CodeMeter
コードメータ

ユーザーズガイド　Ver. 4.20

No. 5004

2010 年 6 月 19 日

サンカーラ株式会社
www.suncarla.co.jp

目次

Chapter 1

はじめに

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. IxProtector/WUPI

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

4. 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。

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ことで、オフラインの状態で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。

11. 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。

12. CmIdentity (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"CmIdentity"（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。

13.　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。

14. RoHS 指令対応済み

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

15. ISO9001:2000 認証取得

コードメータは、ISO9001:2000認証されたWIBU-SYSTEMS社（ドイツ）で開発・製造されており、高い品質と信頼性を実現した商品です。

16. その他、多くの高度なセキュリティを実現

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

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

コードメータ開発ツールのインストール作業は以下の３ステップを行ないます。

1. Microsoft .NET Framework 2.0以降をインストールする
2. CodeMeter SDKをインストールする
3. 貴社のライセンスファイルCmFirm.wbcをコピーする

2-2. Microsoft .NET Framework 2.0 以降をインストールする

コードメータの開発作業には、マイクロソフト社の.NET Framework 2.0以降が必要になります。開発作業を行うPCに.NET Framework2.0以降がインストールされていない場合は、CodeMeter SDKをインストールする前に.NET Framework2.0以降をインストールしてください。下記マイクロソフト社Webサイトから最新バージョンを入手できます。すでに、.NET Framework2.0以降がPCにインストールされている場合はこの作業は不要です。なお、Windows Vista/7をお使いの場合は、すでに.NET Framework 3.0以降がインストールされていますので、そのまま次の「2-3. CodeMeter SDKをインストールする」に進んでください。

http://msdn.microsoft.com/ja-jp/netframework/aa569263.aspx

2-3. 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

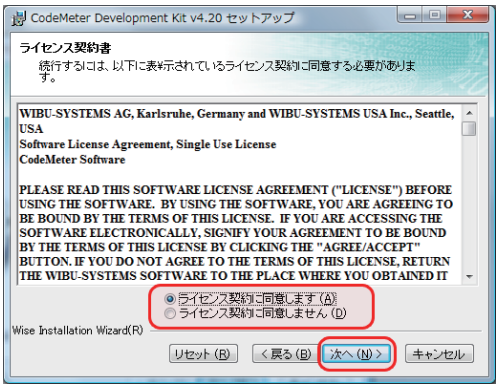
"CodeMeter SDK"をクリックします。
あとは、メッセージに従い、コードメータ開発キットをインストールします。

③ CodeMeter Development Kit v4.20 をインストールする

「CodeMeter Development Kit v4.20 Installation Wizardへようこそ」画面が表示されますので、「次へ」をクリックします。

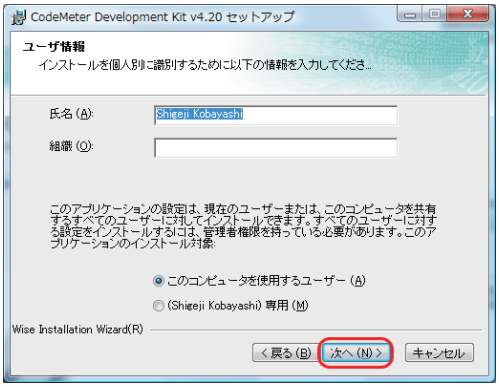
④ライセンス契約書

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。



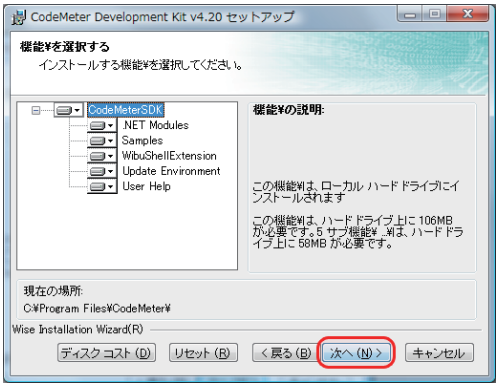
⑤ ユーザー情報を入力する

ユーザー名などを入力します。



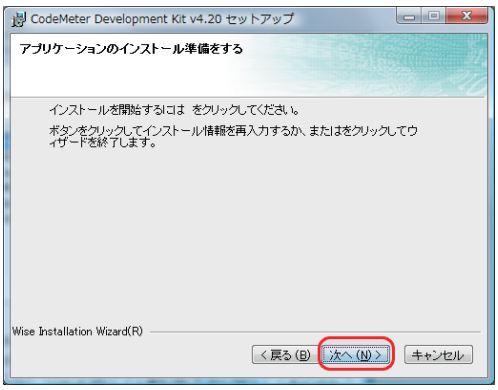
⑥インストールする機能を選択する

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



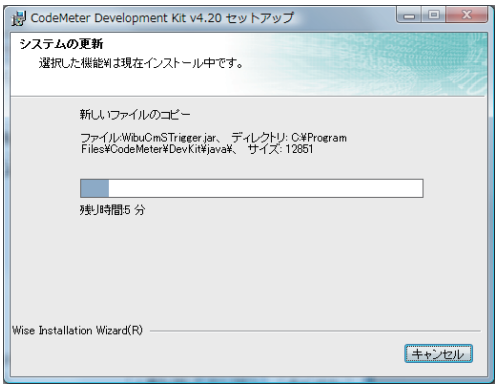
⑦インストールを開始する

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。

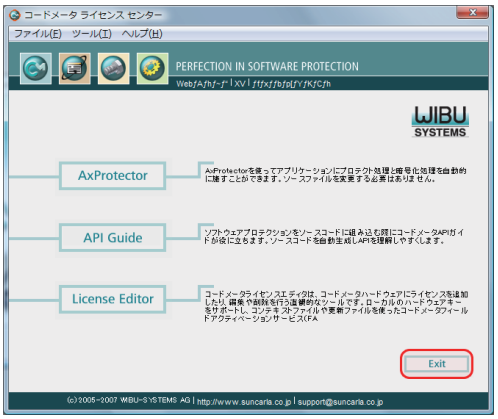


⑨正常にインストールされた

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。「終了」ボタンをクリックして画面を閉じます。

⑩ コードメータライセンスセンター画面を閉じる

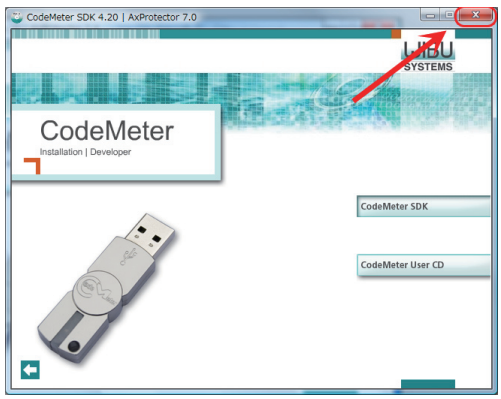
インストール作業の途中で、右のコードメータライセンスセンター画面が開きますが、ここでは使用しませんので、右下の"Exit"をクリックして画面を閉じます。



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をコピーします。次の「2-4. 貴社のライセンスファイルCmFirm.wbcをコピーする」にお進みください。評価版の場合は、これでインストール作業は終了です。

