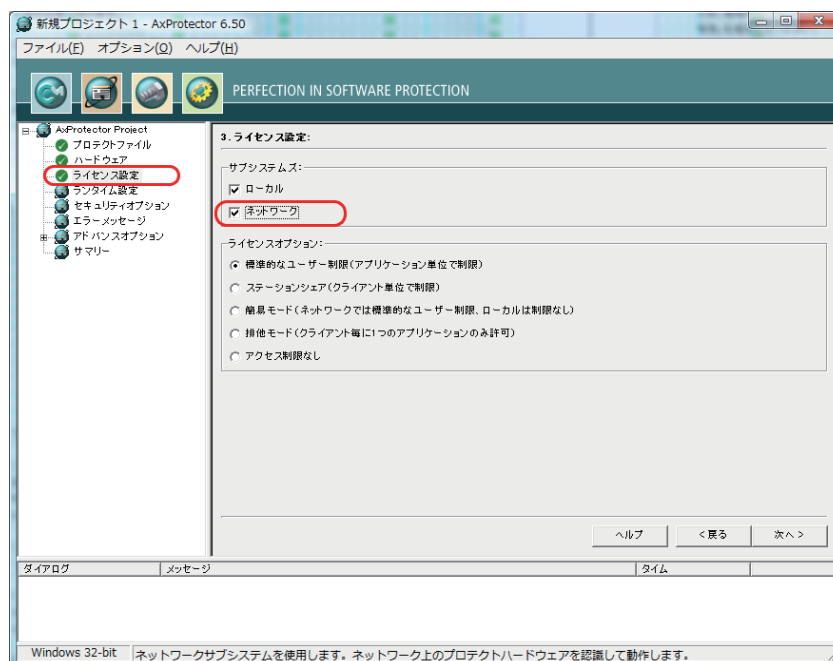
7-4. ネットワーク対応型プロテクトの作成方法

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「ライセンス設定」の「サブシステムズ」で「ネットワーク」にチェックを入れる。
- 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「ネットワーク」を指定する

自動暗号化ツール「AxProtector」の「ライセンス設定」の「サブシステムズ」で「ネットワーク」にチェックを入れます。「ライセンスオプション」は、貴社のライセンスポリシーに応じて選択してください。



ライセンスオプション:

○ 標準的なユーザー制限(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。

○ 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。

○ 簡易モード(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「標準的なユーザー制限」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。

○ 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)

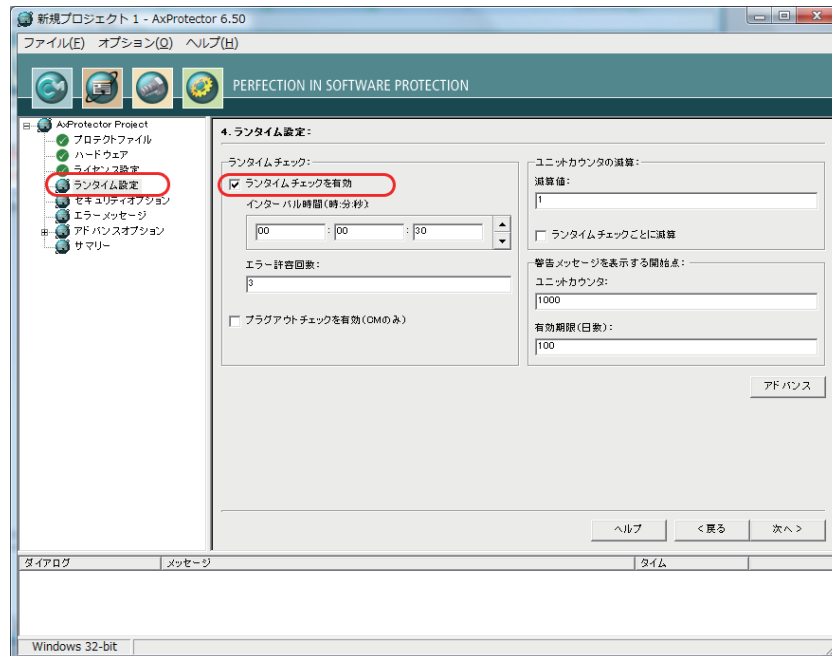
同一クライアント上でのアプリケーションの重複起動を防止します。

○ アクセス制限なし(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クラスターが解放されるため、使用中ライセンスが復元されてライセンス制御ができなくなります。必ず、「ランタイムチェックを有効」を指定してください。



[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。

Chapter 8

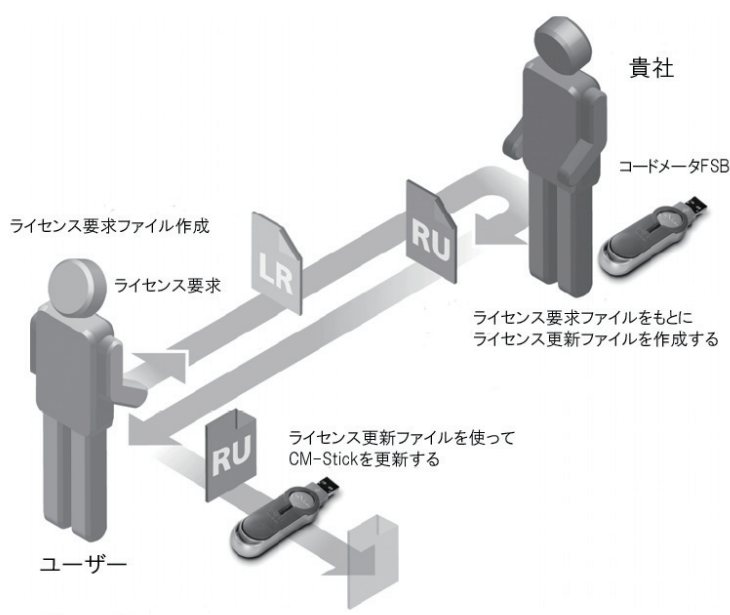
リモートアップデート機能について

- 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. CM-Stick 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 (CM-Stick) 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 (CM-Stick) の内容を変更するには、コードメータFSB (CM-Firm Security Box) 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 (CM-Stick) 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 (CM-Stick) 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。

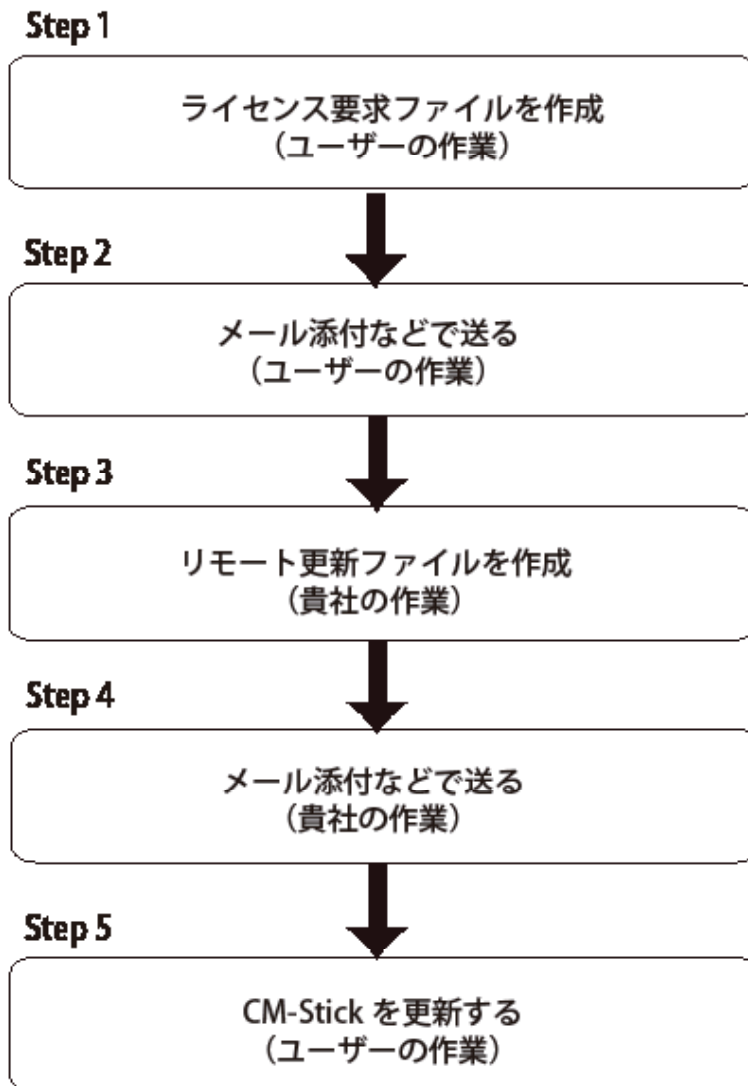


ユーザーに配布したライセンス更新ファイルは指定したコードメータキー (CM-Stick) に対して 1 回だけ使用可能です。同じコードメータキー (CM-Stick) に対して 2 回以上使用したり、別のコードメータキー (CM-Stick) に使用することができないため、確実にセキュリティを確保できます。

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 (CM-FSB) 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、CM-Stick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には貴社のコードメータFSB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CM-Stickの内容を更新します。



8-3. ライセンス要求ファイルの作成（ユーザー側）

ライセンス要求ファイルは、コードメータコントロールセンターから作成します。

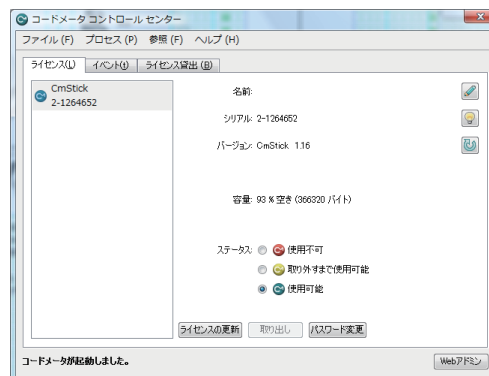
① 更新する CM-Stick を PC に装着します。

更新するCM-StickをPCに装着します。



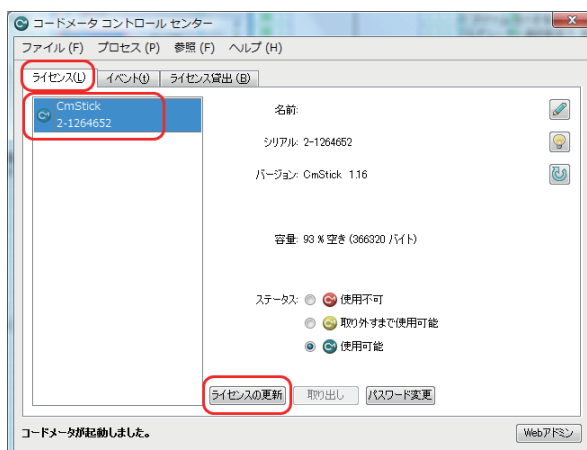
② コードメータコントロールセンターを開く

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。



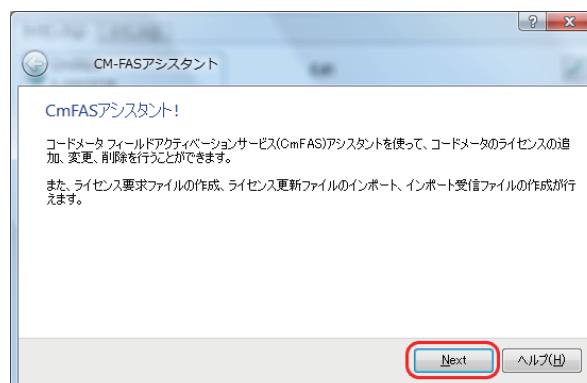
③ 「ライセンスの更新」をクリックする

「ライセンス(L)」のタブを選択し、更新するCM-Stick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。



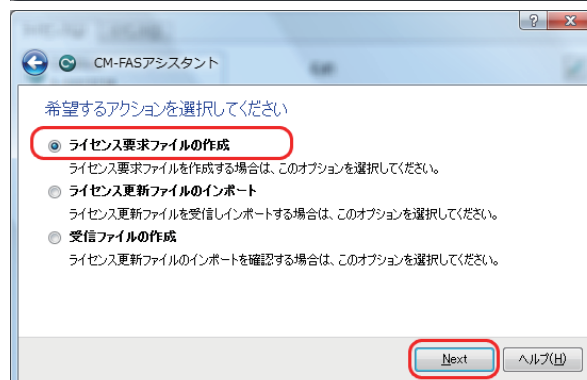
④ CM-FAS アシスタント画面が表示される

「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

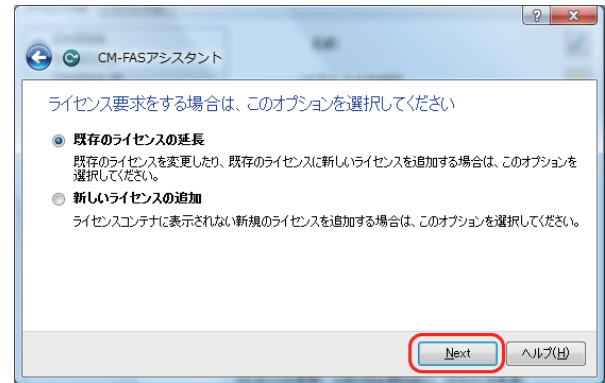
「既存のライセンスの延長」とは、すでに登録されているファームコードに対してプロダクトの追加や有効期限の更新などを行うオプションです。

「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]

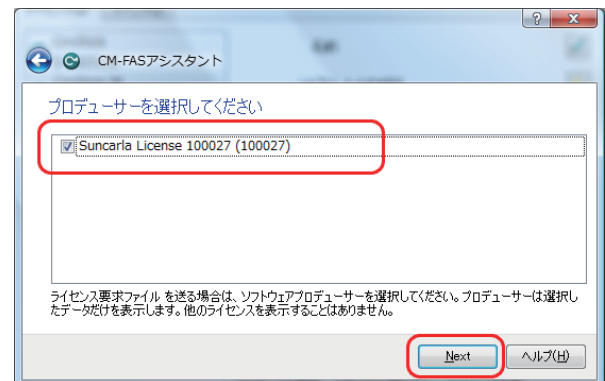
すでに登録されているファームコードに対して、新規のプロダクトコードを追加する場合は、「既存のライセンスの延長」を選択します。

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

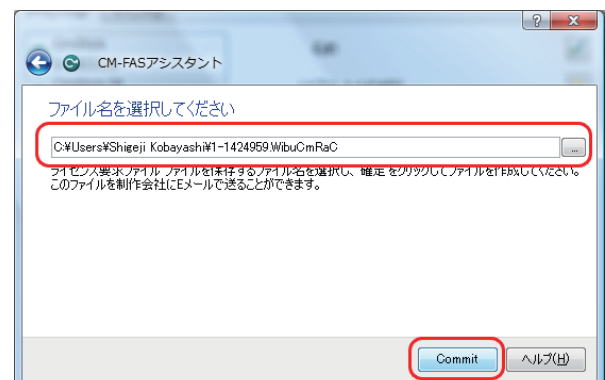
プロデューサ選択画面で、更新を要求するファームコードを選択し、「Next」ボタンをクリックします。異なるファームコードが複数表示されている場合は、更新を要求するファームコードを選択してください。



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。拡張子を変更しないでください。また、ファイル名の左部には、CM-Stickのシリアル番号が表示されます。ファイル名は、デフォルトのままで使用されることをお勧め致します。

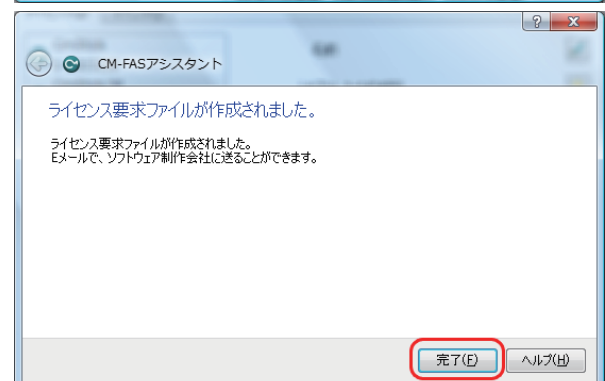
設定後、「Commit」ボタンをクリックします。



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

CM-Stick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。

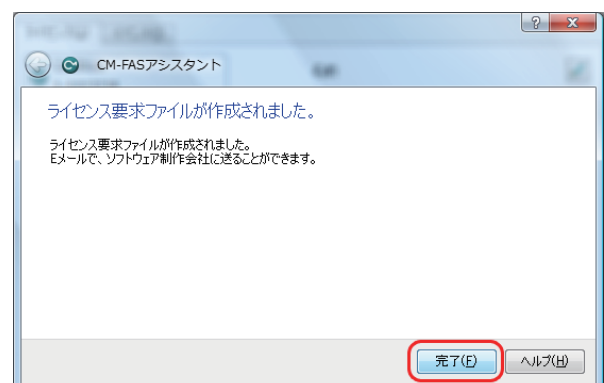
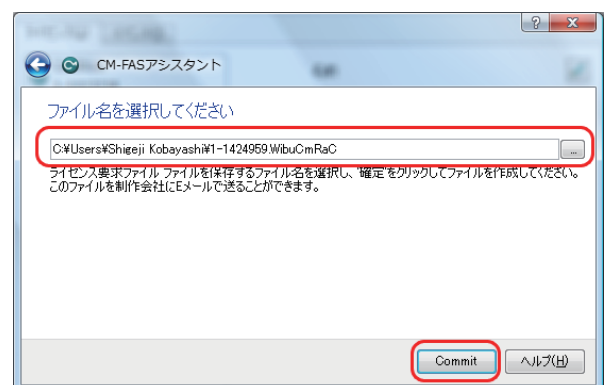
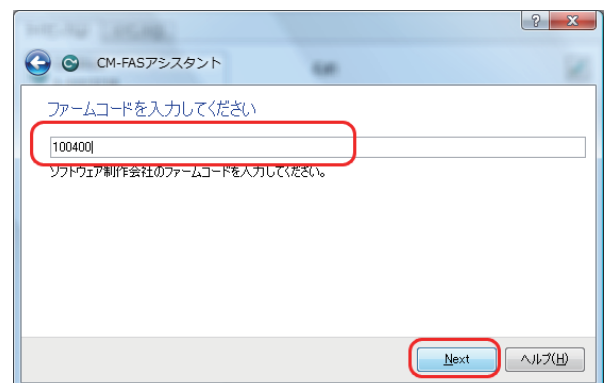
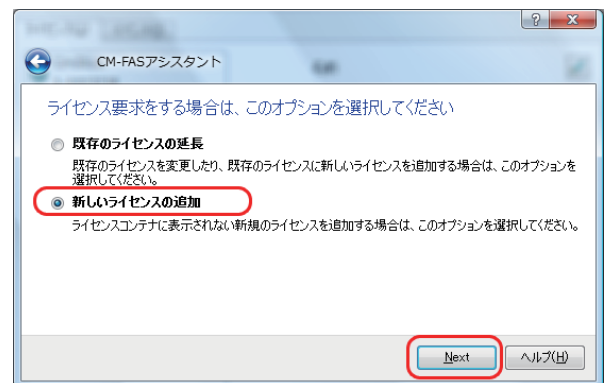
ファームコード入力画面で、要求するファームコードを半角数字で入力し、「Next」ボタンをクリックします。(ここでは、ファームコード= 100400を要求)

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、CM-Stick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。