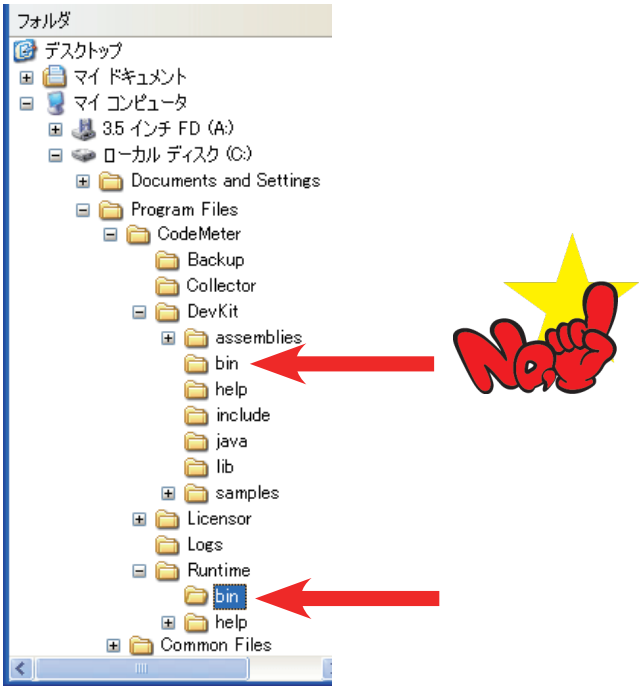


2-4. 貴社のライセンスファイル CmFirm.wbc をコピーする

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、コードメータ開発キットをインストールしたPCの¥Program Files¥CodeMeter¥Runtime¥binの中にエクスプローラ等を使ってコピーします。すでに、binフォルダの中に評価版用CmFirm.wbcファイルが存在しますので、貴社のライセンスファイルCmFirm.wbcをそのまま上書きコピーします。

［注意］
「評価版」をご利用のお客様は、この作業は不要です。すでにbinフォルダに存在している評価用CmFirm.wbcファイル(Firm Code = 10)をそのままご使用ください。

［注意］
binフォルダは、Program Files¥CodeMeter¥DevKitの下にも存在しますが、ライセンスファイルCmFirm.wbcをコピーするのは、¥Runtimeフォルダの下のbinフォルダです。ファイルの保存場所を間違えるとプロテクト作業ができませんのでご注意ください。



Chapter 3

実行形式プログラムにプロテクトをかける

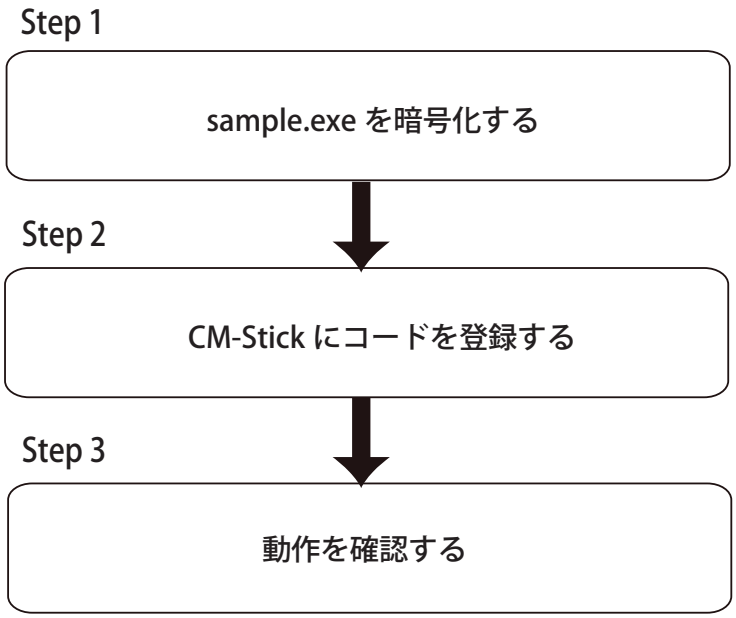
3-1. 実行形式プログラムにプロテクトをかける

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/v420/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

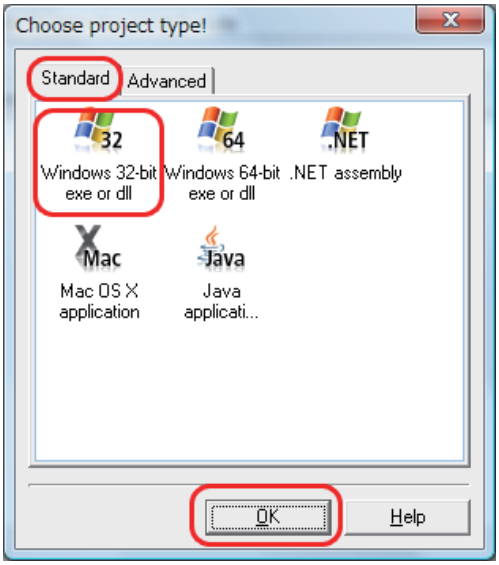
sample.exeを暗号化（プロテクト処理）します。プログラムを暗号化するには、自動暗号化ツール「AxProtector」を使用します。この「AxProtector」は、プログラムファイル全体を強力に暗号化し、デバッガからの解析を非常に困難にします。ソースコードを変更する必要がないのでとても便利です。

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(CM-FSB) を装着する

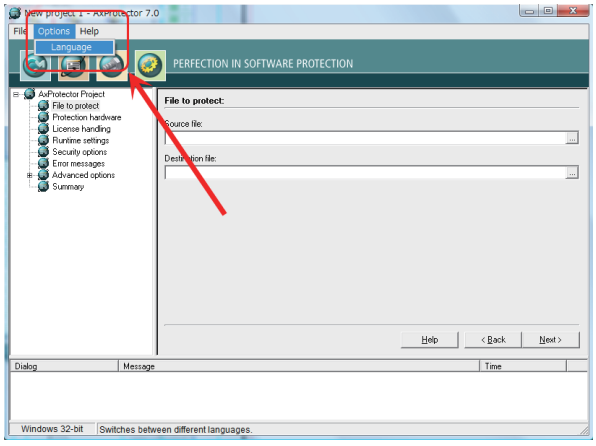
AxProtectorを使ってファイルを暗号化するには、貴社のコードメータFSB (CM-FSB)が必要です。まず、コードメータFSBをPCのUSBポートに装着し、【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】から「AxProtector」を起動し「Windows 32-bit exe or dll」を選択し、OKボタンをクリックします。

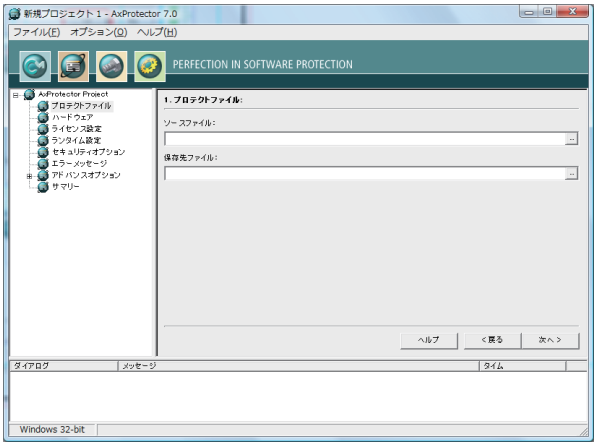
"AxProtectorGui.exe"は、インストール先の
¥Program Files¥WIBU-SYSTEMS¥AxProtector
¥DevKit¥binフォルダの中にあります。