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

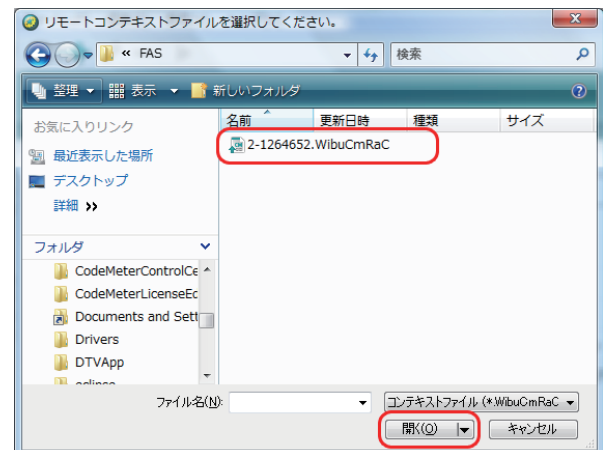
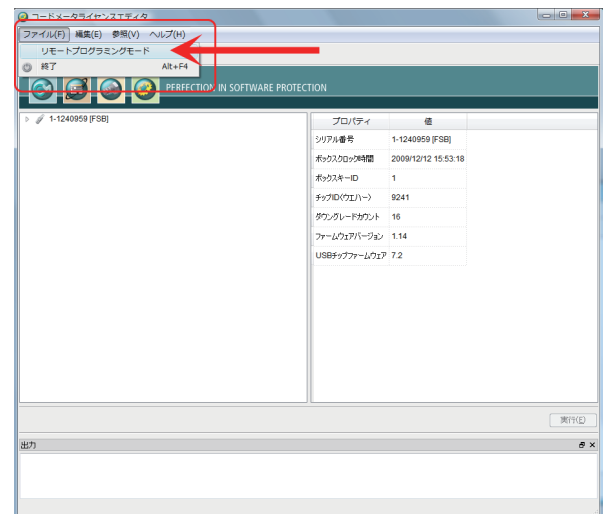
コードメータライセンスエディタを使って、ライセンス更新ファイルを作成します。ライセンス更新ファイルを作成するには、貴社のコードメータFSBが必要になります。

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、「ファイル」メニューから「リモートプログラミングモード」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の選択画面で、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。



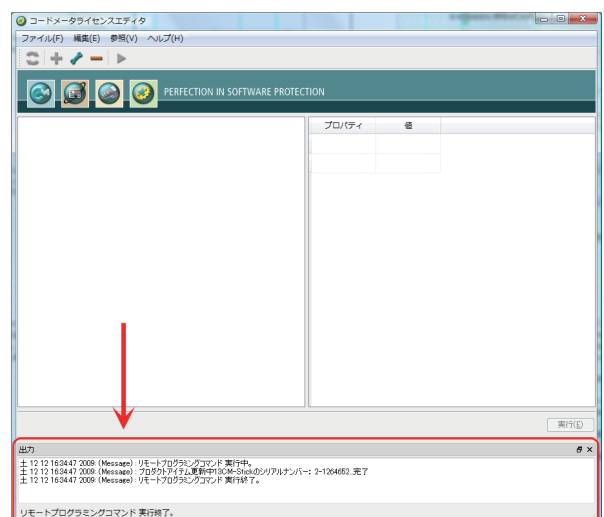
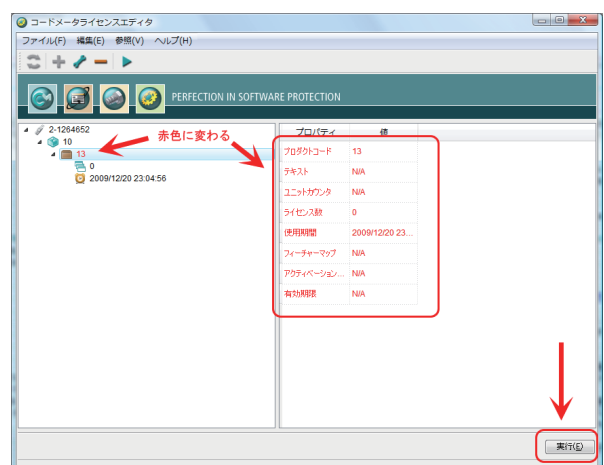
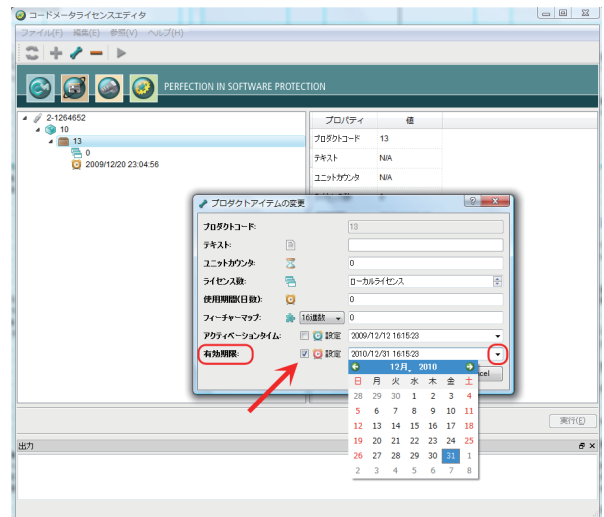
⑤ ライセンス更新ファイルを作成する

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

プロダクトアイテムの変更画面で更新する内容を編集します。ここでは、使用有効期限を2010年12月31日に変更します。

プロダクトアイテムの変更画面でOKボタンをクリックすると、ライセンスエディタ画面に戻ります。今編集しているプロダクトコード(=13)が赤色に変わっています。

「実行」ボタンをクリックすると、ライセンス更新ファイル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。また、コードメータライセンスエディタ画面の下部には、実行内容が表示されます。

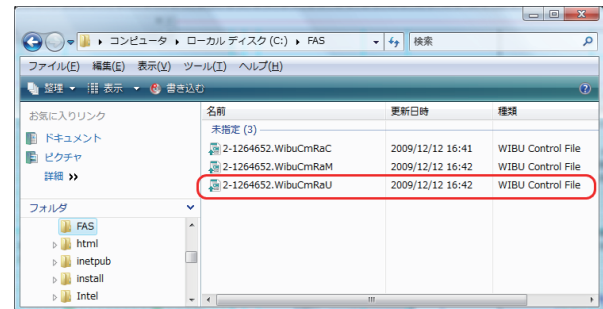


⑥ ライセンス更新ファイルを確認する

エクスプローラ上で、作成されたライセンス更新ファイルを確認します。

ライセンス更新ファイルの拡張子は、".WibuCmRaU"になります。ファイル名の左部は変わりません。

X-XXXXXXX.WibuCmRaU



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。

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

Program Files¥CodeMeter¥DevKit¥bin

〔例〕 コマンドプロンプトを開いて、

>CD ¥Program Files¥CodeMeter¥DevKit¥bin ↓

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。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、CM-Stick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 /PETA10Dec31 /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2010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。

ライセンス貸出用サーバー CM-Stick 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CM-Stick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

指定したリモート要求ファイルからリモート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リモート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

CM-Stick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CM-Stick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)

<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CM-Stick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 CM-Stick 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CM-Stick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

指定したリモート要求ファイルからリモート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リモート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CM-Stick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CM-Stick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. CM-Stick を更新する（ユーザー側）

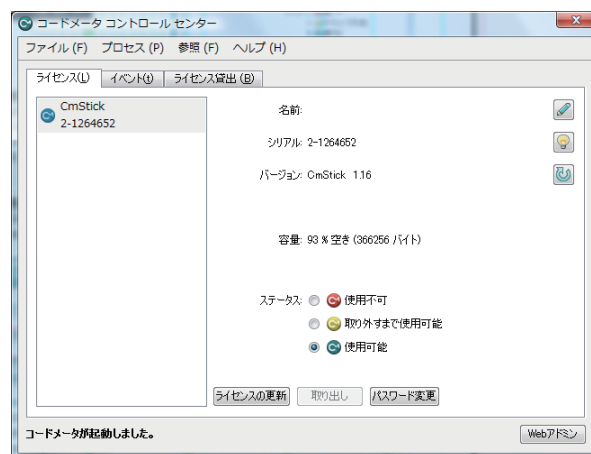
ライセンス更新ファイル"2-1264652.WibuCmRaU"を使って、CM-Stickの内容を更新します。CM-Stickの更新は、コードメータコントロールセンターを使用します。

① CM-Stick を PC に装着する

装着するCM-Stickは、ライセンス要求ファイルを作成したときのCM-Stickである必要があります。別のCM-Stick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。

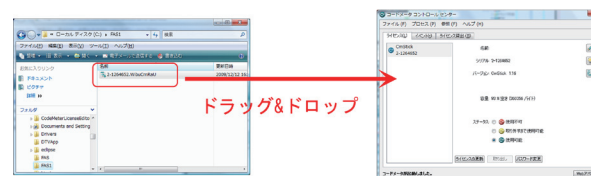
② コードメータコントロールセンターを開く

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。



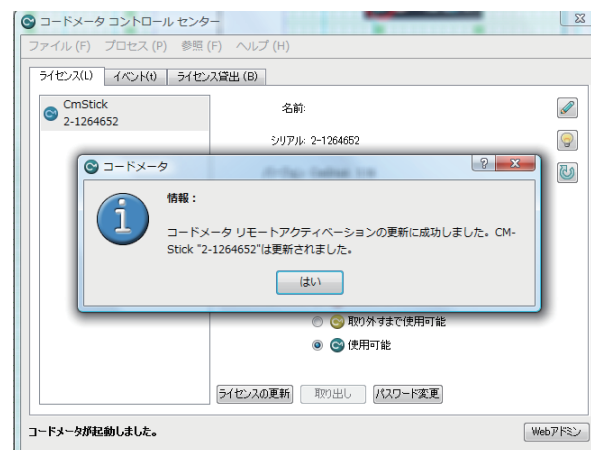
③ ライセンス更新ファイルをドラッグする

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。CM-Stickの更新作業が開始されます。



④ CM-Stick が更新された

右の画面が表示され、CM-Stickが更新されます。WebAdmin上でCM-Stick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CM-Stickに対して2回使用することはできません。また、別のCM-Stickに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。

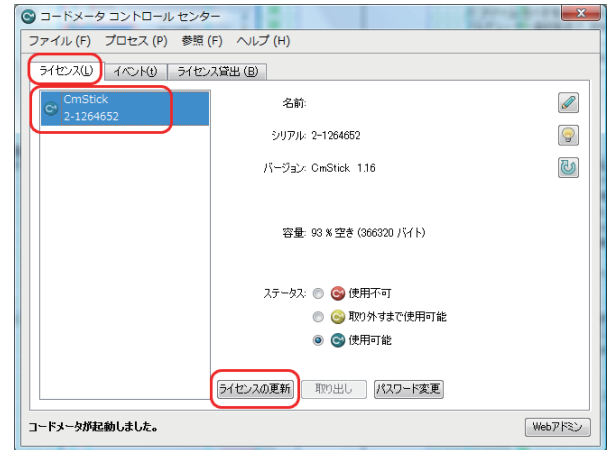
また、ライセンスの更新を、コードメータコントロールセンターの「ライセンスの更新」ボタンをクリックして行うこともできます。

① CM-Stick を PC に装着する

装着するCM-Stickは、ライセンス要求ファイルを作成したときのCM-Stickである必要があります。別のCM-Stick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。

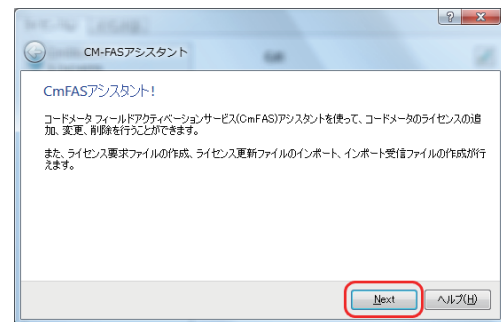
② コードメータコントロールセンターを開く

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

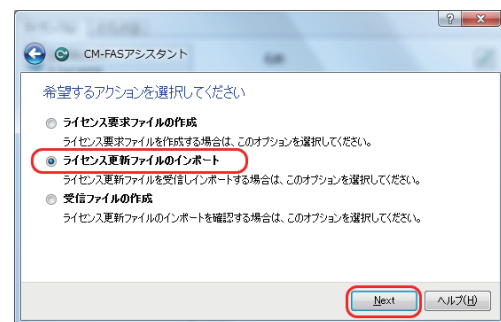


③ 「ライセンスの更新」ボタンをクリックする

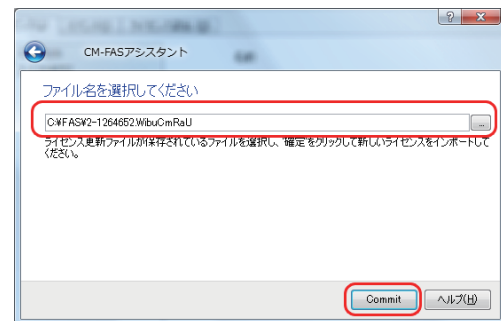
「ライセンスの更新」ボタンをクリックします。CM-FASアシスタントが起動しますので、メッセージに従って進めます。



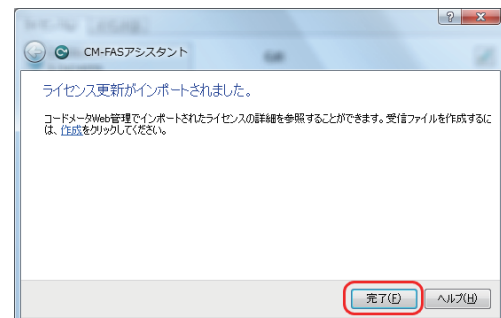
「ライセンス更新ファイルのインポート」を選択し、"Next"ボタンをクリックします。



ライセンス更新ファイルを選択します。
"Commit"ボタンをクリックすると、CM-Stick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、CM-Stickは更新されました。念のため、WebAdminで確認してください。



Chapter 9

コードメータ ライセンスエディタの使い方

- 9-1. コードメータライセンスエディタについて
- 9-2. ストラクチャとナビゲーション

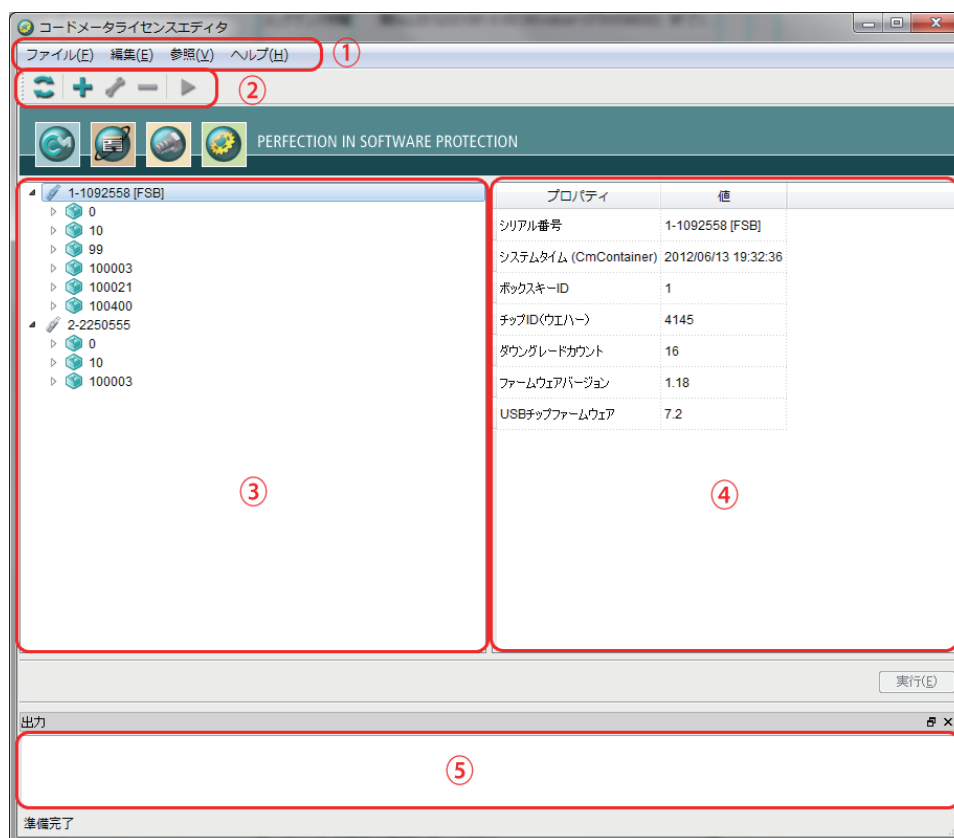
9-1. コードメータライセンスエディタについて

コードメータライセンスエディタ (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) を作成・編集・削除することができます。さらに、ファイルベースのリモートプログラミング (CmFAS: CodeMeter Field Activation Service) をサポートしています。

このコードメータライセンスエディタは、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. ストラクチャとナビゲーション

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出力ウィンドウ

メニューバー

「ファイル」メニュー

リモートプログラミングモード

*.WibuCmRaC または *.WibuCmRaM ファイルを使って、CmDongles内のライセンス情報をコードメータライセンスエディタにロードします。

ダイレクトプログラミングモード

CmDongles内のライセンス情報をコードメータライセンスエディタにロードします。

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>キー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>キーからも実行できます。

アイテムの編集

既存のファームアイテムの編集を行います。<CTRL+M>キーからも実行できます。

アイテムの削除

既存のファームアイテムを削除します。<CTRL+D>キーからも実行できます。

実行

CmDongleへ編集したライセンス内容を保存します。<CTRL+X>キーからも実行できます。

リフレッシュ

CmDongle内のライセンス情報をリフレッシュします。<CTRL+R>キーからも実行できます。

「参照」メニュー

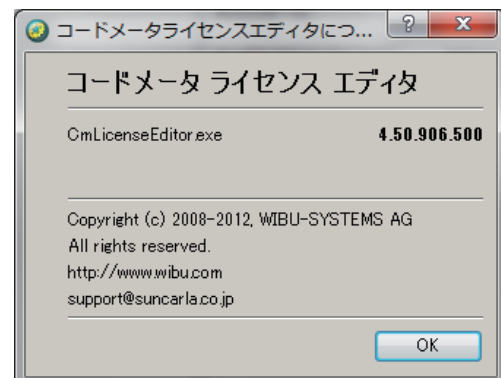
ステータス

出力ウィンドウを再アクティベートします。

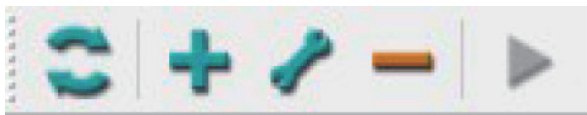
「ヘルプ」メニュー

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

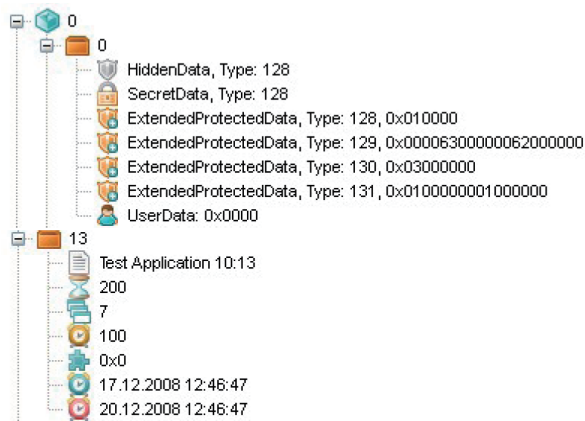


標準機能をシンボルバーから実行することができます。シンボルバーを移動することも可能です。

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	13
テキスト	テスト用
ユニットカウンタ	300
ライセンス数	10
使用期間	100
フィーチャーマップ	N/A
アクティベーション...	2012/06/13 17:07:40
有効期限	2015/06/13 17:07:40
メンテナンス期間	開始: 2012/01/01 0:00:00(value=378658800) 終了: ...

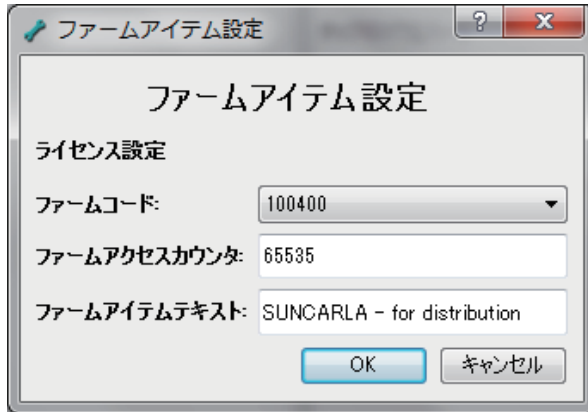
出力ウィンドウ

実行結果やエラーメッセージが表示されます。

リモートプログラミングモード

「ファイル」/「リモートプログラミングモード」から、ライセンス要求ファイル"*.WibuCmRaC" または"*.WibuCmRaM"を選択します。

ファームアイテムの作成と編集



ファームコード

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することも可能です。

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、この値は 1 減ります。

[NOTE]

デフォルト値は65535(0xFFFF)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ベートになっていますが、この値をプログラムすることができます。この値が0になると、ファームアイテムがロックされ使用できなくなります。

ファームアイテムテキスト

ファームアイテムの詳細をテキスト入力します。

プロダクトアイテムの作成と編集

プロダクトアイテム修正

プロダクトコード:	13
テキスト:	テスト用
ユニットカウンタ:	300
ライセンス数:	10
使用期間(日数):	360
フィーチャーマップ:	16進数
アクティベーションタイム:	<input checked="" type="checkbox"/> 設定 2012/06/13 17:07:40
有効期限:	<input checked="" type="checkbox"/> 設定 2015/06/13 17:07:40
メンテナンス期間	<input checked="" type="checkbox"/> 設定 開始 2012/01/01 0:00:00 終了 2015/12/31 23:59:59

OK キャンセル

テキスト

プロダクトコードに関する詳細をテキスト表示します。

ユニットカウンタ

ユニットカウンタを設定します。

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。

デフォルト設定は、「ローカルライセンス」です。

ライセンス割り当ての仕様についてはAxProtectorで設定します。

使用期間(日数)

ライセンスが有効な日数を指定します。

フィーチャーマップ

フィーチャーマップを指定します。

フィーチャーマップは、モジュール管理、ファンクション(機能)管理、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。

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)を設定する場合は、「設定」ボタンにチェックを入れます。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。

[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。

有効期限(Expiration Time)

ライセンスの有効期限を設定する場合は、「設定」ボタンにチェックを入れます。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。

メンテナンス期間 (Maintenance Period)

メンテナンス期間を設定する場合は、「設定」ボタンにチェックを入れます。

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タイムと値を入力する画面が表示されます。

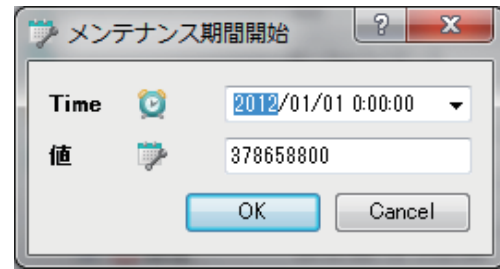
値は、2000年1月1日から数えた秒数になります。現在のところ、2136年2月7日まで設定できます。

タイムは直接入力するか、カレンダーコントロールを使って入力します。

「OK」ボタンをクリックして保存します。

[NOTE]

この「メンテナンス期間」を利用するには、CmDongleのファームウェア1.18以上が必要になります。



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、CM-Stickなどのコードメータキー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も可能です。

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。

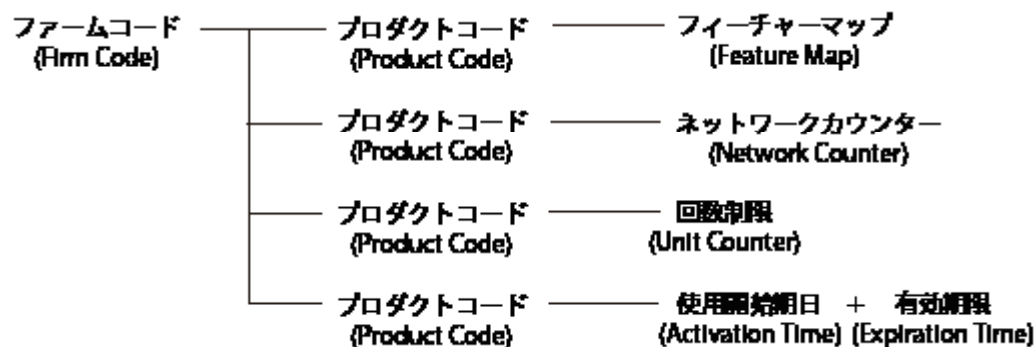
- ① コードメータ開発キットがインストールされていること。
- ② 貴社のライセンスファイルCmFirm.wbcがインストールされていること。
- ③ 貴社のコードメータFSB (CM-FSB)がPCに装着されていること。

作業は、Windows XP/Vista/7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はファームコード (Firm Code)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、フィーチャーマップ (Feature Map)や、ネットワークカウンタ、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。

コードメータキーの登録作業は、まずファームコードを登録することから始まります。そして、ファームコードの中に、必要なプロダクトコードやセキュリティオプション項目を登録していきます。ただし、商品版の場合、弊社から出荷する時点ですでに貴社のファームコードが登録されていますので、実際の作業はプロダクトコードやセキュリティオプション項目の登録作業から始まることになります。



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CM-Stickに登録してみます。

① CM-Stickとコードメータ FSB(CM-FSB) をそれぞれ任意の USB ポートに挿入します。

CM-Stickにファームコードを登録するには、必ずコードメータFSB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 CD(Change Directory) コマンドで以下のフォルダをカレントにします。

¥Program Files¥CodeMeter¥DevKit¥bin

[例] コマンドプロンプトを開いて、

>CD ¥Program Files¥CodeMeter¥DevKit¥bin ↓

④ 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)

```

C:\Program Files\CodeMeter\DevKit\bin>cmboxpgm /F10 /CA
Started at 2008-01-23 16:23:20
CmBoxPgm has Version 3.20.0.500
*** Skip Firm Security Box 1-1099190
*** Add Firm Item, CmStick 1-1020965, FC=10
CmBoxPgm finished at 2008-01-23 16:23:22
  
```

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。

*ファームコードを1つ新規追加すると、コードメータFSBのファームアイテム99のユニットカウンタが1つ減ります。

*指定したファームコードがすでにCM-Stickに存在すると、エラー22が表示されます。

*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 CM-Stick とコードメータ FSB(CM-FSB) をそれぞれ任意の USB ポートに挿入します。

CM-Stick 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 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 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¥CodeMeter¥DevKit¥bin

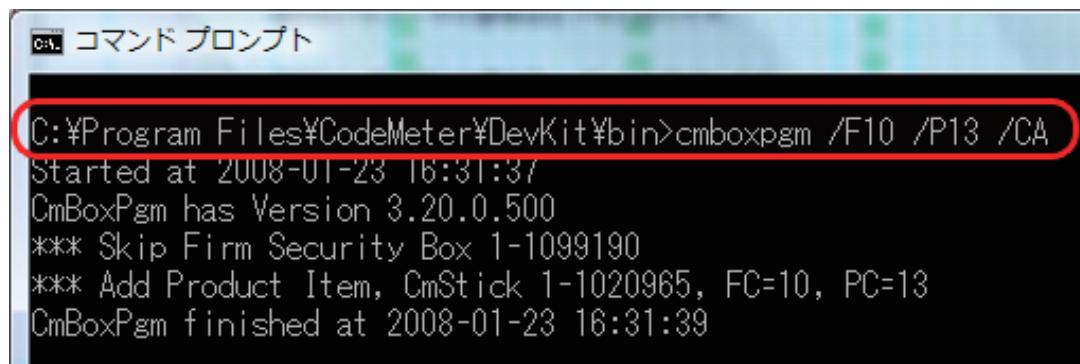
[例] コマンドプロンプトを開いて、

>CD ¥Program Files¥CodeMeter¥DevKit¥bin ↓

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) = 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)



```
C:\Program Files\CodeMeter\DevKit\bin>cmboxpgm /F10 /P13 /CA
Started at 2008-01-23 16:31:37
CmBoxPgm has Version 3.20.0.500
*** Skip Firm Security Box 1-1099190
*** Add Product Item, CmStick 1-1020965, FC=10, PC=13
CmBoxPgm finished at 2008-01-23 16:31:39
```

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に続けて登録したい任意のプロダクトコード (Product Code) を指定します。

/CA ... Product Code、Firm Codeを新規に追加 (作成) する命令です。

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) が削除されることになります。

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することができますが、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弊社にて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CM-BOX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) = 2010年12月31日を新規登録する場合：

CmBoxPgm /F10 /P13 /PETA10Dec31 /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。2011年4月1日は、11Apr01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。

CmBoxPgm /F10 /P13 /PETA10Dec31 /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=2010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 /PETA12Aug31 /CU ↓ (↓はEnterキーです)

2010年12月31日が2012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。

[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、CM-BOX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CM-Stick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CM-BOX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CM-BOX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CM-BOX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CM-BOXで選択する。マスクmはオプション。

* CM-BOX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CM-BOXを選択します。

/L

CM-BOXの内容をリスト表示する。(L=List)

CmBoxPgm /L

最初に見つけたCM-BOX(FSB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CM-BOX (最初に見つけたCM-BOX)の内容をリスト表示する。

CmBoxPgm /QB2 /L

インデックス 2 のCM-BOX (2 番目に見つけたCM-BOX)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CM-BOX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CM-BOX (2 番目から 4 番目に見つけたCM-BOX, 計 3 個)の内容をリスト表示する。FSB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。但し、FSBは除く。

CmBoxPgm /QN2,4 /R

インデックス範囲2から4のCM-BOX (2 番目から 4 番目に見つけたCM-BOX, 計 3 個)の内容を削除する。ただし、FSB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 (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CM-BOX (1番目から 4 番目に見つけたCM-BOX, 計4個、ただし、FSB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CM-BOX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 /FPTA2010Nov04, 13:14:15PST /CU /L

シリアル番号1-1234を持つCM-BOX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2010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CM-BOX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:/PATA11Dec31,00:00:00)

/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:/PETA11Dec31,00:00:00)

/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CM-BOX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CM-BOX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 /F206 /P2001 /PET /CD

シリアル番号1-1234のCM-BOX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」 タブ
- 11-5. 「イベント」 タブ
- 11-6. 「ライセンス貸出」 タブ
- 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。

CodeMeter.exeが起動する。

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]

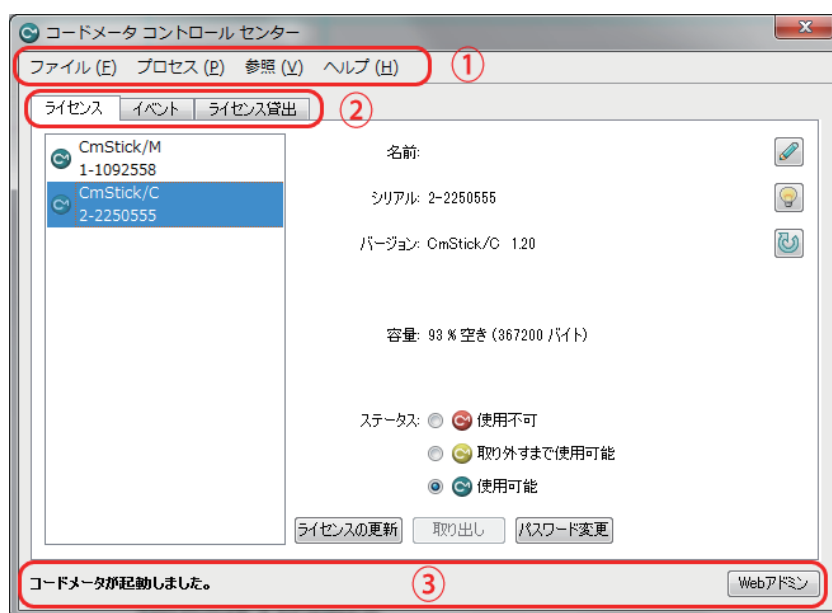
CodeMeterLinが起動する。

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  または  をダブルクリックする。
2. Windowsタスクバーにあるコードメータアイコン  または  上で、マウスを右クリックすると表示されるメニューから起動する。

「ウィンドウ表示」: コードメータコントロールセンターをウィンドウ表示する。

「Webアドミン(W)」: Webアドミンを起動する。

「全てのCmDongleを取り出す」: 接続されているCmDongleを安全に取り出す。

「ヘルプ」: コードメータオンラインヘルプを開く。






「コードメータコントロールセンターについて」: バージョン情報を表示する。

「終了(Q)」: コードメータコントロールセンターを終了する。

3. [スタート]-[すべてのプログラム]-[CodeMeter]-[CodeMeter Control Center]から起動する。

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

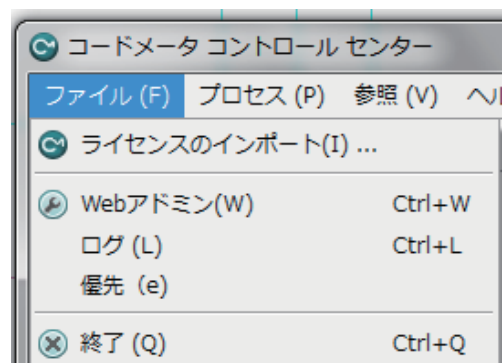
「ファイル (F)」メニュー

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
ター画面に直接ドラッグ&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(L)」を有効にすると、Webアドミン上の「診断 | ログファイル」に反映されます。

ログファイルは、%Program Files\CodeMeter\Logsに保存されます。(Windowsの場合)

優先(e)

コードメータWebアドミンを起動し、ネットワーク設定画面を表示します。

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

ただし、コードメータライセンスサーバーは終了しません。

プロセス (P) メニュー

全てのCmDongleを取り出す

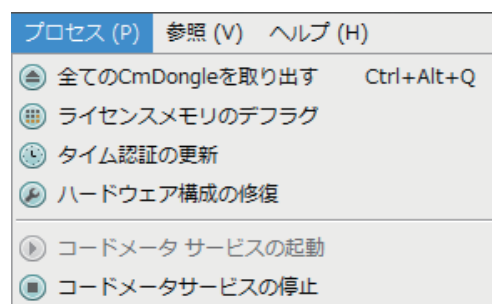
一度に全てのCmDongleを取り出します。
<CTRL>+ALT+Q

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ーのデフ
ラグを行います。<STRG>+ALT+D

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。
すべてのタイムスタンプがリフレッシュされます。



ハードウェア構成の修復

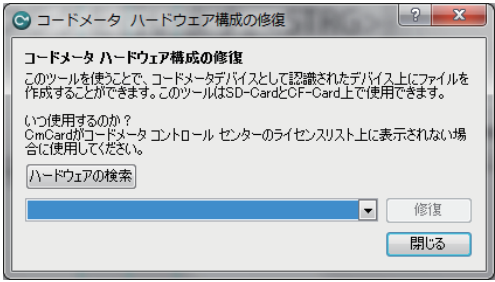
SDカードやCFカードなどのCmDongleのハードウェア構成を修復します。CmCard(SD, CFなど) がコードメータコントロールセンターにリスト表示されない場合に使用します。

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。



「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

ライセンス貸出

「ライセンス貸出」タブの表示・非表示の選択

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 checked="" type="checkbox"/> 接続されている全てのCmContainerを表示	Alt+S
<input checked="" type="checkbox"/> オープンされている全てのハンドルのリスト	
<input checked="" type="checkbox"/> 利用可能なライセンス エントリをすべて表示	Alt+E
<input checked="" type="checkbox"/> ライセンス貸出	

「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>

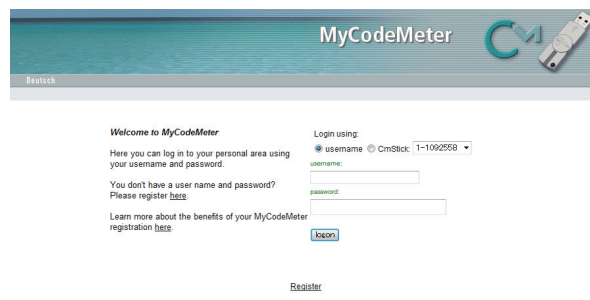
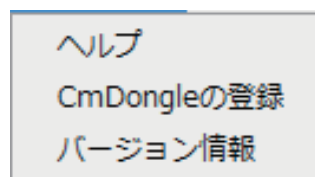
にアクセスします。

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

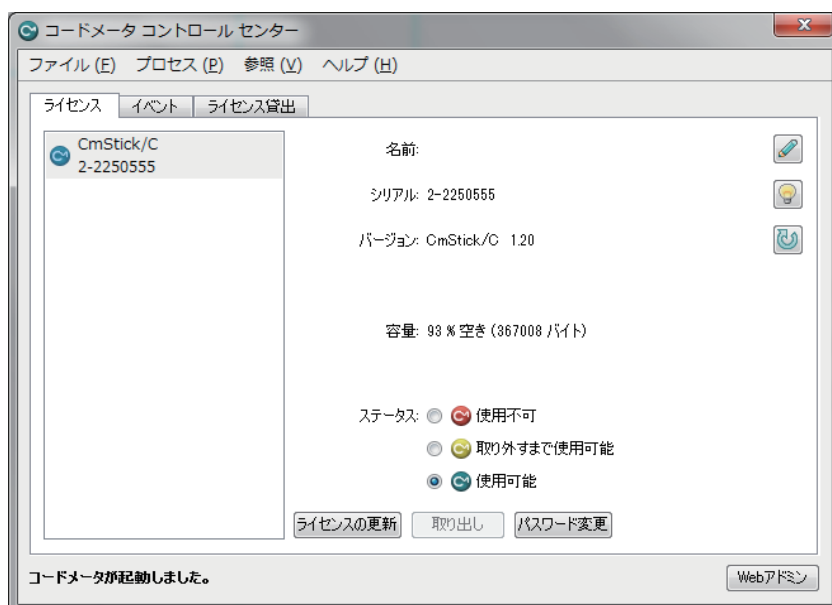
バージョン情報

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」 タブ

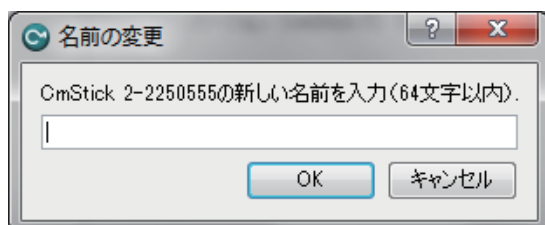
「ライセンス」タブでは、接続されているCmContainerの情報の表示やファームウェアのバージョンアップなどの変更が行えます。また、「ライセンスの更新」からライセンスの更新やパスワードの変更などが行えます。





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。

* 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のファームウェアアップデートサーバーに自動的にアクセスします。アップデートの際、CmDongleのパスワードが要求されます。

* 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。途中で取り外すと、CodeMeter SmartCardチップが破損する場合があります。

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、第三者による不正アクセス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は使用可能の状態になります。