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。

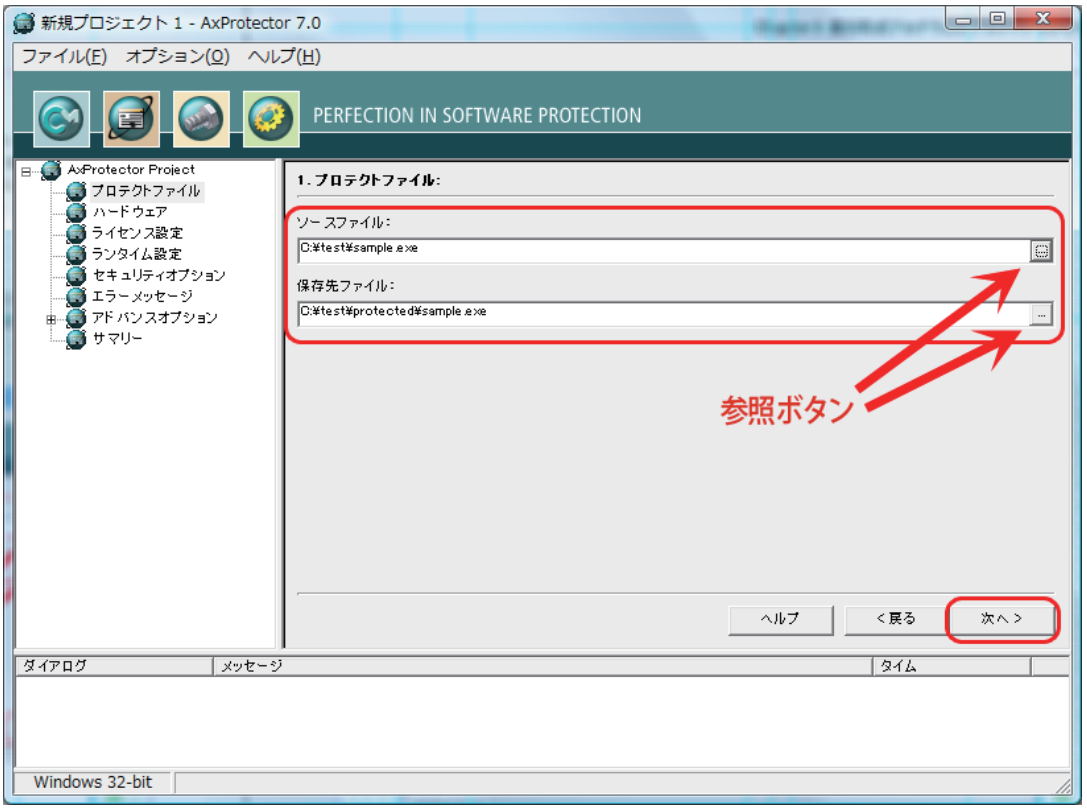




1. プロテクトファイル：

「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。

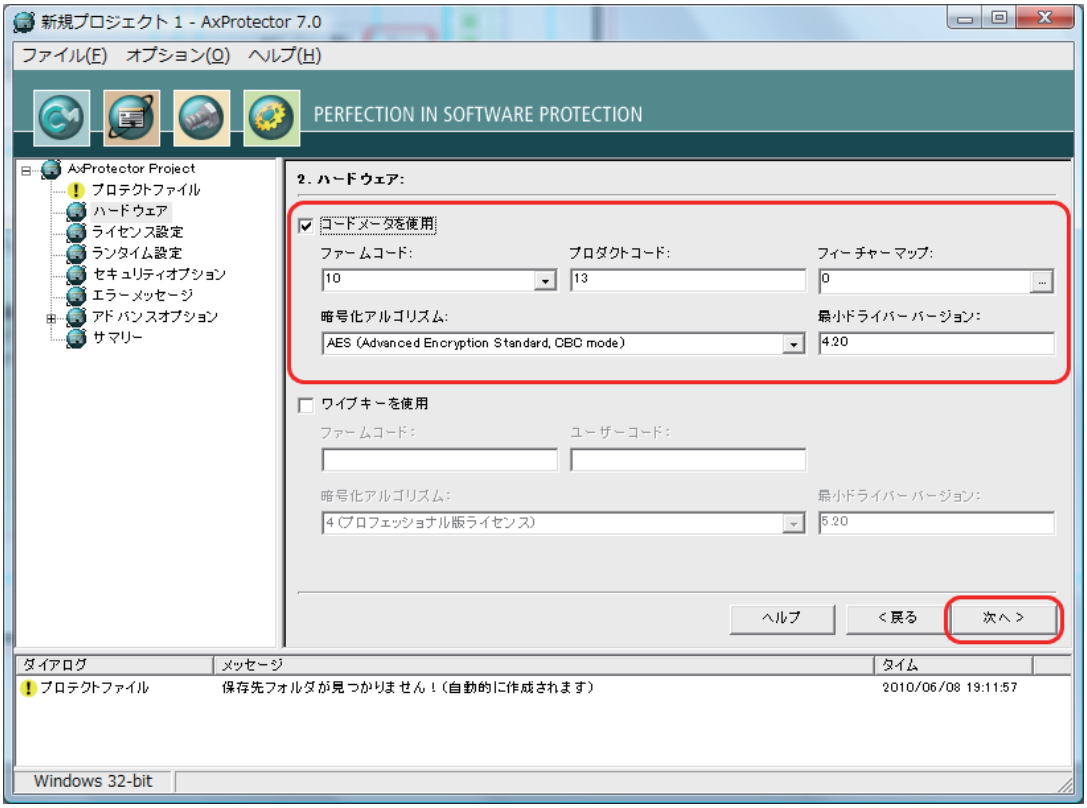
［NOTE］
オリジナルファイルを上書きするのを防ぐために、必ず保存先ファイル名はソースファイル名と別名にするか、または異なるパス名（異なるフォルダ）にしてください。ソースファイルを指定すると、同一フォルダにprotectedフォルダが自動的に作成されます。



2. ハードウェア：

「2. ハードウェア」画面で、暗号化されたプログラムが起動時に必要とする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コードメータCM-Stick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、コードメータとワイブキー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。それ以外の、フィーチャーマップ、暗号化アルゴリズム、最小ドライバーバージョンはデフォルトのままにします。



[NOTE]
コードメータとワイブキーを両方選択してプログラムを暗号化すると、コードメータまたは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。

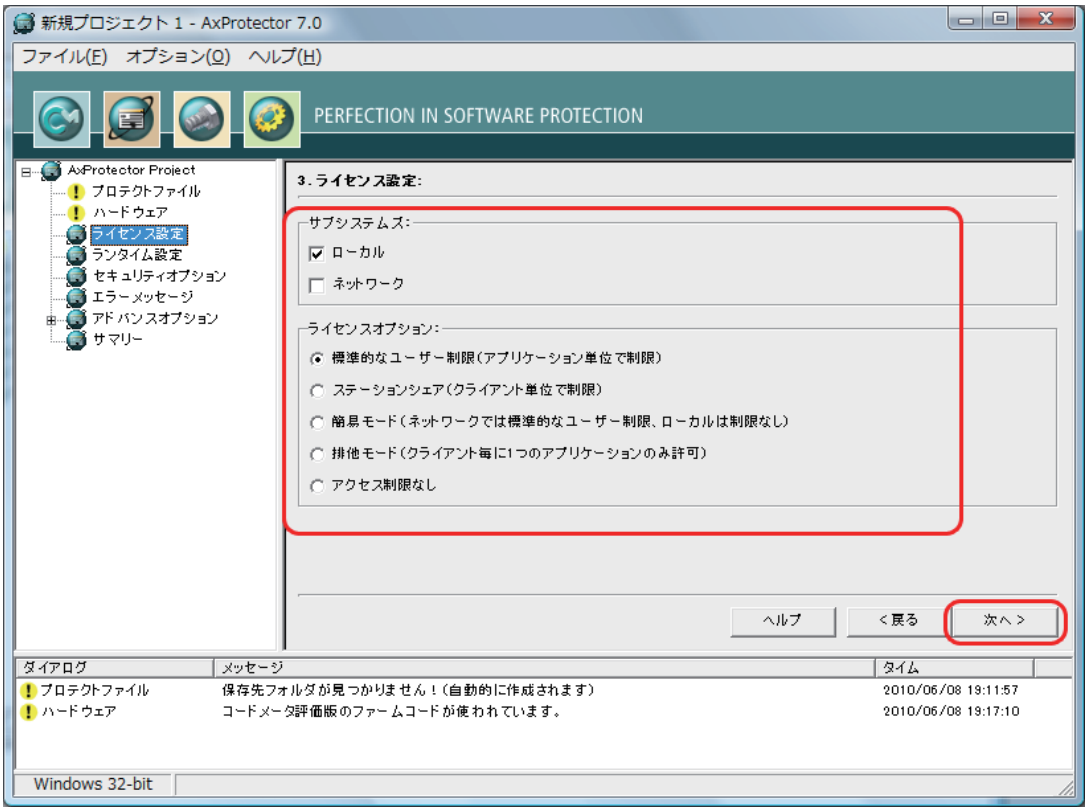
[NOTE]
ファームコードは貴社専用の6桁コード（整数値）になり、コードメータ開発キット(CM-FSB)をご購入した時点で割り当てられます。プロダクトコードは、0～4294967295(32bits)の範囲の整数値が有効で、貴社にて自由に割り当てることができます。

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）のコードメータサーバーにアクセスします。指定したコードメータキーが見つかると暗号化されたプログラムは起動します。この"ネットワーク"オプションは、コードメータのネットワーク機能を使ってライセンス管理をするときに役に立ちます。

"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。



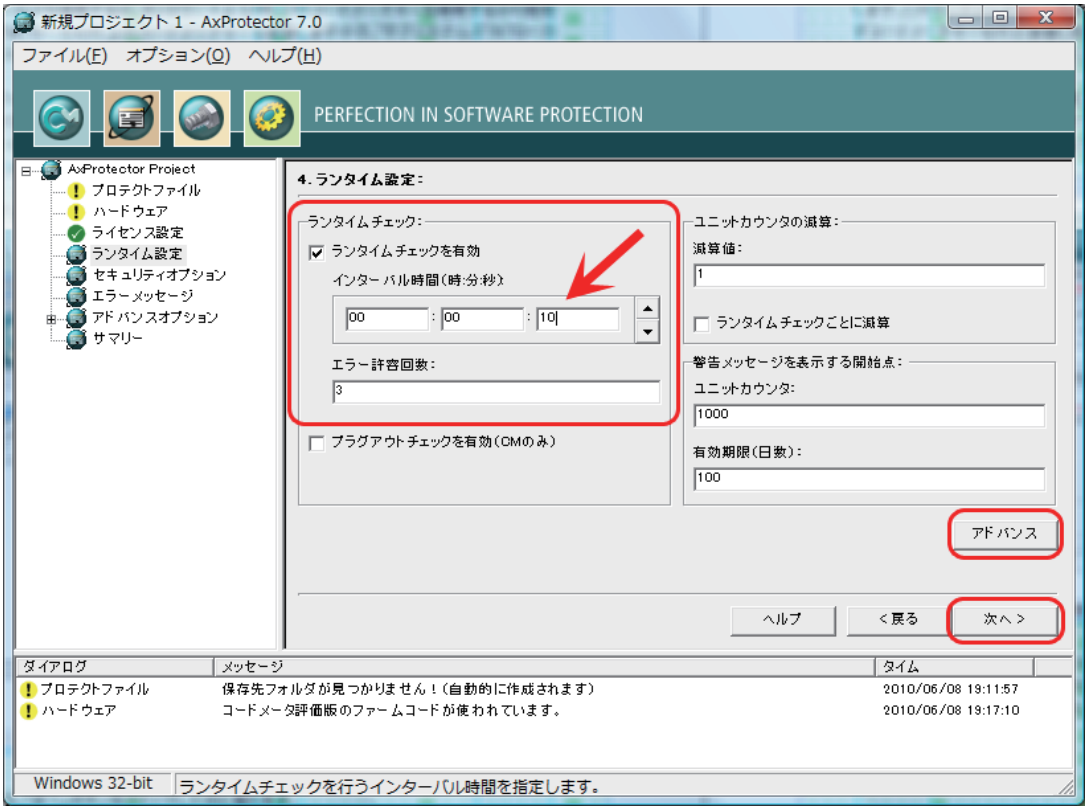
設定が終了したら、「次へ」ボタンをクリックして次に進みます。

[NOTE]
ライセンスオプションの詳細につきましては、「Chapter 5 自動暗号化ツール AxProtectorについて/ 5-4 . 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。

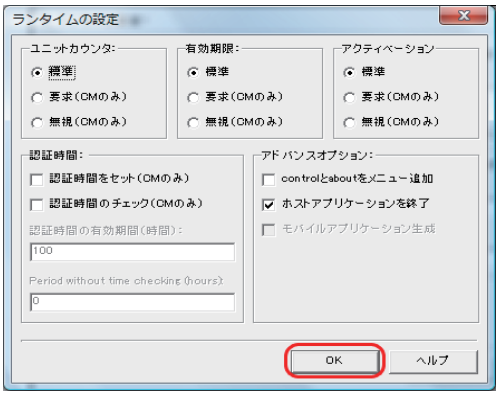
4. ランタイム設定：

「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により、プロテクトされたプログラムが動作するに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。



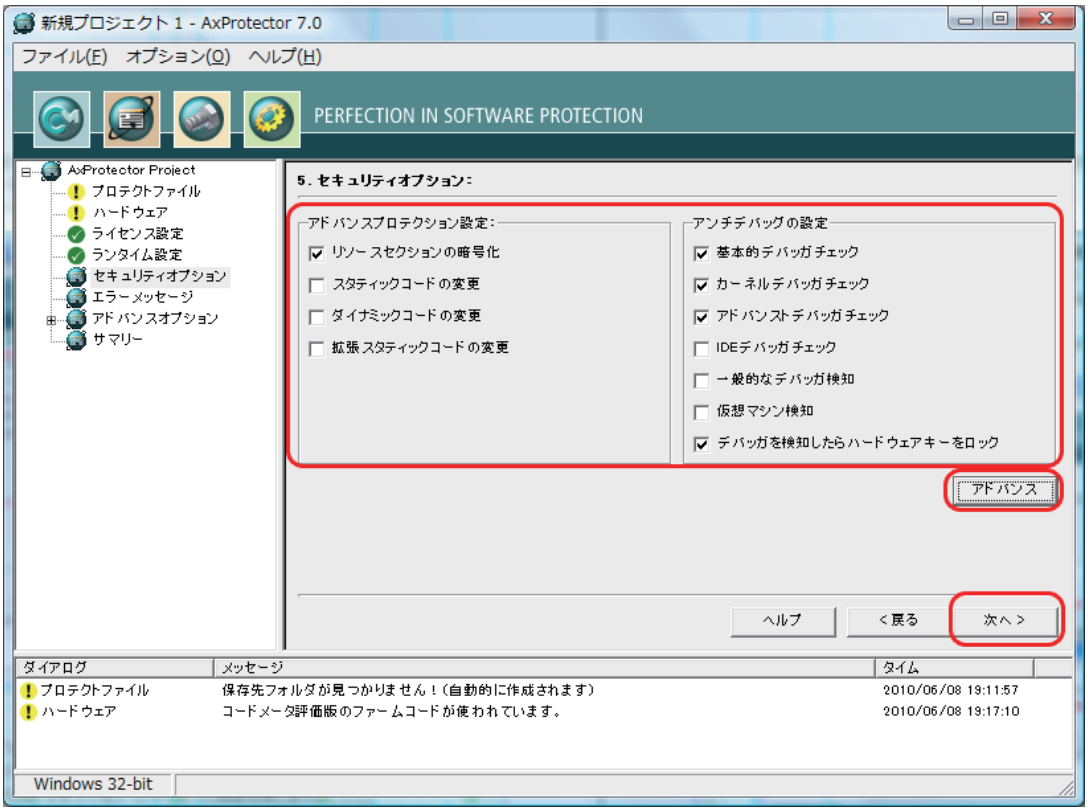
また、「アドバンス」ボタンをクリックすると、さらにきめの細かいランタイム設定を行うことができます。ここでは、特に設定しませんので、「OK」ボタンをクリックして閉じます。