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。

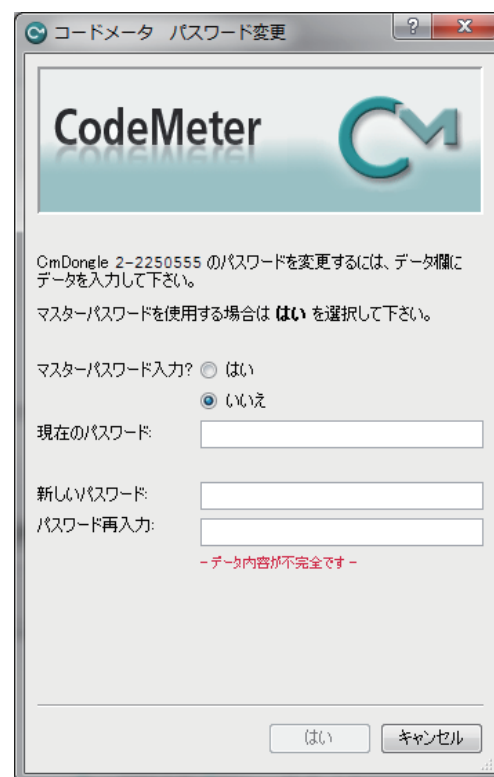
パスワード変更

選択したCmDongleのパスワードを変更します。

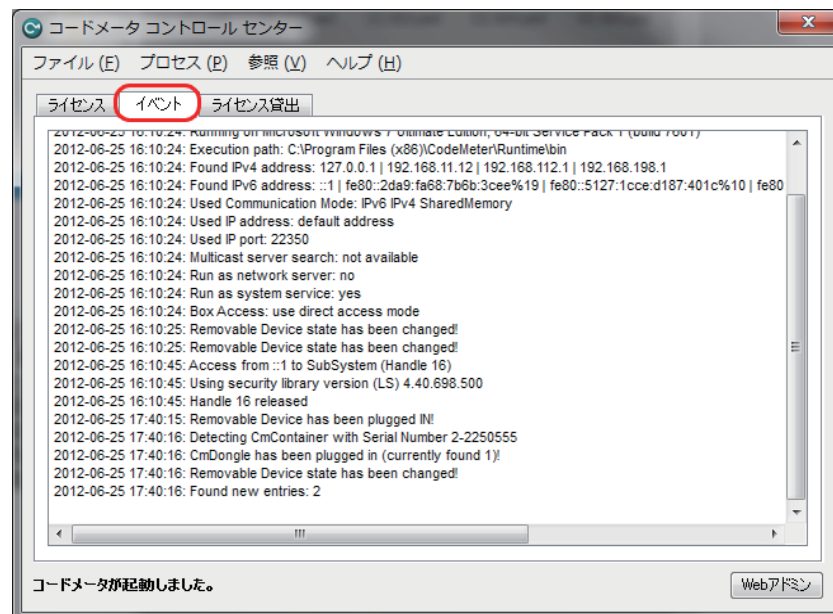
「コードメータ パスワード変更」画面で、次のプロセスを行います。

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。

CmDongleのパスワードを忘れた場合は、CmDongleマスターパスワードを使って新しいパスワードを作成することができます。



11-5. 「イベント」 タブ



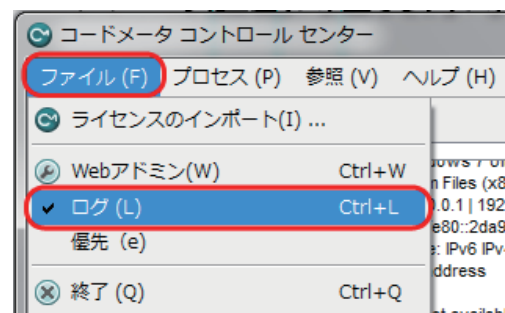
「イベント」タブでは、コードメータライセンスサーバーの起動時と実行時の情報ログや、次の内容を表示します。

- 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

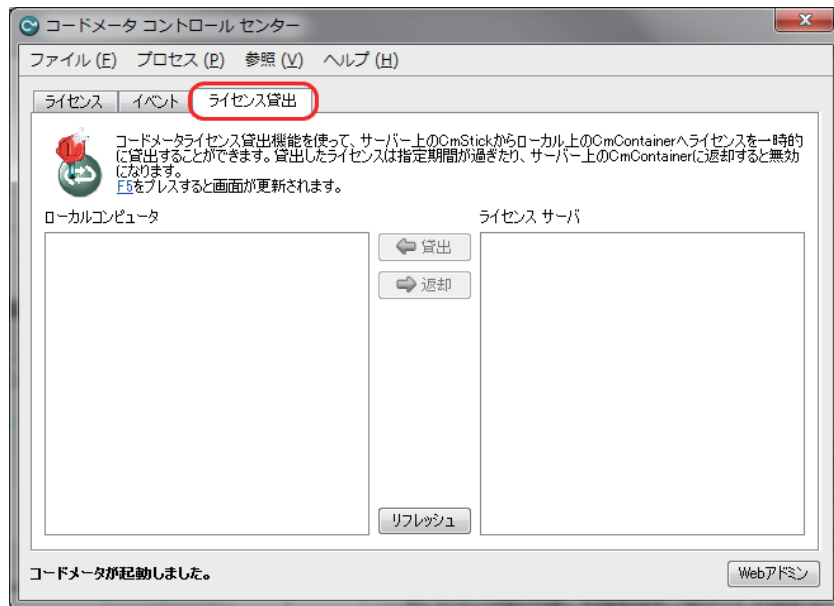
「参照(V)」メニューで、表示する内容を選択できます。

また、ログをとるには、「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¥CodeMeter¥Logsフォルダに作成されます。(拡張子"log")



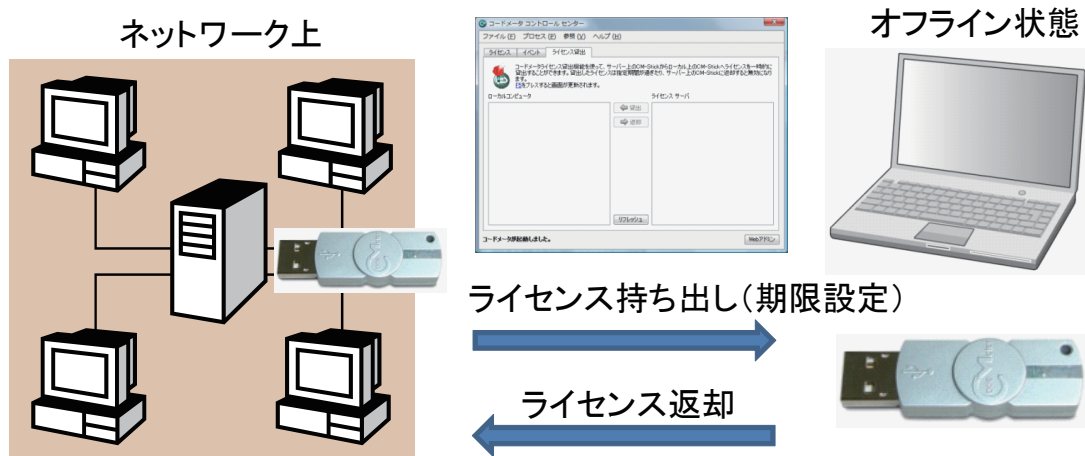
11-6. 「ライセンス貸出」タブ



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は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増え、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るため、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CM-Stickをライセンス貸出用に設定する
2. クライアント用CM-Stick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 CM-Stick 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CM-Stick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CM-Stick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CM-Stick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 CM-Stickとコードメータ FSB(CM-FSB) を USB ポートに挿入します。

サーバー用CM-Stickにライセンス貸出設定を行うには、コードメータFSB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CM-Stick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、

CM-Stick 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CM-Stick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 top screenshot shows the 'CodeMeter WebAdmin' interface. The 'License Borrowing Server' section is highlighted with a red box. It contains a table with the following data:

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 bottom screenshot shows the 'CodeMeter WebAdmin' interface. The 'Product Item Details' section is highlighted with a red box. It contains a table with the following data:

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

CM-Stick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

CM-Stick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CM-Stick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)

<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。

[, 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。

2. クライアント用 CM-Stick をライセンス貸出用に設定する

①クライアント用 CM-Stickとコードメータ FSB(CM-FSB) を USB ポートに挿入します。

クライアント用CM-Stickにライセンス貸出設定を行うには、コードメータFSB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CM-Stick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CM-Stick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、

CM-Stick シリアル番号：2-1264652のCM-Stick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CM-Stick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CM-Stick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。

[各パラメータの説明]

サーバー用CM-Stick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CM-Stickを作成する。

(Syntax)

/blc:[cm][ca],<fc>,<pc> [, serverID]

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ム	ネットワークカウンタ
300	Borrow License Client	n/a	n/a	[非貸出]	1
100003 Bundling Articles					
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ム	ネットワークカウンタ
1	SecurKey Lite	n/a	n/a	n/a	1
100400 SUNCARLA - for distribution					
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ム	ネットワークカウンタ
13	-	n/a	n/a	n/a	ローカル
14	-	n/a	n/a	n/a	ローカル

[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CM-Stick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

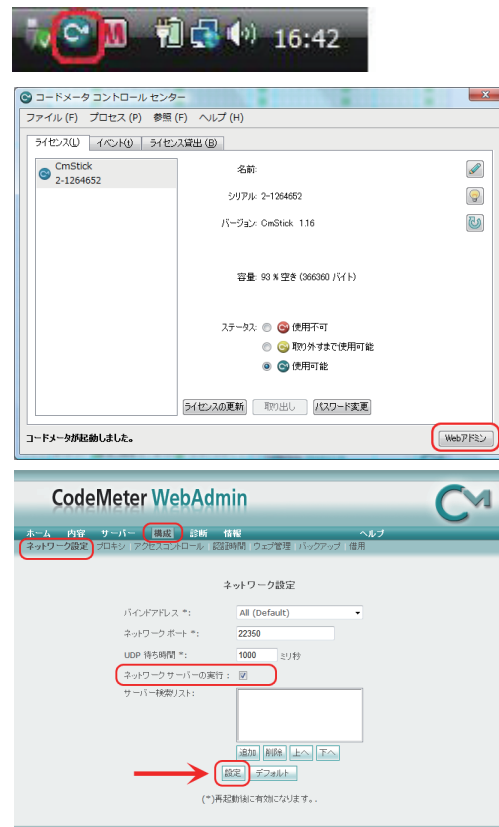
①サーバー用 CM-Stick をコードメータサーバーに装着する

ライセンス貸出設定を行ったサーバー用CM-Stick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CM-Stick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、「ネットワーク設定」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。

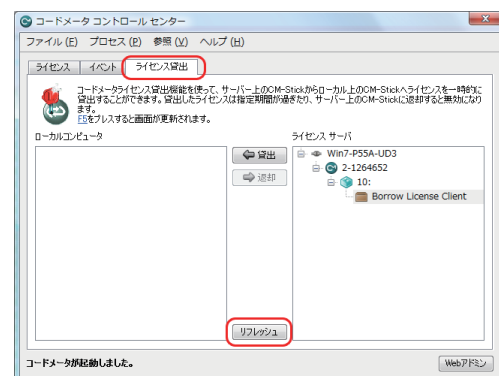


③ クライアント用 CM-Stick を装着する

ライセンス貸出設定が済んでいるクライアント用CM-Stick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]

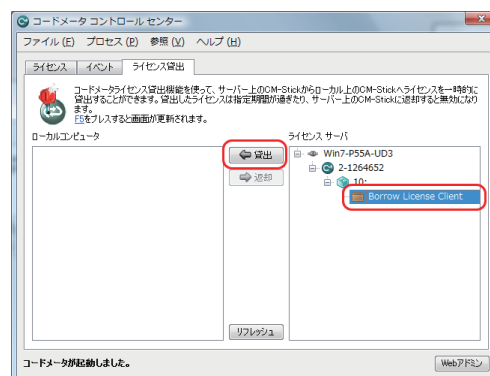
「リフレッシュ」ボタンの代わりに"F5"をプレスすることも可能です。また、ネットワーク状況により、"ライセンスサーバー"ウィンドウにライセンスが表示されるまで時間がかかる場合があります。

[NOTE]

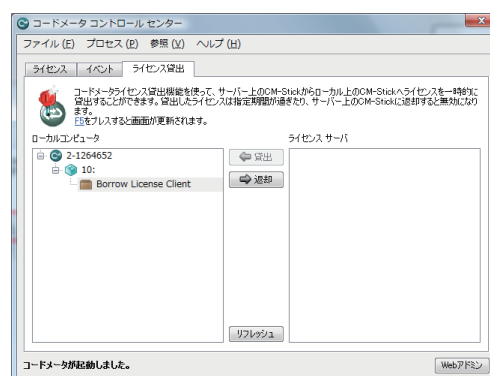
ライセンス貸出を行うには、ローカルPCはネットワークに接続している必要があります。

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。



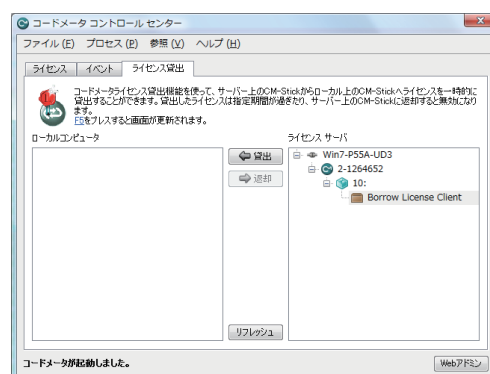
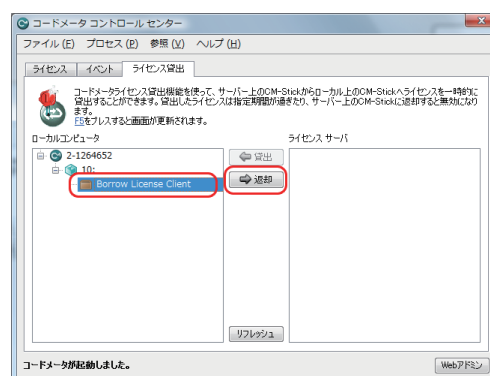
⑥ ライセンスを「返却」する場合

クライアント用CM-Stick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CM-Stick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。



11-8. ライセンス貸出の有効期限について

ライセンス貸出の際、貸出されたライセンスの有効期限は、コードメータサーバーのWebアドミンの[構成]/[借用]ページの最大借用期間の数字が反映されます。分単位で入力し、「設定」ボタンをクリックすると貸出ライセンスの有効期限が反映されます。



"最大借用期間"に何も設定されていない場合は、ライセンス貸出用CM-Stick(サーバー用)を作成する際に設定した"/bls"のパラメータ<duration>が反映されます。

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

[NOTE]

Webアドミン上の"最大借用期間"とCM-Stick(サーバー用)の"/bls"の<duration>を両方設定した場合、"/bls"の<duration>の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。

"/bls"のパラメータについては、前章「11-7. ライセンス貸出・返却の方法」をご参照ください。

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

12-1. 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. Host, 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、「Webアドミン(W)」を選択する。
- ・コードメータコントロールセンターから「Webアドミン」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、「Webアドミン(W)」を選択する。
- ・コードメータコントロールセンターから「Webアドミン」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。
- 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. ホーム

基本的な情報

「ホーム」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WebAdminの全体のステータス情報を表示します。



ホスト名:

「ホスト名」ボタンには、コードメータライセンスサーバーが起動している実際のPC名が表示されます。ポート22350を使ってネットワークにサーチリクエストが送られます。

ホスト名を変更するには次のようにします。

1. ホストのDNS名ボタンをクリックします。
「CodeMeter Web管理」ダイアログが表示されます。

2. 「サーバーを選択して下さい。」ドロップダウンでコードメータとコードメータライセンスサーバーが起動している別のPCを選択します。

3. ネットワークアドレスを使用する場合は、「IP Addressを使用」にチェックを入れます。

4. OKなら「選択」ボタンをクリックします。

IPアドレス:

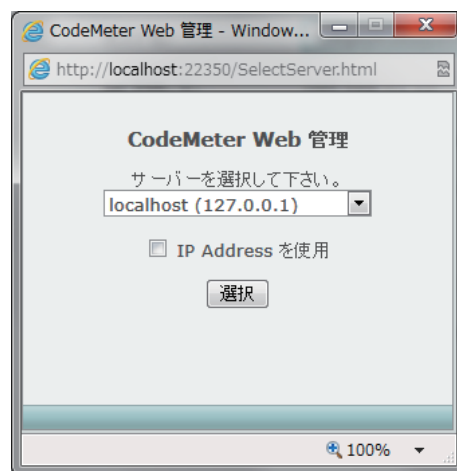
使用中のIPアドレスが表示されます。

OS:

使用中のOSが表示されます。

サーバー起動:

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。



Runtime Version:

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。

サーバーバージョン:

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。

Web管理バージョン:

使用中のWebAdminのバージョンを表示します。

12-3. 内容

CMスティック

「内容」-「CMスティック」ページでは、選択したCmContainerの情報を表示します。

The screenshot shows the 'CodeMeter WebAdmin' interface. The top navigation bar includes 'ホーム', '内容', 'サーバー', '構成', '診断', '情報', and 'ヘルプ'. Below this, a sub-navigation bar lists 'CMスティック', 'ライセンス', 'ユーザーデータ', and 'バックアップ/リストア'. The main content area displays the configuration for a selected CM Stick. It includes a dropdown menu for 'CMスティック:' with the value '2-2250555', a text field for '名前:' with the value '<no name>', and a section for 'CmContainer のタイプ:' showing 'CmStick/C 1.20'. Below this, the '最初のデバイス:' is listed as 'D: (No Flash)'. The 'ステータス:' section shows three radio buttons: '使用不可' (selected), 'CMスティックを抜くまで使用可能', and '使用可能'. At the bottom, system information is displayed: 'システムタイム (PC): 2012-06-27 19:31:42', 'システムタイム (CmContainer): 2012-06-27 19:31:40', '認証タイム (CmContainer): 2012-06-27 19:31:36', and '空きメモリー: 93 % (366.944 Bytes)'. There are buttons for '設定' and 'クリーン アップ'.

CMスティック:

表示されているCmContainerを選択します。複数表示されている場合は、適切なCmContainerを選択します。

名前:

選択したCmContainerの名前を表示します。

名前を変更する場合は、コードメータコントロールセンターを利用します。

CmContainer のタイプ:

選択したCmContainerのタイプを表示します。Shows the Type of the selected CmContainer.

最初のデバイスFirst Device:

選択したCmDongleのドライブ情報を表示します。

ドライブサイズは、フラッシュメモリー付のCmDongleの場合のみ表示されます。

ステータス:

選択したCmContainerの現在のステータスを表示します。

- 使用不可:

接続しているCmContainerは無効でアプリケーションは利用できません。

- CMスティックを抜くまで使用可能:

CmDongleは接続している間は有効ですが、PCから取り外すと自動的に無効になります。

- 使用可能:

CmContainerは完全に有効です。PCから取り外しても有効です。

CmContainerのステータスは、コードメータコントロールセンターで変更することができます。

[NOTE]

CmDongleが紛失したり、盗まれた場合のセキュリティを考慮すると、「CMスティックを抜くまで使用可能」ステータスでの使用がお奨めです。再度CmDongleを使用する場合に、パスワード要求されるため、第三者の不正使用を防止することができます。

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。「設定」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タイムサーバーにアクセスし、上記3つのタイムを更新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。

クリーンアップ

CmDongleチップメモリーのデフラグを行います。

バックアップ/リストア

「内容」/「バックアップ/リストア」ページで、CmDongle内のパーソナルデータの保存およびCmDongleへのリストア(復元)ができます。

このバックアップ/リストアメカニズムは、CmDongle内のファームコード"0"に関連するユーザーデータが対象で、ファームコード/プロダクトコードなどのライセンス情報はバックアップ/リストアすることができません。ファームコード/プロダクトコードなどのライセンス情報をリストアする場合は、Wibuサポートまでご連絡ください。

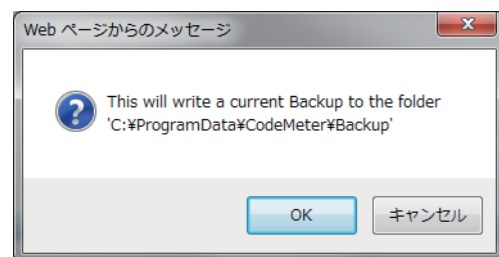


CMスティック

バックアップまたはリストアを行うCmDongleをドロップダウンリストから選択します。

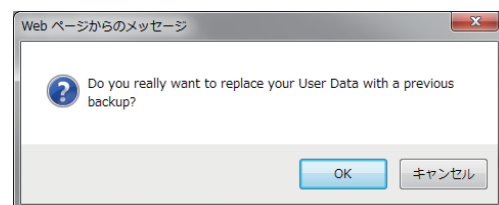
バックアップ開始

- 1.「バックアップ開始」ボタンをクリックします。最終バックアップ時刻が表示されます。
- 2.右のダイアログが表示されますので、「OK」ボタンをクリックします。



リストア

- 1.「ファイル選択」をクリックし、バックアップファイルを選択します。
- 2.「リストア」ボタンをクリックします。
3. 右のダイアログで「OK」ボタンをクリックします。



[NOTE]

リストアを行うと、バックファイル作成後に変更した内容はすべて失われます。

12-4. 構成

ネットワークサーバステータス、プロキシ、アクセスコントロール、リモートアクセス、タイムサーバ、バックアップの設定を行います。

ネットワーク設定

ネットワーク環境を設定する場合、「構成」/「ネットワーク設定」を利用します。

バインドアドレス:

コードメータライセンスサーバーがバインドされたネットワークアドレスを選択します。

Select the Network Address to which the service CodeMeter License Server is to be bound.

--- Primarily, this is required when the PC has several network cards (virtual adapter) and is to act as a network server providing its licenses on the network.

ネットワークポート:

コードメータ通信を行うネットワークポートを指定します。デフォルトは22350です。

この22350ポートは、IANA (Internet Assigned Numbers Authority)でコードメータ通信として登録されています。

[NOTE]

このポート値を変更することは可能ですが、コードメータでプロテクトされたアプリケーションが起動するときに、すべてのコードメータライセンスサーバーはこのポートを使用します。

UDP待ち時間:(UDP=User Datagram Protocol)

UDPリクエストする時間を設定します。デフォルトでは、1000ミリ秒 (1 秒) が設定されています。

[NOTE]

UDP待ち時間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。

ネットワークサーバーの実行:

コードメータネットワークサーバーとして実行する場合にチェックをいれます。

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。

「追加」、「削除」ボタンでサーバー検索リストを編集します。「上へ」、「下へ」ボタンで順番を変更します。変更後、「設定」ボタンをクリックして内容を保存します。

「デフォルト」ボタンで設定を初期状態に戻すことができます。初期状態に戻すと、ネットワークサーバーも設定から外れます。「デフォルト」状態を保存するには、「設定」ボタンをクリックしてください。

ネットワーク設定を行った場合、コードメータサービスの再起動が必要になる場合があります。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。

サーバー接続に成功しているかどうかを調べるには、「ホーム」ページで、「ホスト名」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。また、コードメータコントロールセンターの「イベント」タブからコミュニケーション状況を調べることも確認できます。

[NOTE]

サーバー接続が確立されていない場合、クライアントPC上でサーバーのIPアドレスを指定してください。

プロキシ設定

「構成」/「プロキシ」で、プロキシサーバ使用時の設定を行います。

CodeMeter WebAdmin

ホーム 内容 サーバー **構成** 診断 情報 ヘルプ

ネットワーク設定 | プロキシ | アクセスコントロール | 認証時間 | ウェブ管理 | バックアップ | 借用

プロキシ設定

プロキシサポート: ☐ 使用可能

- プロキシサーバー:

- プロキシポート:

認証 ☐ 使用可能

- プロキシユーザー:

- プロキシパスワード:

プロキシサポート:

プロキシサーバーを使用するためのオプションです。

「使用可能」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。

[NOTE]

認証時間の更新やオンラインショップによる販売の場合は、プロキシサーバーが必要になります。

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。

アクセスコントロール

「構成」/「アクセスコントロール」ページで、コードメータライセンスサーバーにアクセスするクライアントを管理することができます。

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PCをリスト表示します。ここで表示されないクライアントPCは、コードメータライセンスサーバーを利用できません。

[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はコードメータライセンスサーバーを利用できます。

FSBアクセス:

ネットワーク上でFSB (Firm Security Box)を複数ユーザーでシェアして使用したい場合に「使用可」ボタンにチェックをいれます。チェックをいれると、CmContainerへのプログラムや、アプリケーションのプロテクトが可能になります。

「設定」ボタンをクリックして変更を保存します。「デフォルト」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エンプティになり、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る場合があります。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。

Web管理 (リモートアクセス)

「構成」/「ウェブ管理」ページで、コードメータWebアドミンへのリモートアクセス設定およびインターフェース言語の変更ができます。

ローカルアクセスのみ許可 (無制限)

Webアドミンへのローカルアクセスを無制限に許可します。

認証が必要です:

Webアドミンへの書き込み変更をする場合に認証 (ユーザー名とパスワード) を要求します。このWeb管理ページで登録したユーザー名とパスワードが反映されます。また、認証はシングルサインベースのため、一度認証が確立されると、Webアドミンを終了するまで認証が有効になります。(Webアドミンを終了するまでユーザー名とパスワードが要求されない)

リモート読み込み:

Webアドミンへのリモートリードアクセスを可能にします。(読み込みのみ可能)

Language:

コードメータWebアドミンで使用するユーザーインターフェース言語を選択します。

ドイツ語、英語、フランス語、イタリア語、日本語、中国語が可能です。「設定」ボタンをクリック後、コードメータライセンスサーバーを再起動すると適用されます。

認証時間 (Time Server)

「構成」/「認証時間」でコードメータタイムサーバーの管理を行います。



タイムサーバー:

認証時間を更新するWibu-Systemsコードメータタイムサーバーをリスト表示します。タイムサーバーはインターネットアドレスまたはIPアドレスで指定します。「追加」「削除」「上へ」「下へ」ボタンでタイムサーバーリストを編集できます。

タイムアウト:

コードメータタイムサーバーのタイムアウト時間を設定します。

設定後、「設定」ボタンをクリックします。

「デフォルト」ボタンをクリックするとデフォルト設定に戻ります。

バックアップ

「構成」|「バックアップ」ページで、CmDongleのデータバックアップを行う場所とインターバル時間を設定します。



The screenshot shows the 'CodeMeter WebAdmin' interface. The title bar includes 'ホーム', '内容', 'サーバー', '構成' (selected), '診断', '情報', and 'ヘルプ'. Below the title bar is a navigation menu with links: 'ネットワーク設定', 'プロキシ', 'アクセスコントロール', '認証時間', 'ウェブ管理', 'バックアップ', and '借用'. The main content area is titled 'バックアップ' and contains the following settings:

- バックアップパス名:
- バックアップ間隔: hours
- Certified Time: ☐ update before each backup

At the bottom of the settings are two buttons: '設定' (Set) and 'デフォルト' (Default).

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。

Certified Time:

バックアップを行う前に認証タイム更新を行う場合はチェックをいれます。

「設定」ボタンをクリックして変更を保存します。

「デフォルト」ボタンをクリックするとデフォルトの状態に戻ります。

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 をインストールする
- 13-5. sample.pdf を暗号化する
- 13-6. CM-Stick にコードを登録する
- 13-7. 動作を確認する
- 13-8. 複数の PDF ファイルを一括して暗号化する
- 13-9. エラーメッセージをカスタマイズする
- 13-10. プレインテキストパスワードについて
- 13-11. 複数の PDF ファイルを一括して暗号化する
- 13-12. コマンドラインで暗号化する方法
- 13-13. 暗号化された PDF ファイルをユーザーに配布する場合

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ファイルをプロテクトする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CD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CD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。

ステップ1

コードメータ開発ツールをインストールする
(既にインストール済みの場合は不要)

ステップ2

SmashPDFAuthor.exe をインストールする

ステップ3

sample.pdf を暗号化する

ステップ4

CM-Stick にコードを登録する

ステップ5

動作確認

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 6/7/8/9/X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 (CM-FSB)

作業はWindows XP/Vista/7 (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|PDFAuthor Ver6.00 をインストールする

SmartShelter|PDFAuthor Ver6.00 をインストールします。SmartShelter|PDF Author Ver6.00 は、コードメータCD Ver4.30c 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

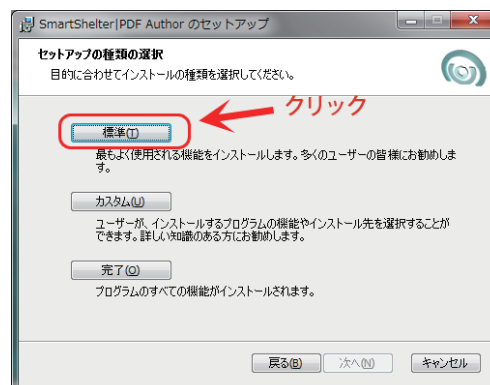
コードメータCDをCD/DVDドライブに挿入し、SmartShelterPDFフォルダの中の"SmashPdfAuthor.exe"をダブルクリックして起動します。

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。



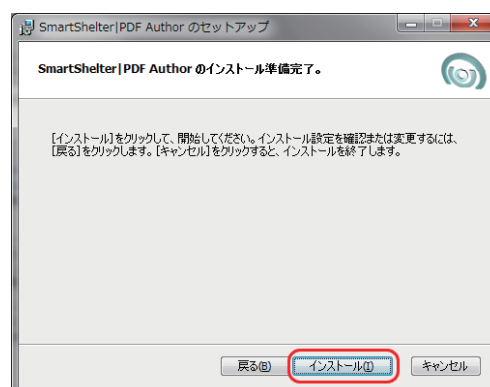
② 「標準」 ボタンをクリックする

セットアップの種類の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。



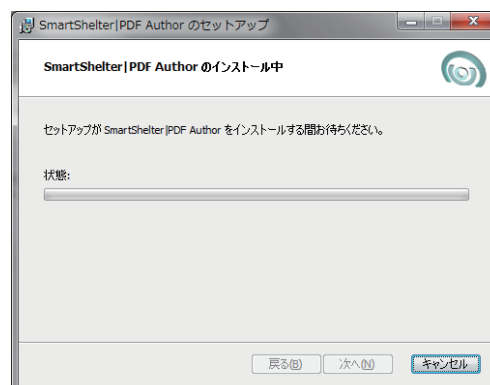
③ 「インストール」 ボタンをクリックする

SmartShelter|PDF Authorのインストール準備完了画面で「インストール」をクリックします。



④ インストールが開始される

SmartShelter |PDF Authorがインストールされます。



⑤ 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」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 を装着する

まずはじめに、貴社のコードメータFSBをPCに装着します。PDFファイルを暗号化する場合、必ずコードメータFSB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CDのTools¥SmartShelterPDFフォルダにあります。

ここでは、Acrobat X Pro の画面で説明します。

Acrobat 6/7/8/9 の場合も画面は異なりますが、基本的操作は同じです。

③ 「暗号化された文書を保存」をクリック

[ファイル]メニューから、「暗号化された文書を保存」メニューをクリックします。

[NOTE]

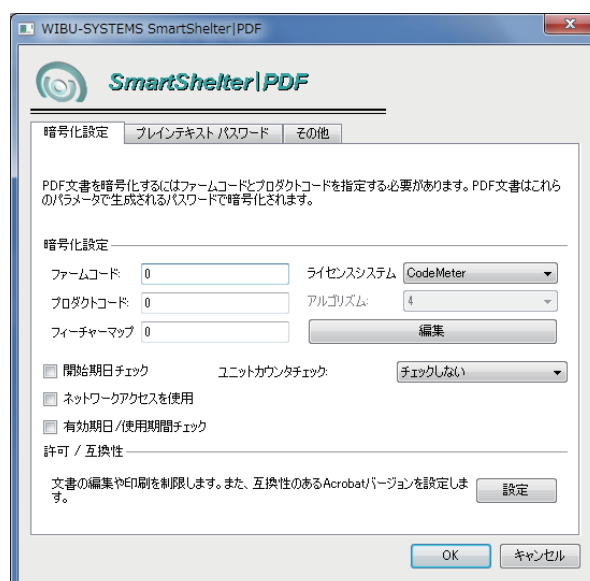
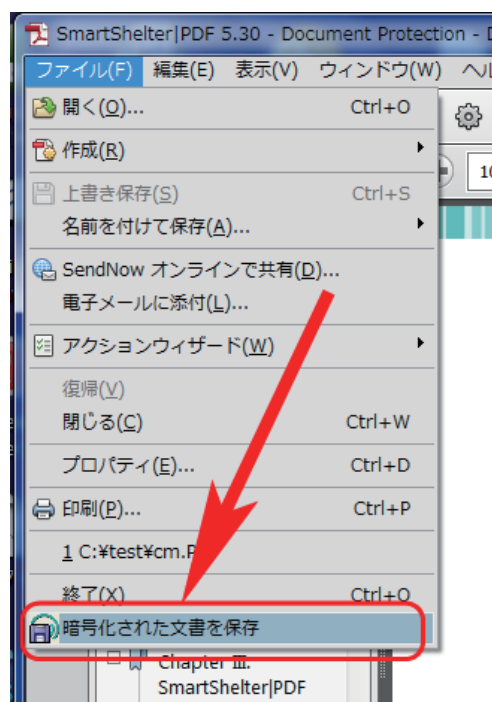
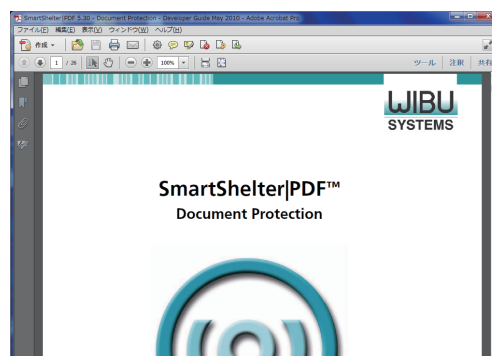
「暗号化された文書を保存」メニューが表示されない場合は、SmartShelter|PDF Authorが正しくインストールされていない場合が考えられます。再度、「4-4. SmartShelter|PDF Authorをインストールする」に戻り、SmartShelter|PDF Authorをインストールしてください。また、PDFファイルをAdobe Readerから開いていないかも確認してください。

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。

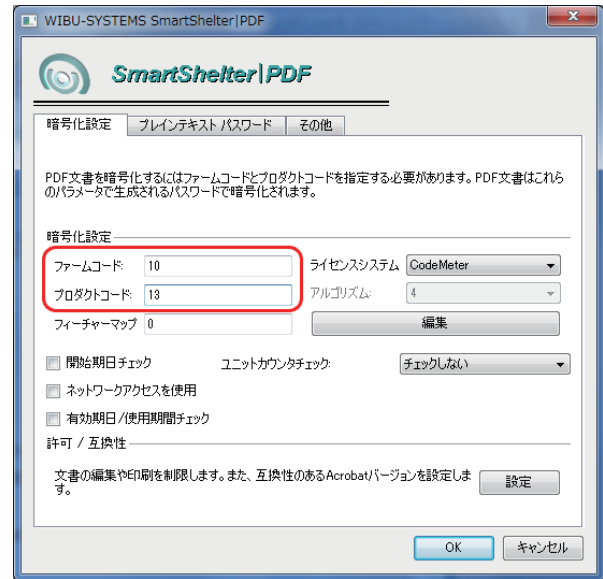
⑥ プロダクトコードを設定する

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、0～4294967295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 0～4294967295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

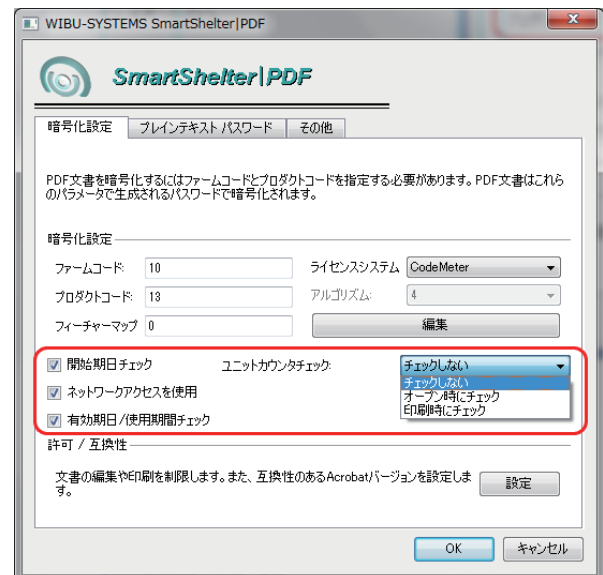
暗号化したPDFファイルをいつから開くことができるかの閲覧開始期日設定を行います。

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CM-Stick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。ただし、クライアントライセンス数を制限することができませんので、この機能を使用するとネットワーク(LAN)に参加しているすべてのクライアントからの閲覧が可能になります。ご注意ください。

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CM-Stick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

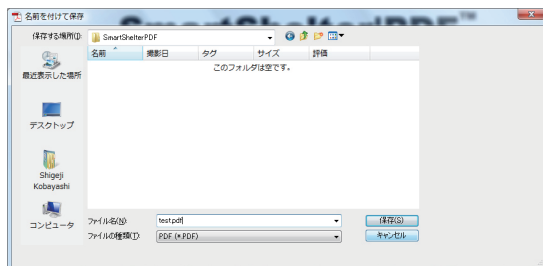
貴社のセキュリティニーズに従い、「許可/互換性」項目を設定します。

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。

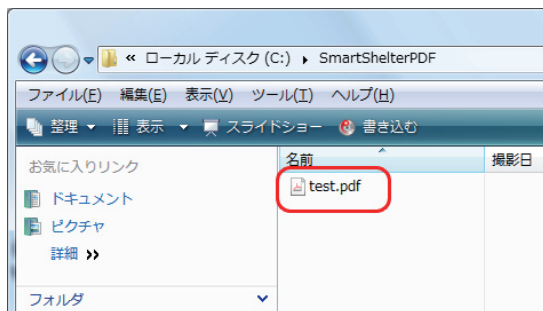
チェックを入れたら、「はい」をクリックし、暗号化設定画面に戻ります。

⑩ 暗号化処理を開始する

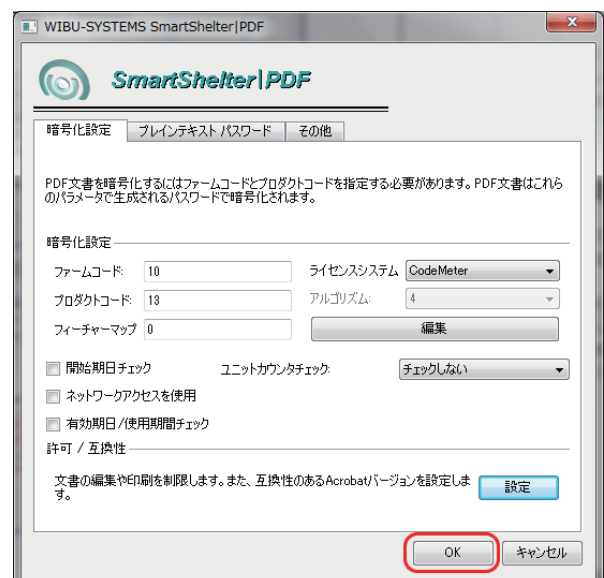
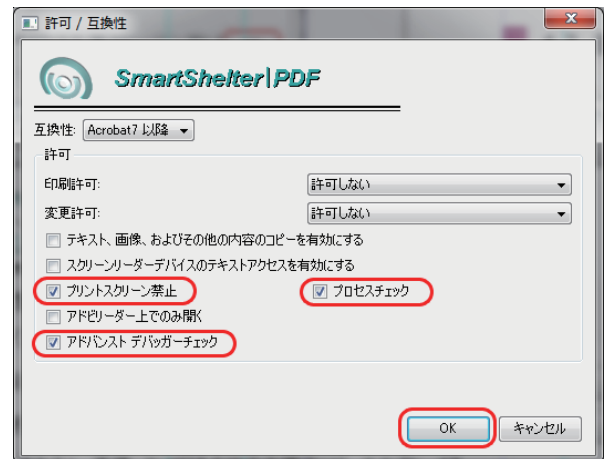
暗号化設定画面の「OK」をクリックすると、「名前を付けて保存」画面が表示されますので、ファイル名を入力して保存します。ここで保存されるときに、暗号化されたPDFファイルが作成されます。