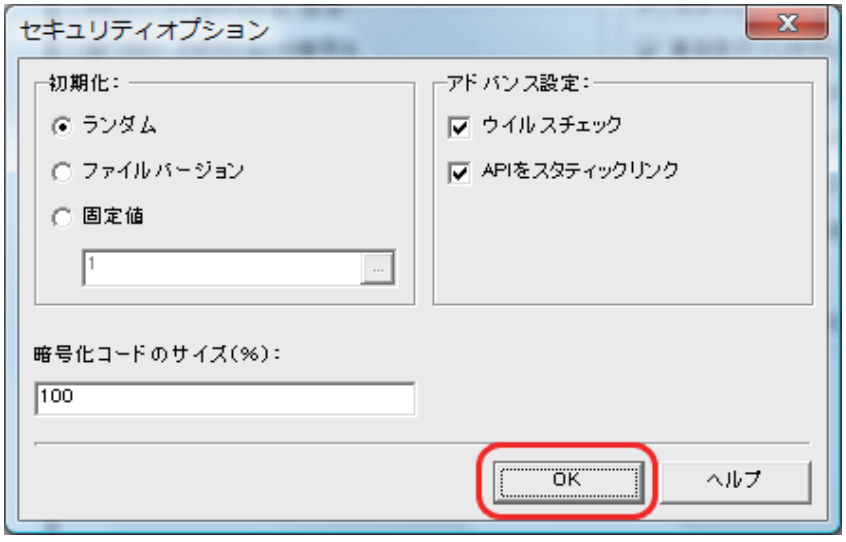
設定が終了したら、「次へ」ボタンをクリックして次に進みます。

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、デバッガ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、中央部右側の"アドバンス"ボタンをクリックすると、ランダムデータ生成やウイルスチェック機能追加、暗号化するサイズ（%指定）などのアドバンス機能を追加できます。ここでは、デフォルトの状態のままOKボタンをクリックし、"セキュリティオプション"（アドバンス）画面を閉じます。



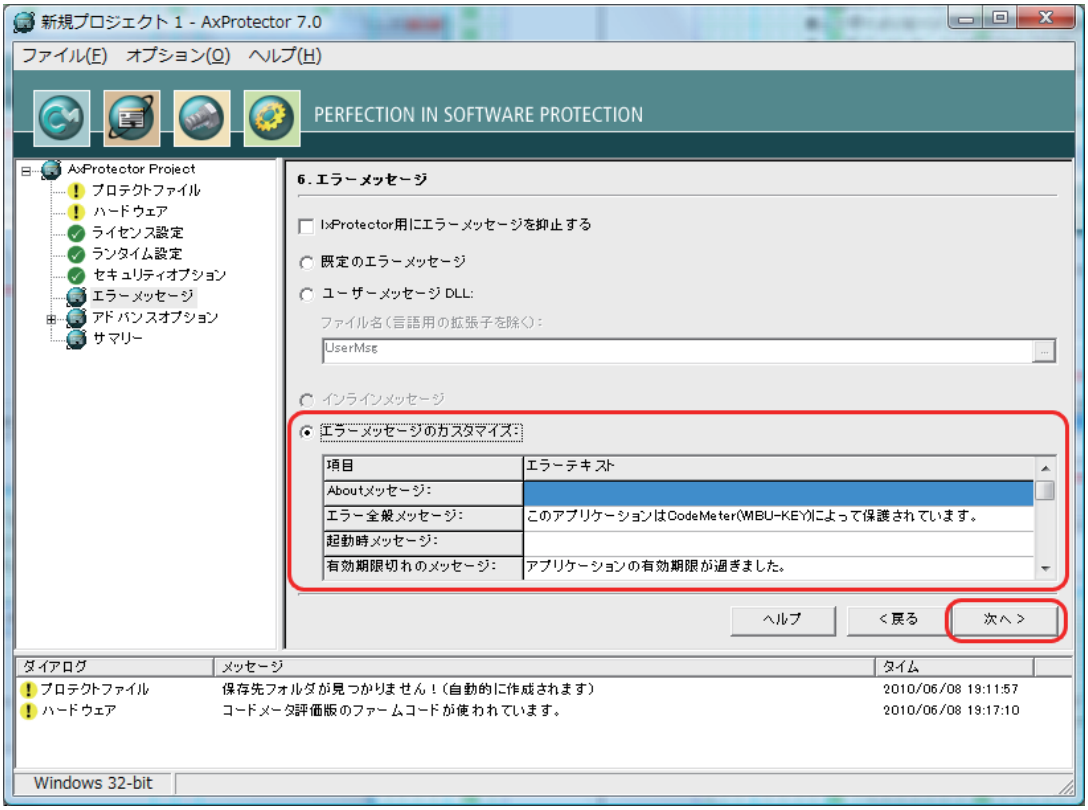
「5. セキュリティオプション」画面に戻り、「次へ」をクリックして次に進みます。

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。



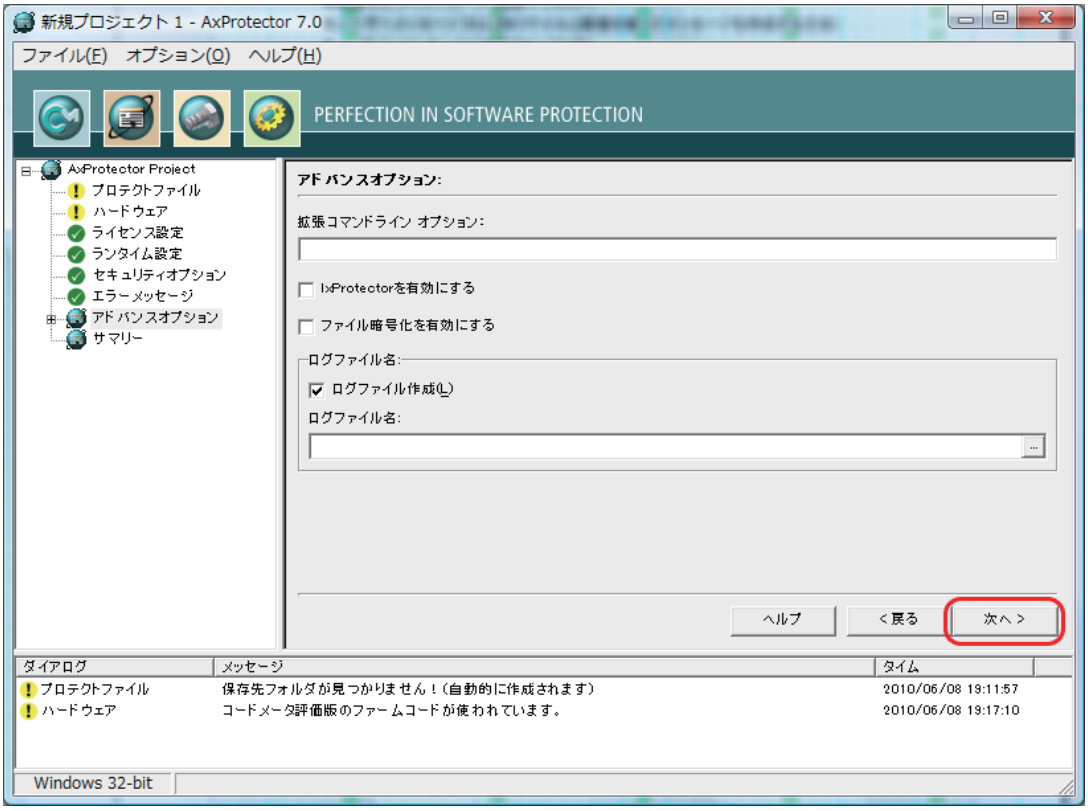
「次へ」をクリックして次に進みます。

[NOTE]
エラーメッセージの詳細につきましては、「Chapter 5 自動暗号化ツール AxProtectorについて/ 5-4 . 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：