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CM-Stickを装着して、暗号化されたtest.pdfを開いてみます。



13-6. CM-Stick にコードを登録する

コードメータCM-Stickにファームコード=10、プロダクトコード=13を登録します。CM-Stick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.CM-Stick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、CM-Stick, CM-Stick/M2GB, CM-ExpressCard, CM-PCCard, CM-CFCard, CM-SDCard, CM-Micro SDCardすべて共通です。

13-7. 動作を確認する

作成したCM-Stick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。CM-Stick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 CM-Stick 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CM-Stick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 CM-Stick がないと開かない

CM-Stick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。

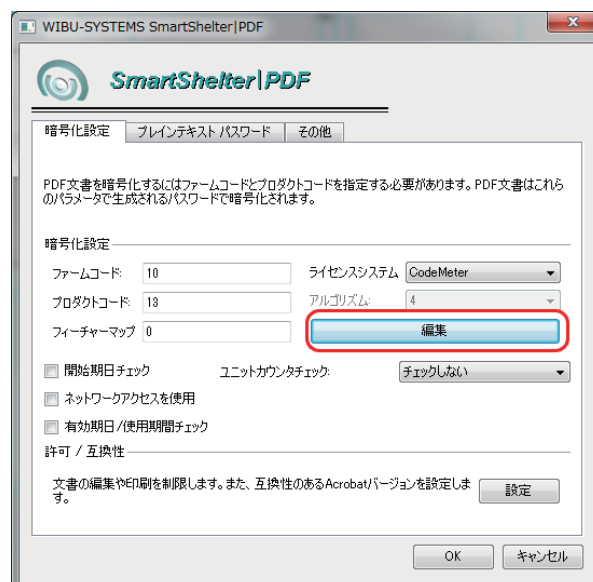


13-8. 複数のファームコード / プロダクトコードを入れて暗号化する

SmartShelterPDF Ver6.00から、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になりました。異なるファームコードやプロダクトコードを持つCM-Stick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。

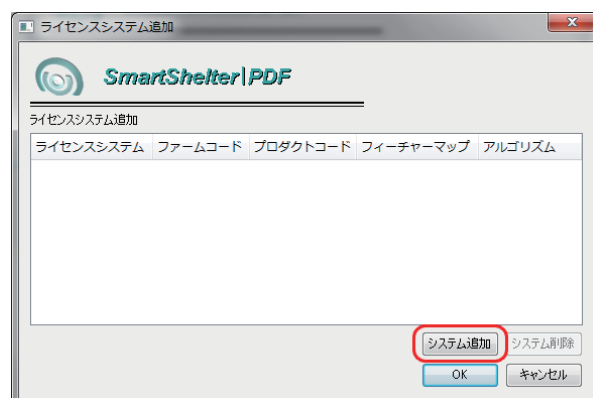
①「編集」ボタンをクリックする

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

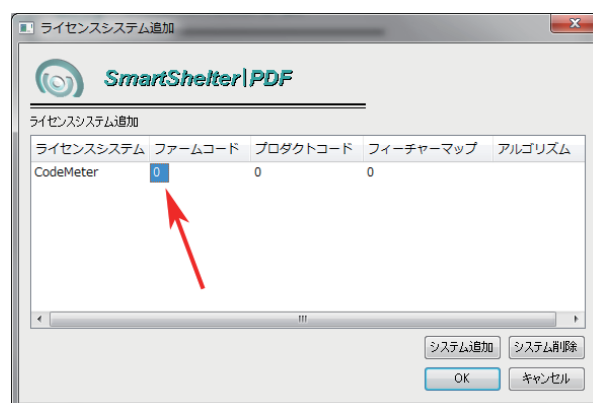
ライセンスシステム追加画面で「システム追加」ボタンをクリックします。



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。

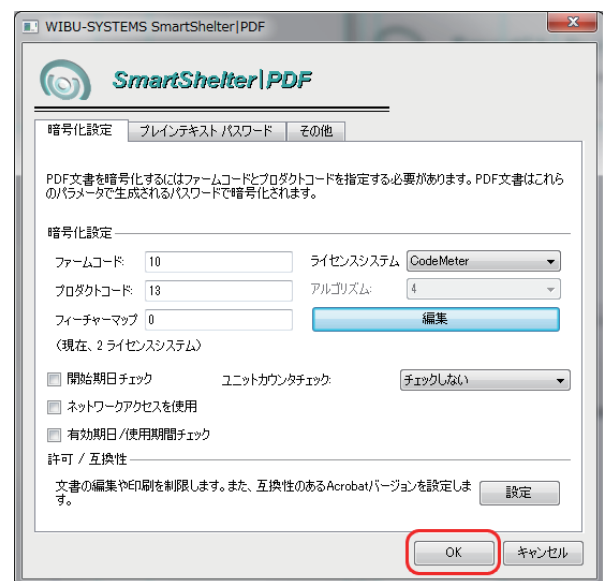
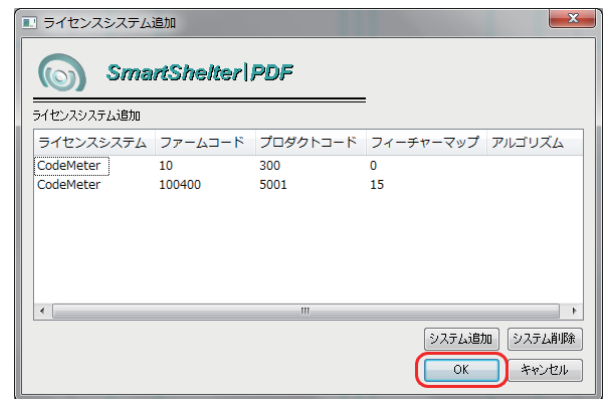
④ 「OK」 ボタンをクリックする

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CM-Stickが存在すればPDFファイルは開きます。



13-9. エラーメッセージをカスタマイズする

該当するCM-Stickが存在しない場合に表示するエラーメッセージをカスタマイズすることができます。

①「その他」タグを選択する

WIBU-SYSTEMS SmartShelter|PDF画面で「その他」タグを選択します。

② ユーザーメッセージ DLL ファイルを指定

ユーザーメッセージDLL名(拡張子なし)項目に、"UserMsgUs"と入力し、「OK」ボタンをクリックします。

[NOTE]

UserMsgUsと入力することで、UserMsgUs.dllが割り当てられます。

③ UserMsgJa.ini ファイルを編集する

日本語OS環境では、UserMsgJa.iniファイルがエラー表示に使用されます。各OS環境に応じてカスタマイズしてください。

UserMsg.ini (英語OS)
UserMsgDe.ini (ドイツ語OS)
UserMsgJa.ini (日本語OS)
UserMsgRu.ini (ロシア語OS)
UserMsgzh.ini (中国語OS)

[NOTE]

UserMsgUs.dllおよびUserMsgxx.iniは、暗号化されたPDFファイルを同じフォルダに保存してください。

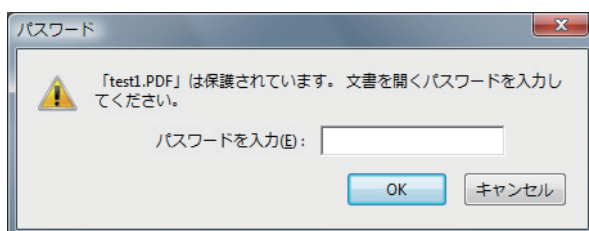
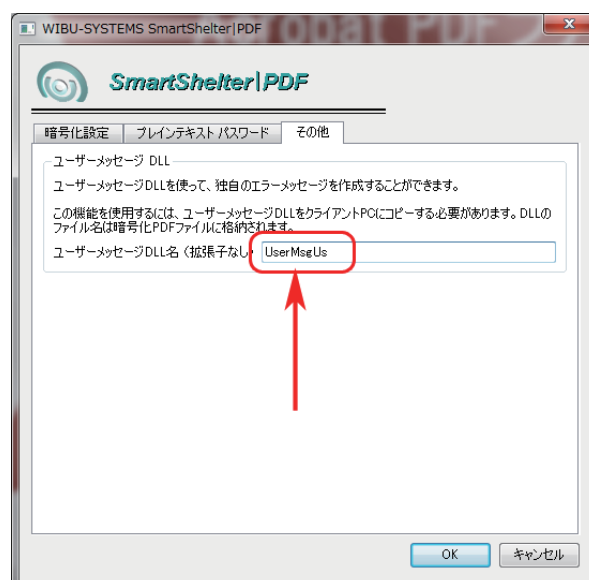
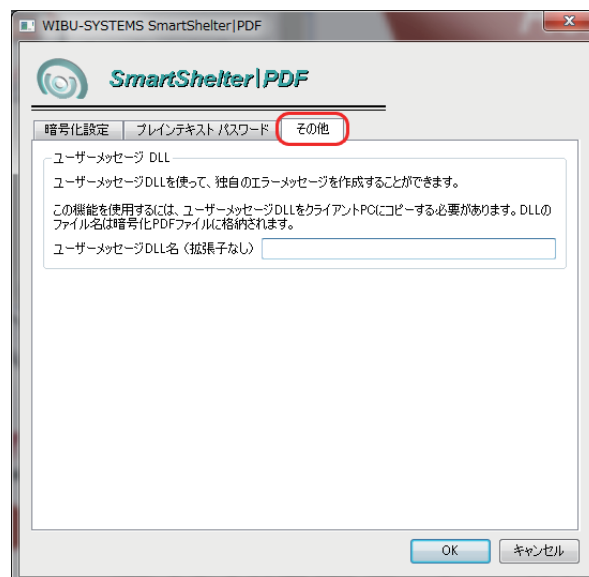
UserMsgUs.dllおよびUserMsgxx.iniが存在しないと、エラーメッセージは表示されません。

UserMsgUs.dllおよびUserMsgxx.iniは、弊社サイトからダウンロードできます。

<http://www.suncarla.co.jp/download/codemeter/SmartShelterPDF/v600/UserMsg.zip>

[NOTE]

コードメータランタイムキットまたはSmashPdfAuthor.exeまたはSmashPdfRdr.exeがインストールされていない場合、パスワード要求画面が表示されます。ここで要求されるパスワードは、PDFを暗号化する際に作成された管理者専用のプレインテキストパスワードまたはオーナーパスワードです。プレインテキストパスワードまたはオーナーパスワードを入力すると、CM-Stickが無くても暗号化されたPDFファイルを開くことが可能です。



【UserMsgJa.ini ファイルの説明】

[Main]

BuyUrl: WebサイトのURLを設定します。

Logo: 左部のメッセージ画面に表示されるBMP画像ファイルを指定します。(BMPのみ有効)

MainText: エラーメッセージ本文を入力します。改行は¥nで行います。

BuyText: 購入サポート窓口情報などを入力します。

OkButton: メッセージ画面の[OK] ボタンの名前を設定します。

CancelButton: メッセージ画面の[キャンセル] ボタンの名前を設定します。

RetryButton: メッセージ画面の[再試行] ボタンの名前を設定します。

IgnoreButton: メッセージ画面の[無視] ボタンの名前を設定します。

BuyNowButton: メッセージ画面の[購入] (HPへのリンク) ボタンの名前を設定します。

BuyHint:

BuyHint=on BuyTextメッセージを表示

Buyhint=off BuyTextメッセージを非表示

UnitCounterMax = プログレスバーに表示するユニットカウンタ最大数 (例:1000)

DaysMax = プログレスバーに表示する有効日数最大数 (例:100)

UnitCounterText = ユニットカウンタのタイトル (例:回数:)

ExpirationDateText = 有効期限のタイトル (例:日数:)

;***** CodeMeter ERROR CODES ***** (BEGIN)

[CM32]

Caption=CodeMeter - 使用可能なクレジットがありません!

MainText=このソフトウェアはペイパーユース版で、すべてのクレジットが使用されました。\\n\\n"キャンセル" ボタンをクリックしてソフトウェアを終了してください。

HeadLine=使用可能なクレジットがありません!

[CM35]

Caption=CodeMeter - ライセンスの有効期限が終了しました!

MainText=このソフトウェアは使用有効期限が設定されたバージョンです。使用有効期限が終了しました。\\n\\n"キャンセル" ボタンをクリックしてソフトウェアを終了してください。

HeadLine=ライセンスの有効期限が終了しました!

[CM37]

Caption=CodeMeter - ソフトウェアはまだアクティベートされていません!

MainText=このソフトウェアは使用有効期限付きバージョンで、まだアクティベーションが行われていません。\\n\\n"キャンセル" ボタンをクリックし、ソフトウェアを終了してください。

HeadLine=まだアクティベートされていません!

[CM71]

Caption=CodeMeter - 使用期間が過ぎました!

MainText=使用期間が過ぎたためアプリケーションを起動することができません。

HeadLine=使用期間が過ぎました!

[CM200]

Caption=CodeMeter - ライセンスが見つかりません! [CM200]

MainText=このソフトウェアに必要な #FirmProductCode# を持つCM-Stickが見つかりません。\\n\\n正しいCM-StickをPCに装着してください。

HeadLine=ライセンスが見つかりません!

[CM52]

Caption=CodeMeter - CM-Stick使用不可!

MainText=接続されているCM-Stickは使用不可の状態です。\\n\\n使用可能な状態にして、"再試行"ボタンをクリックしてください。

HeadLine=CM-Stick使用不可!

[CM106]

Caption=CodeMeter - ライセンスが見つかりません![CM106]

MainText=このソフトウェアに必要な #FirmProductCode# を持つCM-Stickが見つかりません。\\n\\n正しいCM-StickをPCに装着してください。

HeadLine=ライセンスが見つかりません!

[CM212]

Caption=CodeMeter - 最大ユーザー数に達しました!

MainText=利用できる #FirmProductCode# ライセンスがありません。\\n\\n"再試行" をクリックして利用可能なライセンスをスキャンしてください。

HeadLine=最大ユーザー数に達しました!

[CM255]

Caption=CodeMeter - 不明なエラー!

MainText= 不明なCM-Stickエラー、エラー(255)。

HeadLine=ライセンスが見つかりません!

,***** CodeMeter ERROR CODES ***** (END)

13-10. 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場合、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、CM-Stick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。

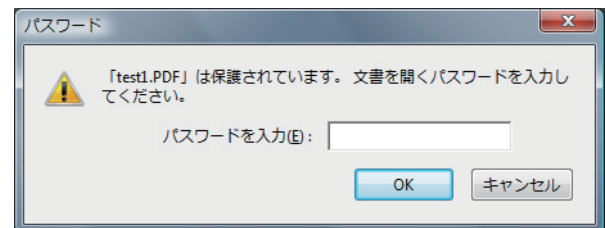
パスワードには、プレインテキストパスワードとオーナーパスワードの2種類があります。

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。



13-11. 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、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。

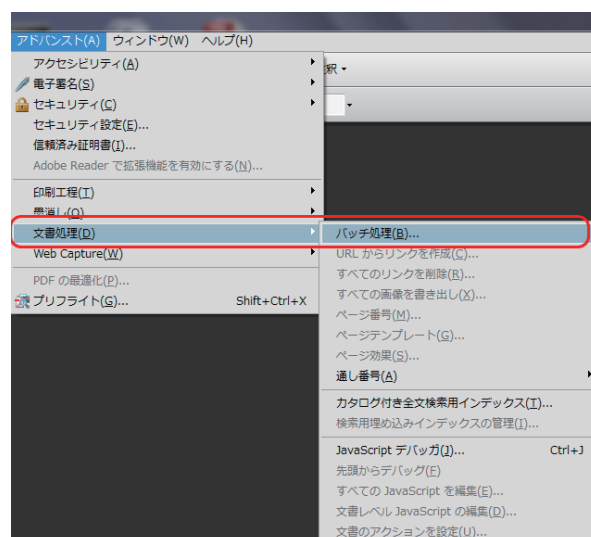
Acrobat 9 ProとAcrobat X Proでのバッチ処理方法をご説明いたします。その他のAcrobatバージョンでも、メニュー画面が異なりますが、基本的操作は同じです。

Acrobat 9 Pro の場合

①「バッチ処理」メニューを選択する

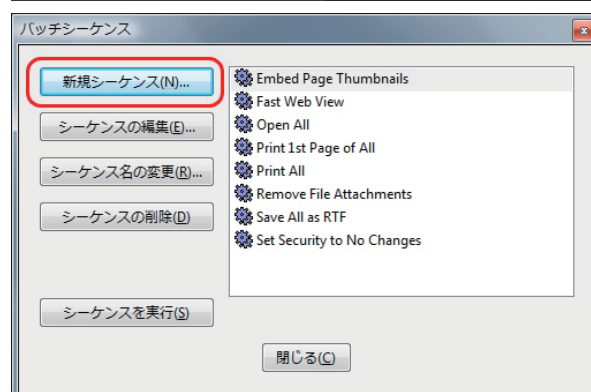
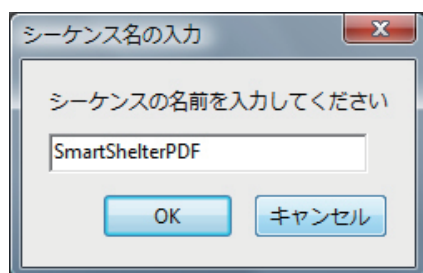
Acrobatの「アドバンスト」/「文書処理」メニューから「バッチ処理」を選択します。Acrobat6/7/8の場合、メニュー画面が多少異なりますが、「バッチ処理」を選択してください。

「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。



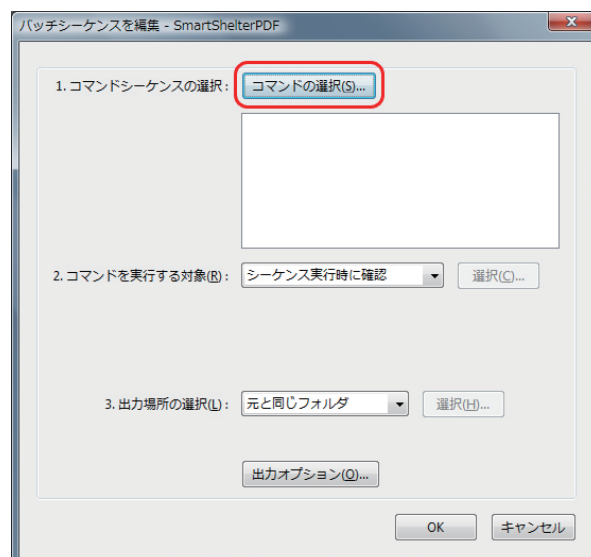
②「新規シーケンス」をクリックする

「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。



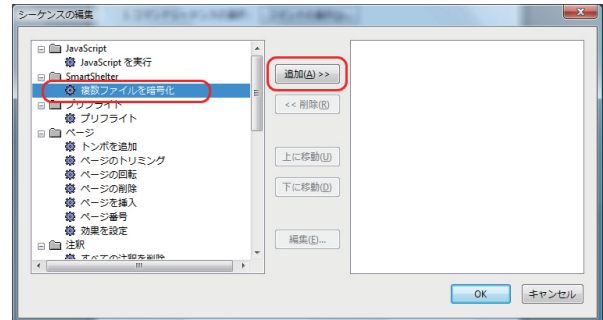
③「コマンドの選択(S)」をクリックする

「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。

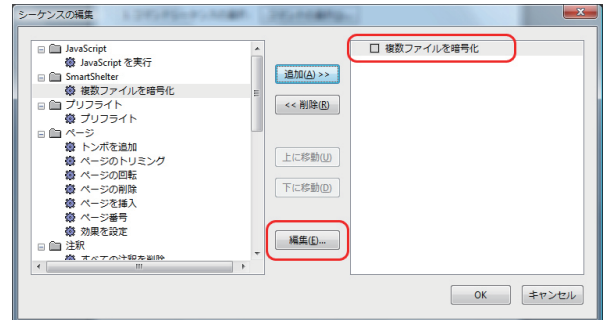


④ 「複数ファイルを暗号化」を選択する

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。

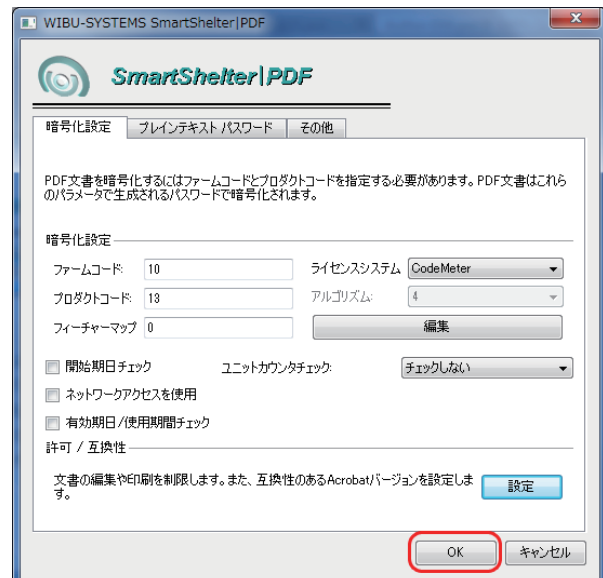
**⑤ 「編集 (E)」 ボタンをクリックする**

「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**

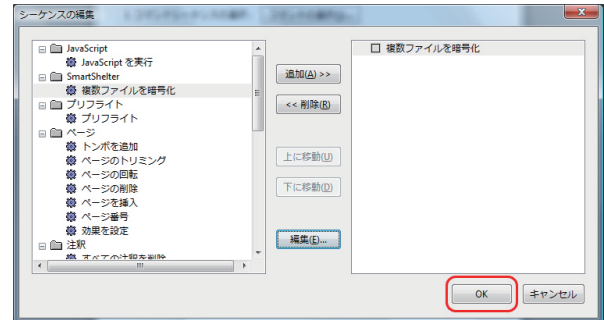
SmartShelterPDFの暗号化内容の設定画面が開きますので、ここで暗号化設定を行います。必要に応じてオプション項目を設定します。

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

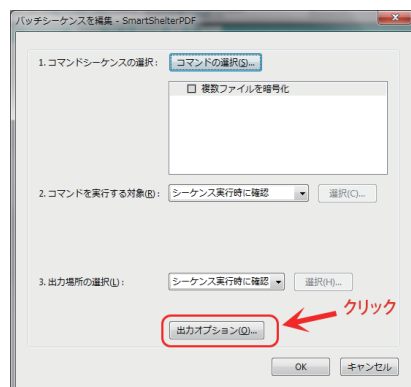
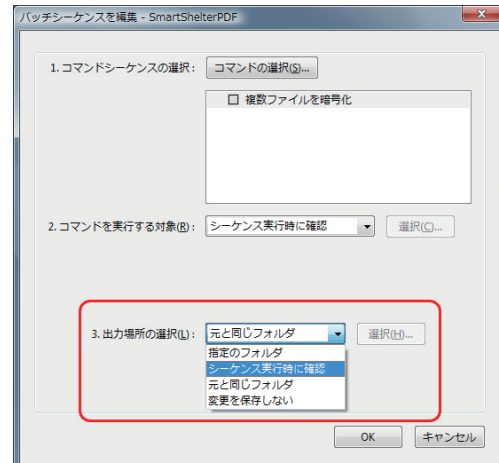
シーケンスの編集画面に戻り、「OK」ボタンをクリックします。



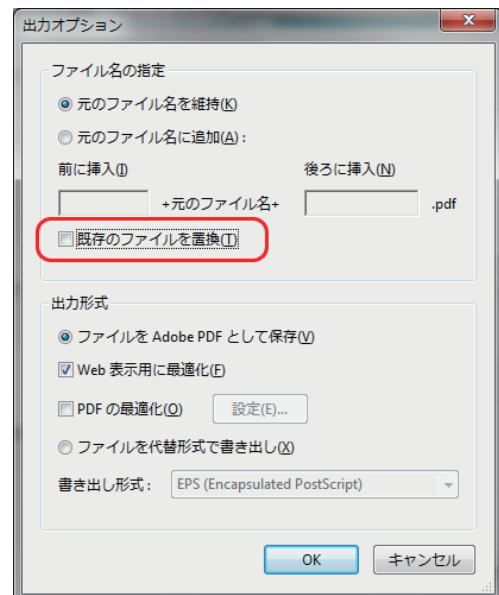
⑧ バッチシーケンスを編集する

「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是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。

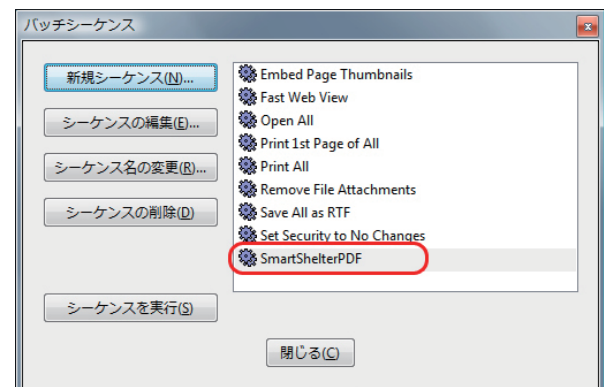


⑨ SmartShelterPDF が登録された

「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。



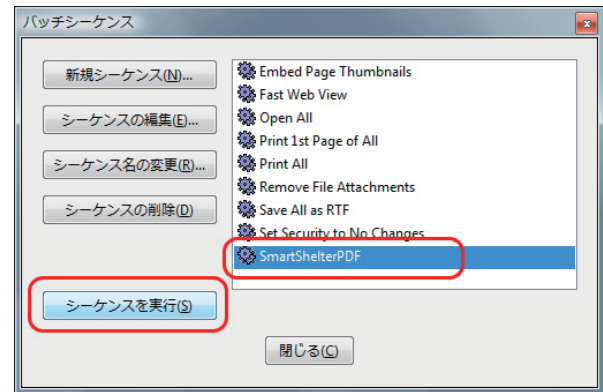
バッチシーケンスを実行して、PDF ファイルを一括暗号化する

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。

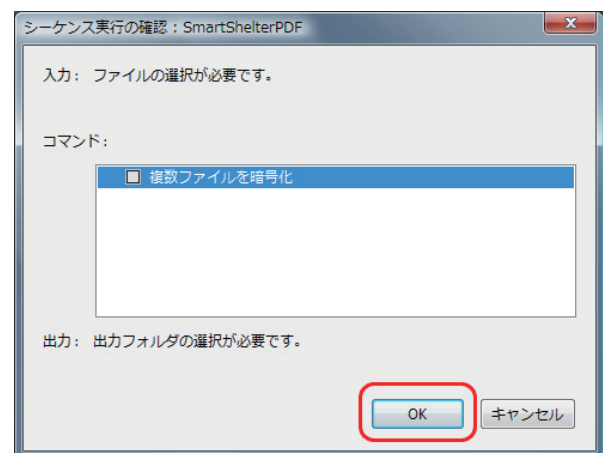


③ バッチシーケンスを実行する

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。

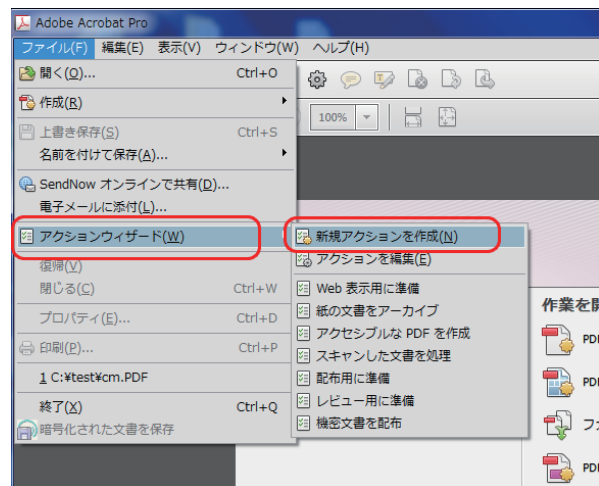


あとは、暗号化するPDFファイルを選択し、指定するフォルダに作成するだけです。

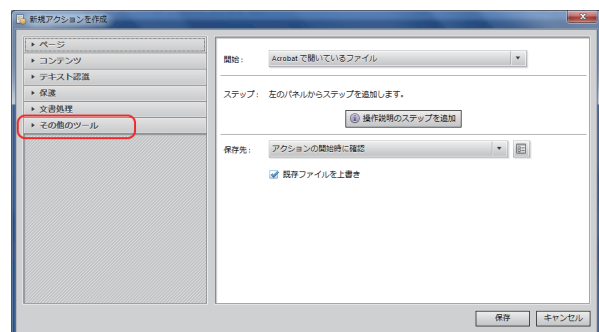
Acrobat X Pro の場合

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。

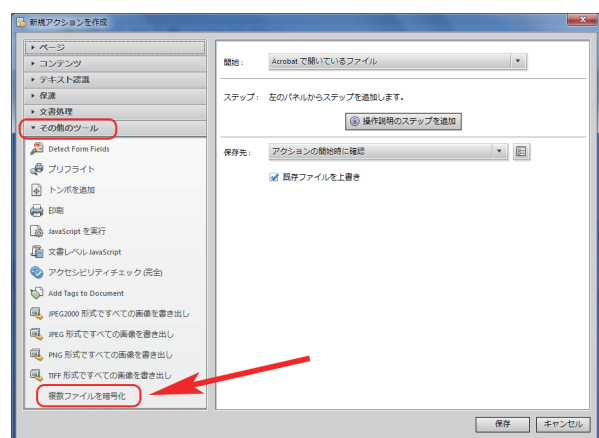


「新規アクションを作成(N)」をクリックすると、「新規アクションを作成」画面が表示されます。



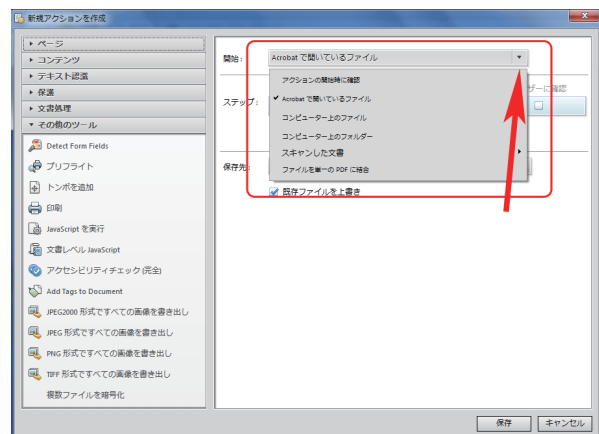
②「複数ファイルを暗号化」を選択する

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。



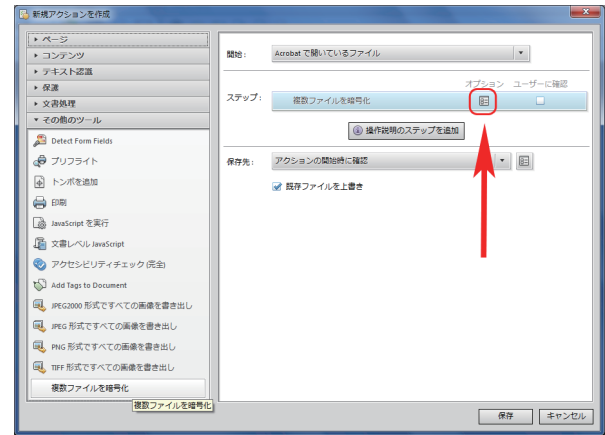
③ 暗号化するファイルを選択する

「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。

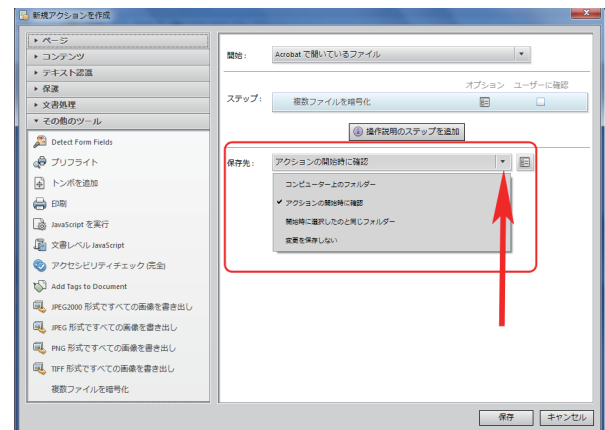
**⑤ 暗号化内容を設定する**

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。

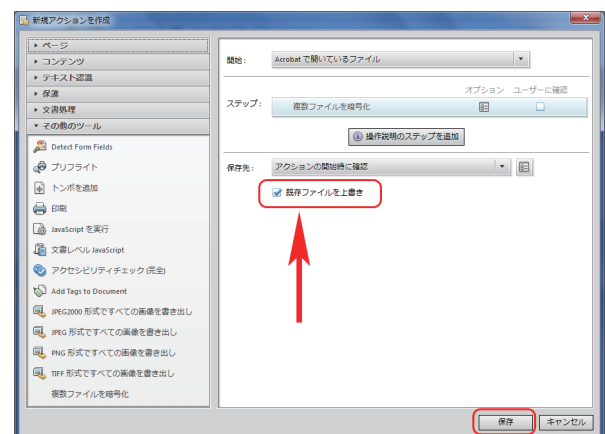
**⑥ 保存先を指定する**

暗号化されたファイルの保存先を指定します。



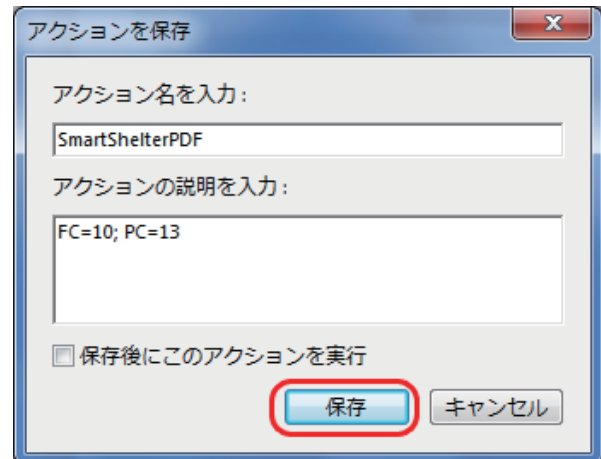
既存ファイルを上書きする場合は、「既存ファイルを上書き」にチェックを入れます。

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

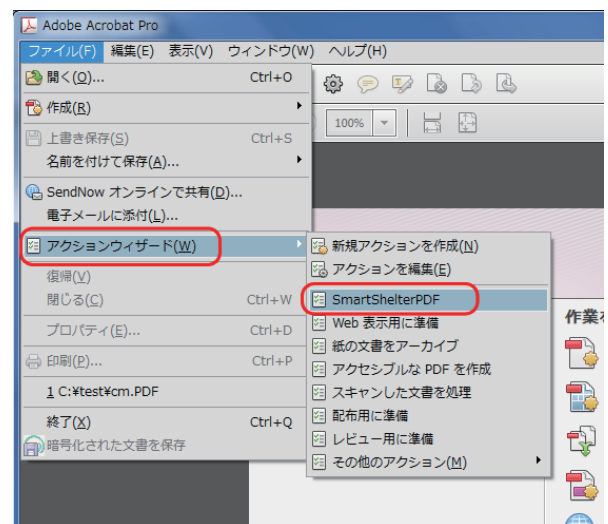
アクションを保存画面で、アクション名を入力します。必要に応じて、アクションの説明を入力してください。入力完了したら「保存」ボタンをクリックします。これで、アクションは保存されます。



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。



13-12. 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、¥Program Files¥SmartShelter PDFに存在します。

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

[注意]

-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。

[注意]

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。

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	フォ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
PRTSCEEN	プリントスクリーンをブロックする
EXT_PRTSCEEN	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。

13-13. 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 CM-Stick（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 (CodeMeterRuntime32.exeまたはCodeMeterRuntime64.exe)
- ③ SmartShelterPDFランタイムキット (SmashPDFRdr.exe)
- ④ 暗号化済みPDFファイル

CodeMeterRuntime32.exe/CodeMeterRuntime64.exeは、コードメータCDのCD-User¥Runtimeフォルダにあります。SmashPDFRdr.exeは、コードメータCDのSmartShelterPDFフォルダにあります。

Mac OS Xの場合：

- ① CM-Stick（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_4.30.485.503.dmg)
- ③ SmartShelterPDFランタイムキット (SmashPdfRdr_6.0.64.dmg)
- ④ 暗号化済みPDFファイル

CmRuntimeUser_4.30.485.503.dmgは、コードメータCDのCD-User¥MacOSフォルダにあります。
SmashPdfRdr_6.0.64.dmgは、コードメータCDのSmartShelterPDFフォルダにあります。

CM-Stick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、著作権上問題ありません。

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 XP/Vista/7 (32bit/64bit) Windows Server 2003/2008 (32bit/64bit)

Mac OS X 10.4,10.5,10.6

Adobe Acrobat 6/7/8/9/X(32bit/64bit) またはAdobe Reader 6/7/8/9/X (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

SmartShelter|PDF Reader Plug-in "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)

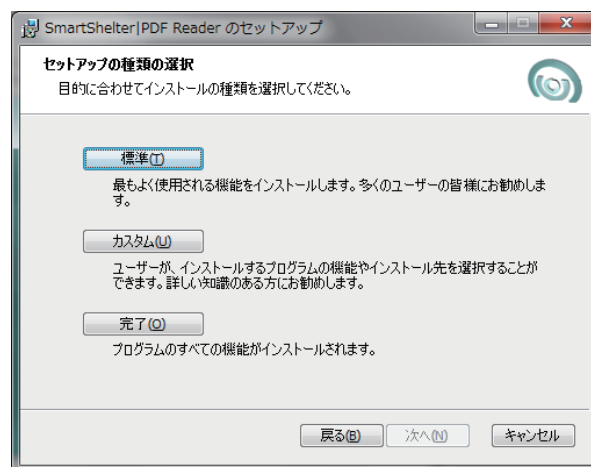
1台のPCにAdobe AcrobatとAdobe Readerの2つが存在している場合、プラグインソフト"SmashPDFRdr.api"はAdobe AcrobatとAdobe Readerそれぞれに自動インストールされます。