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。

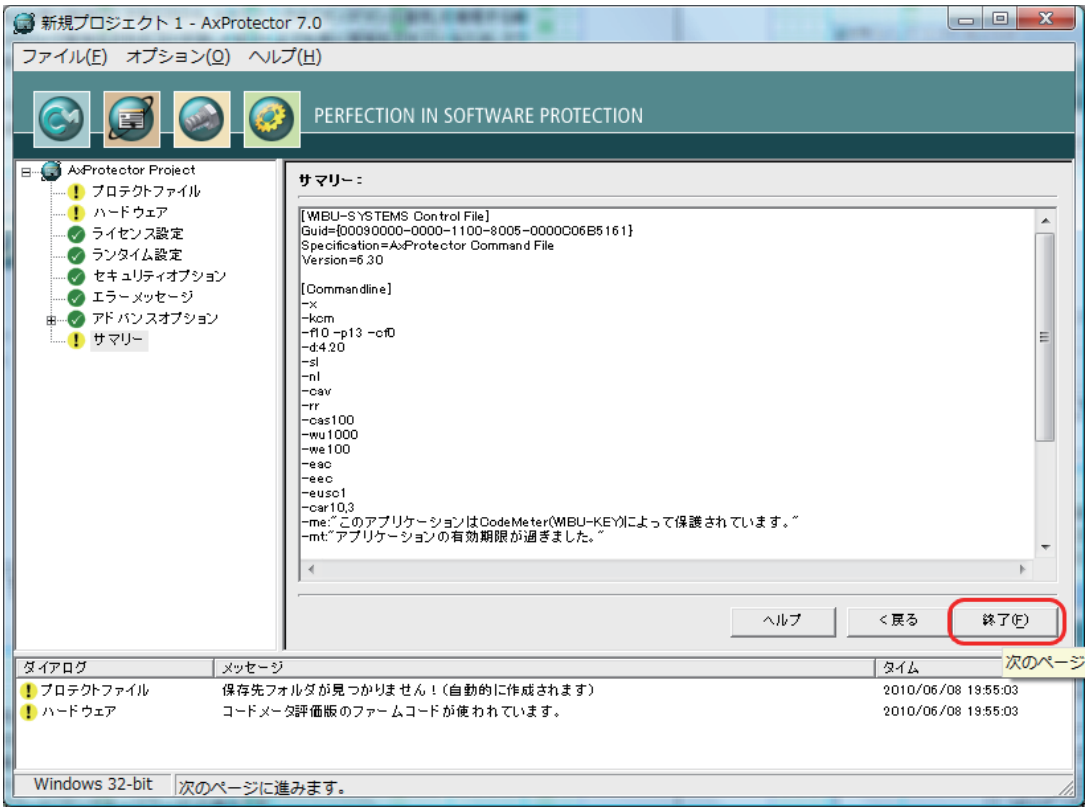
IxProtectorを使用するには、ソースコードにWUPI (Wibu Universal Protection Interface)ファンクションを組み込む必要があります。ここでは、IxProtectorは使用しませんのでチェックをはずしておきます。



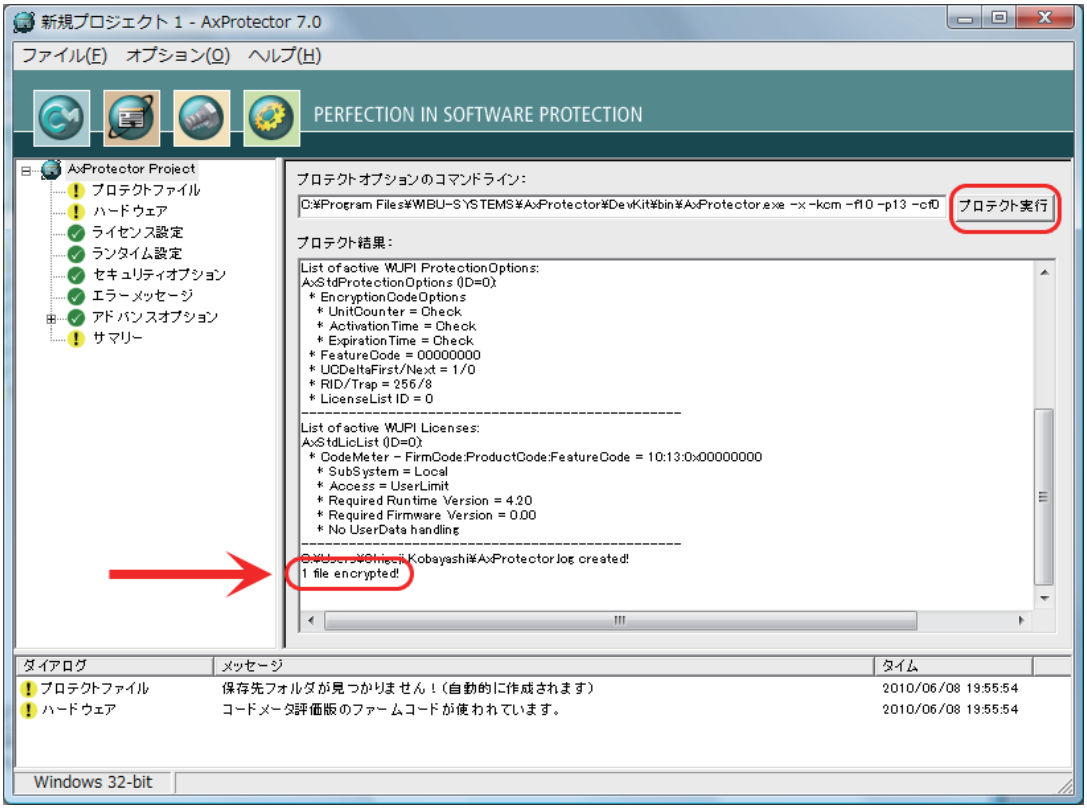
[NOTE]
C#やVB.NETなどで作成したMicrosoft .NET Framework対応プログラム（マネージコード）の場合は、IxProtector/WUPIを使用しなくても、AxProtector (.NETアセンブリ)で暗号化することで自動的に「オンデマンド復号」機能が組み込まれます。「オンデマンド復号」を実現するために、あえてソースコードを修正する必要がありません。IxProtector/WUPIをソースコードに組み込む必要があるのは、VC++やVB6などのネイティブコード（アンマネージコード）の場合です。

8. サマリー（プロテクト内容を確認する）：

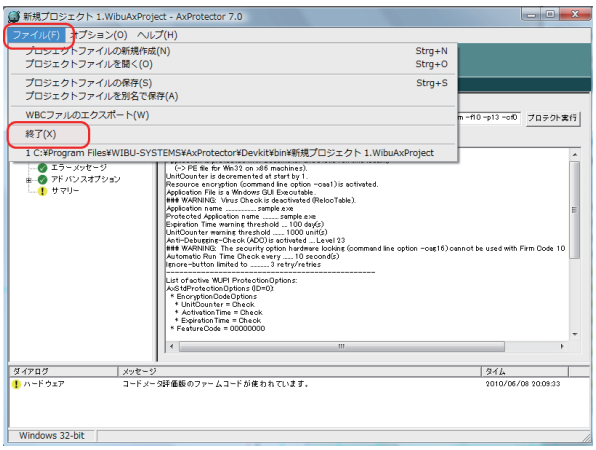
「サマリー」画面に、いままでに設定したプロテクト内容が表示されます。



"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(CM-FSB)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。）

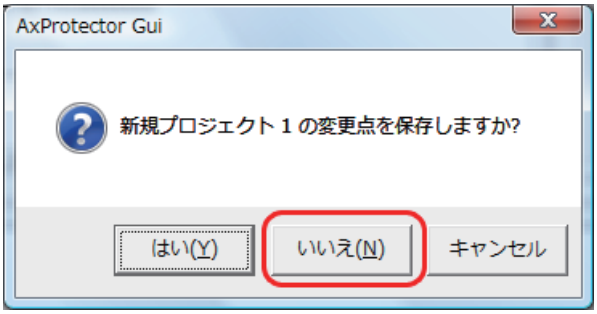


これで、sample.exeの暗号化処理は終了です。メニューバーの「ファイル」-「終了」、または画面右上の☒ボタンでAxProtectorを終了します。



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

¥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. CM-Stick 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CM-Stickに登録します。CM-Stickにコードを登録するには、5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. コードメータプロデューサ(CodeMeter Producer)を使用する方法
3. CmBoxPgmプログラム(コマンドライン環境)を使用する方法
4. 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
5. 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を使って、CM-Stickにコードを登録します。

1. 用意するもの

CM-Stick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 (CM-FSB)
■ CM-Stick

2. コードメータ ライセンスエディタを起動する

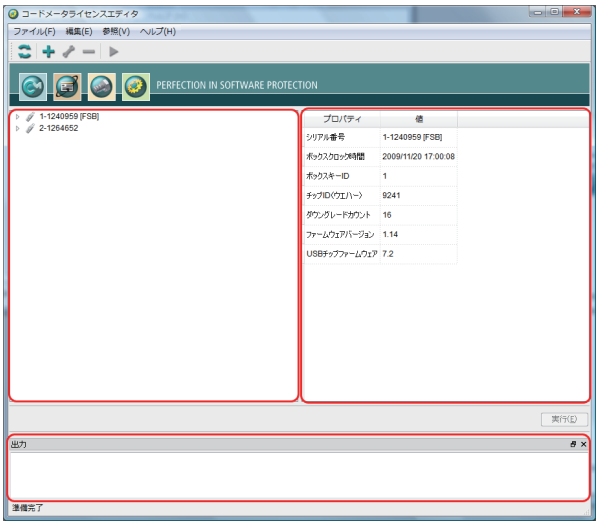
①コードメータ FSB と CM-Stick を装着する

まず、コードメータFSBとコードを登録するCM-Stick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

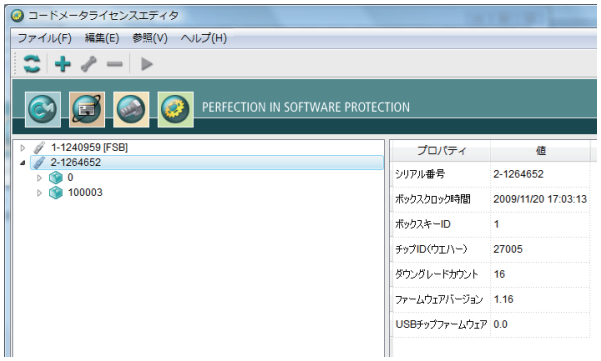
スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

左ペインには、CM-Stick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。右ペインには、左ペインで選択されたCM-Stickのプロパティが表示されます。また、下部の出力ペインには、「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 CM-Stick の▷マークをクリックする

CM-Stick（ここでは2-1264652）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。

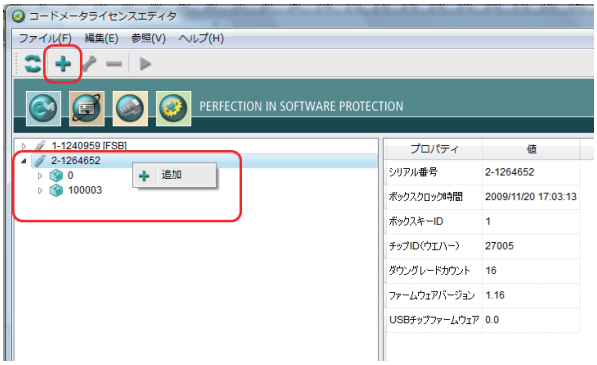


正規版CM-Stickをお持ちの方は、すでに貴社のファームコードが登録されています。その場合は、次の④と⑤の作業は不要ですので、⑥に進んでください。

評価版CM-Stick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

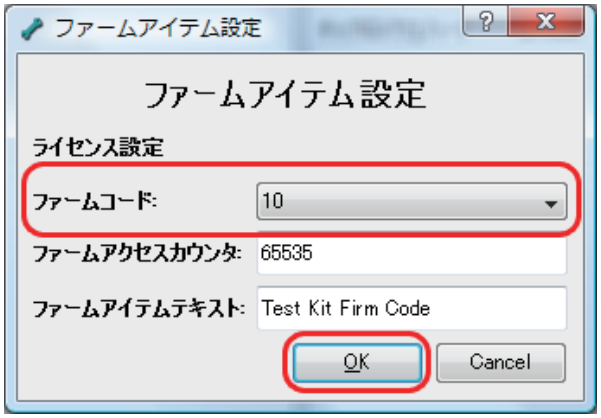
CM-Stick（ここでは2-1264652）を選択し、右マウスをクリックすると「追加」ボタンが表示されます。その「追加」ボタンをクリックすると、ファームアイテム設定画面が表示されます。



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

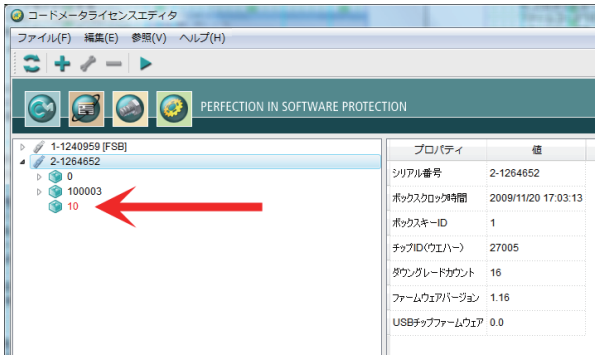
ファームアイテム設定画面で、ファームコードの入力欄にファームコード=10を選択し、「OK」ボタンをクリックします。



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。

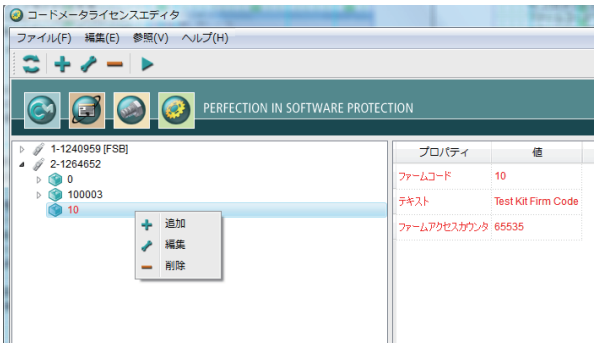
ファームアイテムテキストは、ファームコードに対するコメント文です。半角英数文字で80文字まで使用可能です。

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CM-Stick（ここでは2-1264652)に"10"が赤色で表示されます。