カスタム

SmartShelter|PDF Reader Plug-in "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 用 SmartShelterPDF ランタイムキット "SmashPdfRdr_6.0.64.dmg" について

Mac OS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_6.0.64.dmg" をダブルクリックする

Mac上で"SmashPdfRdr_6.0.64.dmg"をダブルクリックします。

② Adobe Reader または Adobe Acrobat を起動する

Adobe ReaderまたはAdobe Acrobatを起動します。

③ 「Finder に表示」 をクリックする

Adobeアイコンを右クリックし、"Finderに表示"をクリックします。

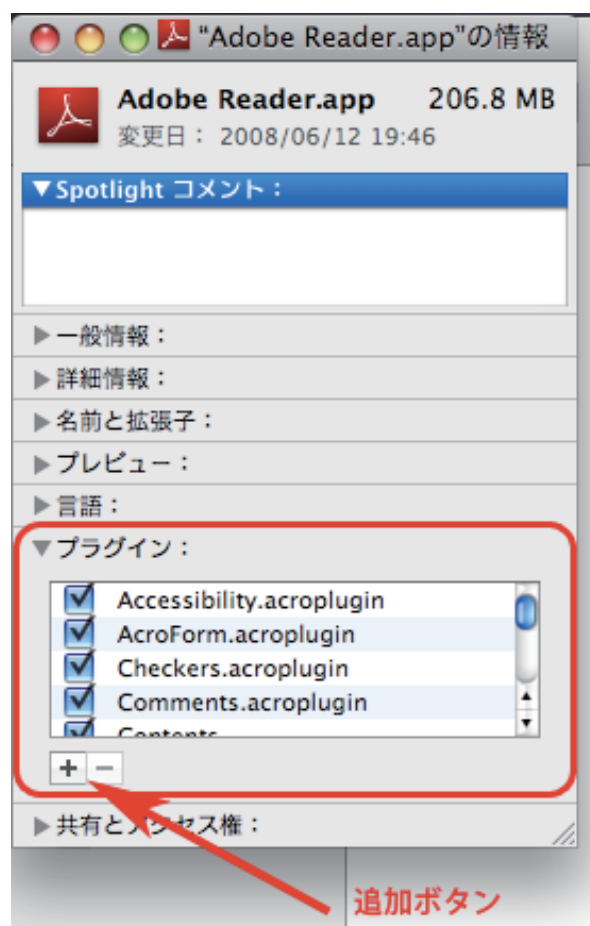


④ 「情報を見る」をクリックする

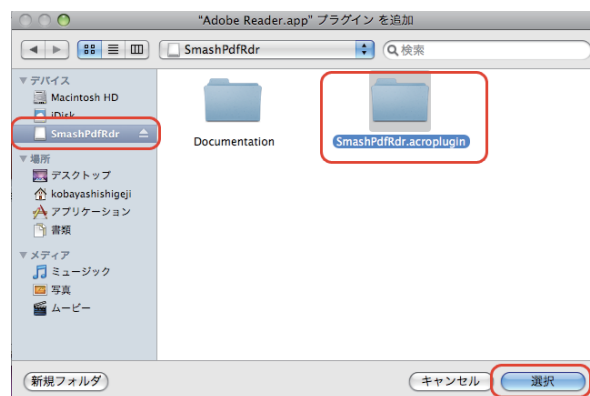
Adobe Reader (またはAcrobat)のアイコンを右クリックし、"情報を見る"をクリックします。

**⑤ プラグインで追加ボタンをクリックする**

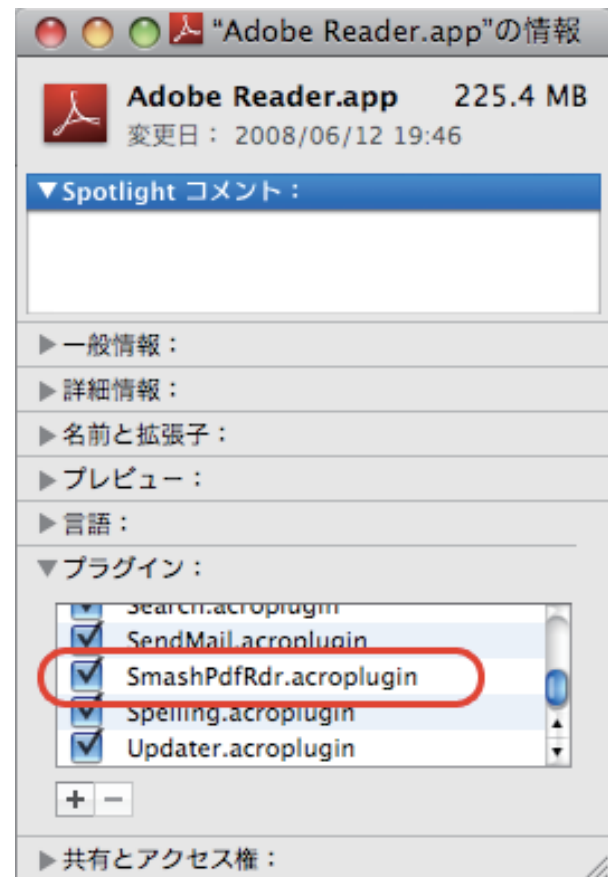
"Adobe Reader.app"の情報上の"プラグイン"で追加ボタン(+)をクリックします。

**⑥ SmashPdfRdr.acroplugin を選択する**

「"Adobe Reader.app"プラグインを追加」画面で "SmashPdfRdr.acroplugin"を選択します。

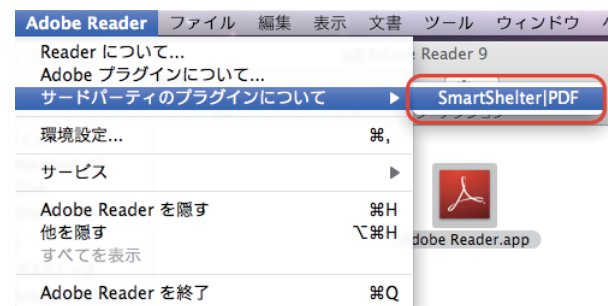


⑦ SmashPdfRdr.acroplugin がインストールされた
"Adobe Reader.app"の情報画面の"プラグイン"に、
"SmashPdfRdr.acroplugin"がインストールされました。
これで、SmartShelterPDFランタイムキットのインストールは完了です。



⑧ 確認する

SmartShelterPDFランタイムキットが正しくインストールされると、「サードパーティのプラグインについて」に"SmartShelter|PDF"が登録されます。もし、こ
こで表示されない場合は、インストールが正常に行われておりませんので、再度インストール作業を行って
ください。

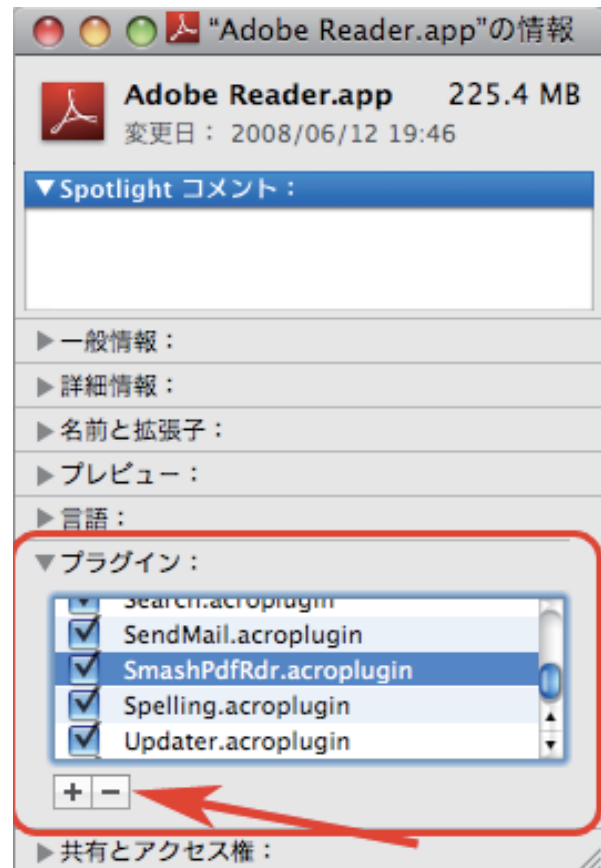
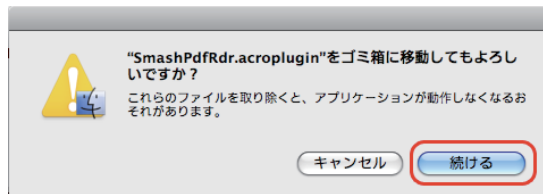


[NOTE]

SmartShelterPDFランタイムキットが正しくインストールされていないと、暗号化されたPDFファイルが
開きませんのでご注意ください。

Mac OS 用 SmartShelterPDF ランタイムキットをアンインストールする場合

SmartShelterPDFランタイムキットをアンインストールする場合は、"Adobe Reader.app"の情報画面の"プラグイン"で"SmashPdfRdr.acroplugin"を選択し、削除ボタン(－)をクリックします。下記画面が表示されますので、「続ける」ボタンをクリックするとSmartShelterPDFランタイムキットがアンインストールされます。

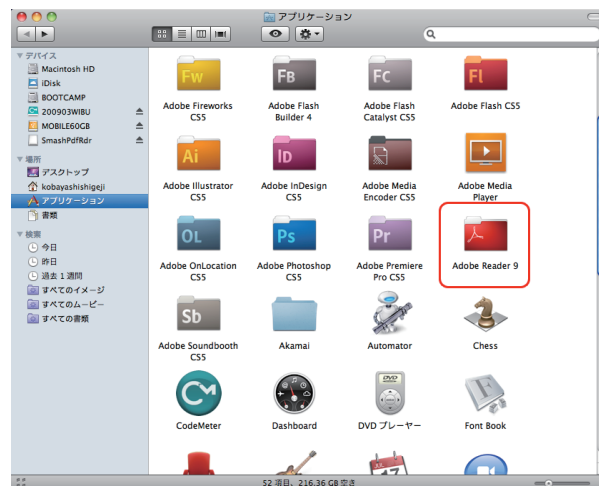


Mac OS X 10.6 (Snow Leopard) の場合

Mac OS X 10.6 (Snow Leopard)の場合、"Adobe Reader.app"の情報画面で"プラグイン"が表示されませんので、下記の手順にてプラグインをインストールしてください。

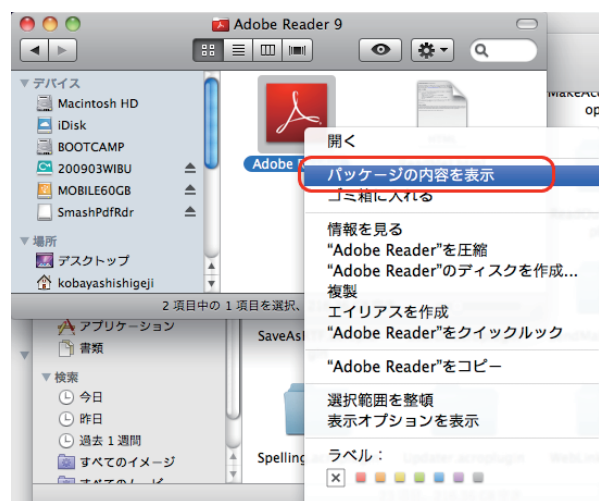
① アプリケーションにて "Adobe Reader 9" を選択する

アプリケーションウィンドウにて"Adobe Reader 9"を選択します。



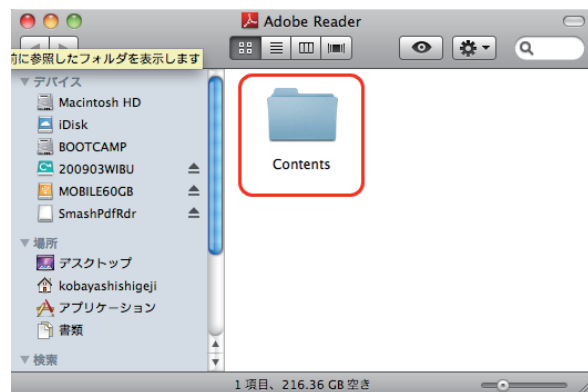
② 「パッケージの内容を表示」を選択する

Adobe Reader 9 アイコン上で右クリックをしてメニューを表示し、「パッケージの内容を表示」を選択します。



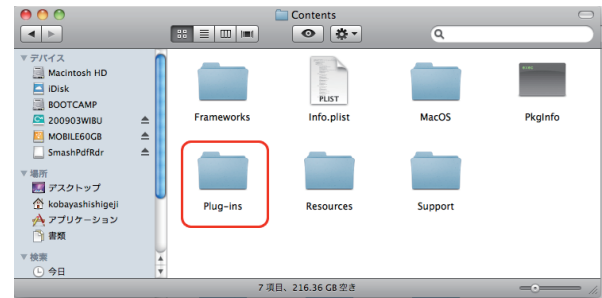
③ "Contents" フォルダをクリックする

"Contents"フォルダをクリックします。



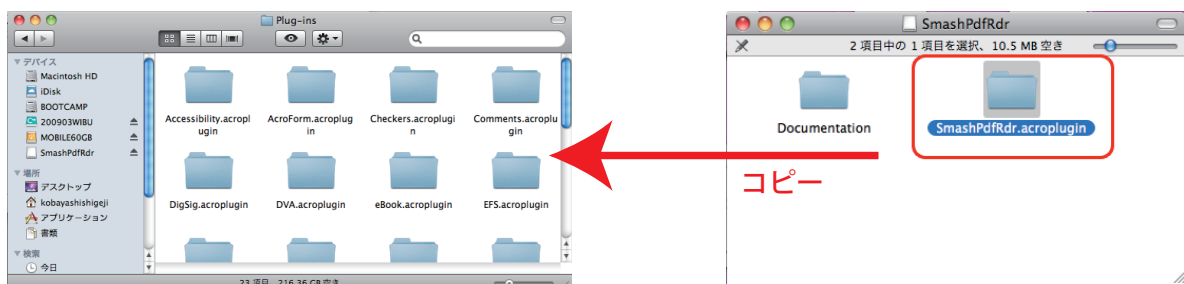
④ "Plug-ins" フォルダをクリックする

"Plug-ins"フォルダをクリックします。



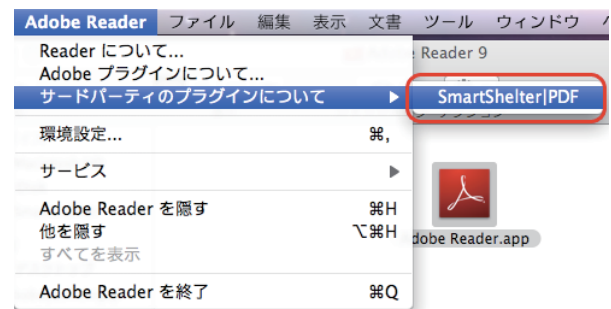
④ "SmashPdfRdr.acroplugin" をコピーする

"SmashPdfRdr.acroplugin"をPlug-inフォルダにコピーします。SmashPdfRdr.acropluginは、SmashPdfRdr_6.0.64.dmgをダブルクリックすると展開されます。



これで、SmashPdfRdr.acropluginのインストールは完了です。「サードパーティのプラグインについて」に「SmartShelter|PDF」が表示されていることを確認してください。

また、SmashPdfRdr.acropluginをアンインストールする場合は、Plug-in画面でSmashPdfRdr.acropluginを削除します。



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 (32bit 版) を配布する
- 14-3. Windows アプリケーション (64bit 版) を配布する
- 14-4. .NET アプリケーション (32bit/64bit 版) を配布する
- 14-5. 暗号化された PDF ファイルを配布する
- 14-6. 暗号化された Flash ファイルを配布する
- 14-7. Mac OS X アプリケーションを配布する
- 14-8. Linux アプリケーションを配布する
- 14-9. 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著作権上問題ありません。

コードメータランタイムキットは、コードメータCD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 (32bit 版) を配布する

32bit版のWindowsアプリケーションを配布する場合は、32bitOS/64bitOSによってインストールするファイルが異なります。

Windows OS(32bit)の場合

(Windows XP/Vista/7/8, Windows Server 2003/2008)

①32bit用コードメータランタイムキット "CodeMeterRuntime32.exe"

Windows OS(64bit)の場合

(Windows XP/Vista/7/8 Server 2003/2008R2/2012)

①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe"

14-3. Windows アプリケーション (64bit 版) を配布する

64bit版のWindowsアプリケーションを配布する場合は、64bit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。

Windows XP/Vista/7/8(64bit), Windows Server 2003/2008R2/2012 (64bit)

①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe"

14-4. .NET アプリケーション (32bit/64bit 版) を配布する

Windows OS(32bit)の場合

32bit用コードメータランタイムキット"CodeMeterRuntime32.exe"

Windows OS(64bit)の場合

64bit用コードメータランタイムキット "CodeMeterRuntime64.exe"

[NOTE]

コードメータランタイムキット Ver4.30から、CodeMeter .NET assembly (Win32/64) "WibuCmNET.msi"は、CodeMeterRuntime32/64.exeに含まれます。

14-5. 暗号化された PDF ファイルを配布する

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)の場合

- ① 32bit用コードメータランタイムキット"CodeMeterRuntime32.exe"
- ② SmartShelterPDFランタイムキット "SmashPDFRdr.exe"

Windows OS(64bit)の場合

- ① 64bit用コードメータランタイムキット"CodeMeterRuntime64.exe"
- ② SmartShelterPDFランタイムキット "SmashPDFRdr.exe"

動作環境

OS: Windows XP/Vista/7/8 (32bit/64bit), Windows Server 2003/2008 R2 /2012(32bit/64bit)
Adobe Acrobat 6/7/8/9/X または Adobe Reader 6/7/8/9/X

14-6. 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 10.4, 10.5, 10.6, 10.7,10.8

- ① Mac用コードメータランタイムキット"CmRuntimeUser_4.50.909.504.dmg" (16MB)

14-7. Linux アプリケーションを配布する

RPM package for SuSe, Red Hat etc

32bit:

CodeMeter-4.30.485-503.i386.rpm

AxProtector-7.11.463-1.i386.rpm (AxProtectorで暗号化したアプリケーション用)

64bit:

CodeMeter64-4.30.485-503.x86_64.rpm

AxProtector64-7.11.463-1.x86_64.rpm (AxProtectorで暗号化したアプリケーション用)

DEB package for Debian, Ubuntu etc

32bit:

codemeter-4.30.485.503-i386.deb

codemeter-lite_4.30.485.503_i386.deb (Pure driver installation for systems without GUI)

axprotector-7.11.463_i386.deb (AxProtectorで暗号化したアプリケーション用)

64bit:

codemeter64_4.30.485.503_amd64.deb

axprotector64_7.11.463_amd64.deb (AxProtectorで暗号化したアプリケーション用)

14-8. Sun Solaris アプリケーションを配布する

SPARC / SPARCV9

CodeMeter Runtime

codemeter_4.30-sol-SPARC.tar.bz2

CodeMeter Runtime 64-Bit Extension

codemeter_4.30-sol-SPARCV9.tar.bz2

i386 / amd64

CodeMeter Runtime

codemeter_4.30-sol-i386.tar.bz2

CodeMeter Runtime 64-Bit Extension

codemeter_4.30-sol-x64.tar.bz2

サンカーラ株式会社

〒103-0013 東京都中央区日本橋人形町3-3-12 人形町103ビル2F

TEL: 03-3249-3421 / Fax:03-3249-3444

E-mail: support@suncarla.co.jp

www.suncarla.co.jp