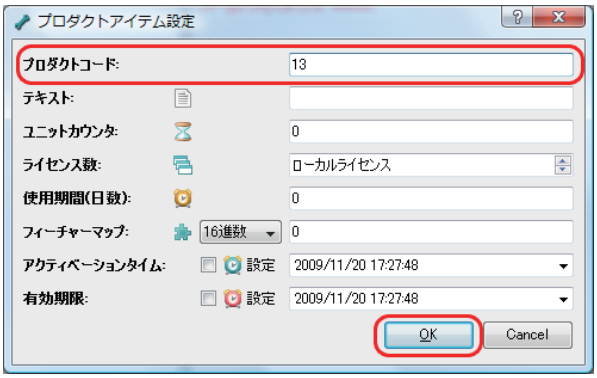
⑥ プロダクトコードを入力する

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。

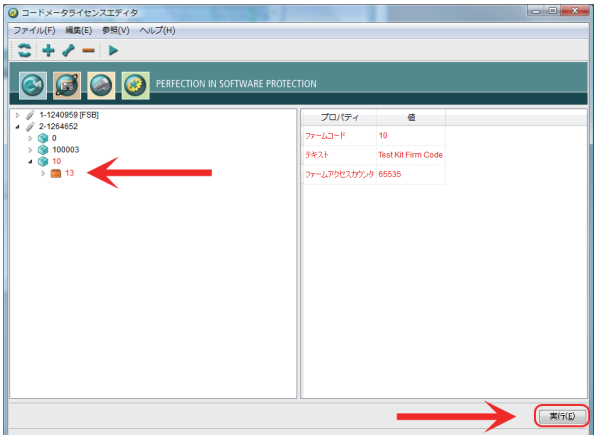
このプロダクトアイテム設定画面では、プロダクトコード設定だけでなく、プロダクトコードに対してユニットカウンタ、ライセンス数、使用期間（日数）、フィーチャーマップ、アクティベーションタイム（使用開始日）、有効期限などのオプション項目も設定することができます。



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 CM-Stick に登録する

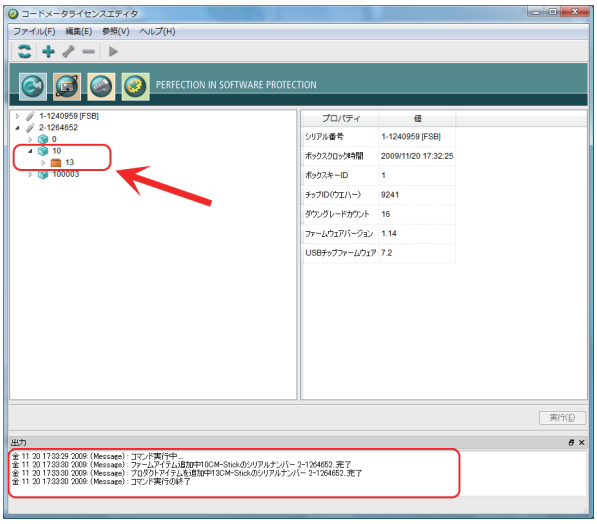
コードメータライセンスエディタの左ペインのCM-Stick 2-1264652にファームコード=10、プロダクトコード=13が赤色で表示されています。プロダクトコード=13が表示されていない場合は、ファームコード=10の左部の▷マークをクリックしてください。



右下の「実行」ボタンをクリックすると、CM-Stickにファームコード=10、プロダクトコード=13が書き込まれます。

⑧ 登録されたことを確認する

CM-Stick 2-1264652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。

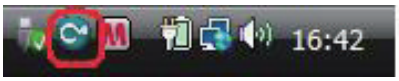


下部の出力ペインには、実行結果のステータスが表示されます。

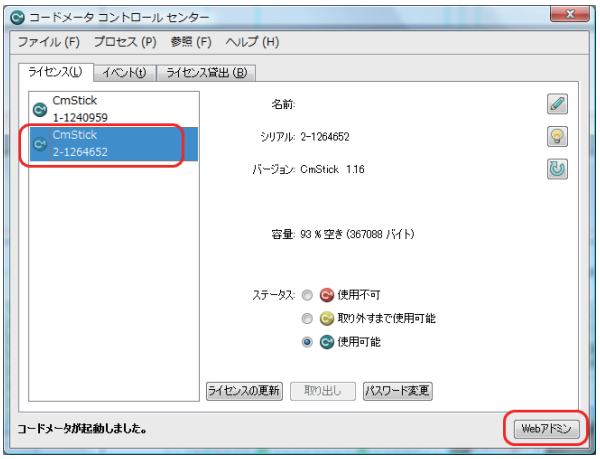
3. Web アドミン (WebAdmin) 上からも確認する

CM-Stick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、CM-Stick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、CM-Stick 2-1264652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、CM-Stick 2-1264652の情報が表示されます。

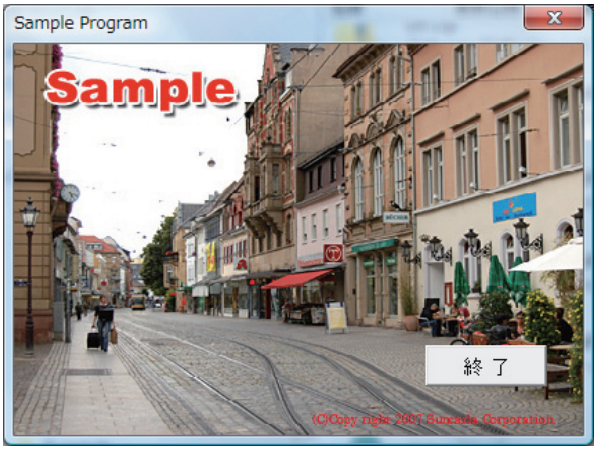
「ライセンス」メニューをクリックすると、CM-Stick 2-1264652のライセンス内容が表示されます。

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。

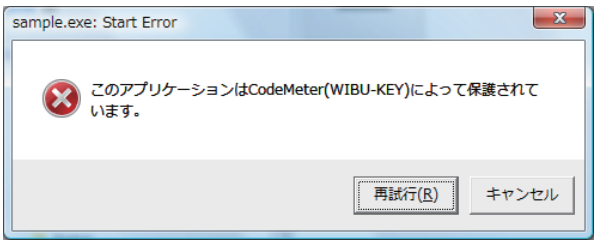
3-4. 動作を確認する

コードを登録したCM-Stick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、CM-Stick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CM-Stickが無いと暗号化されたsample.exeが起動しません。

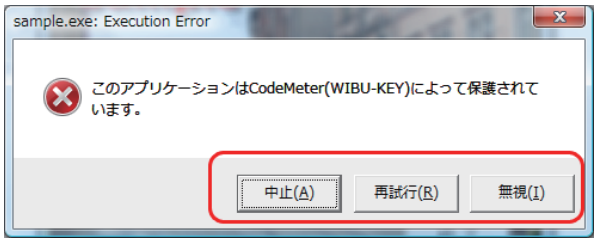


これで、プロテクト処理は完了です。

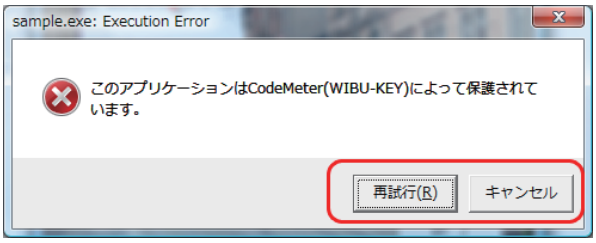
[NOTE]
エラーメッセージは、独自でカスタマイズすることができます。
詳しくは、「Chapter 5 自動暗号化ツール AxProtectorについて/ 5-4 AxProtectorの各入力画面の説明/6. エラーメッセージ」をご参照ください。

ランタイムチェックを確認する

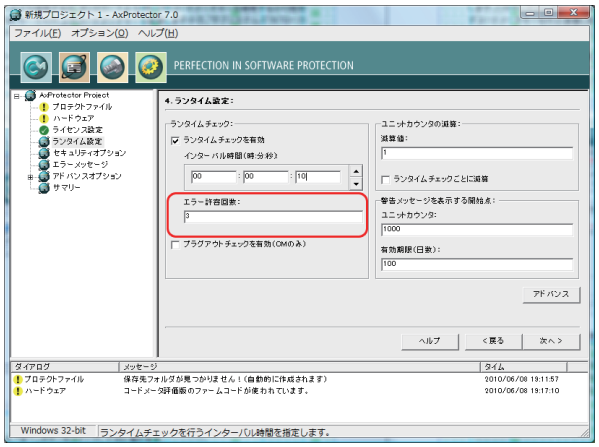
sample.exeが起動された状態で、コードメータCM-StickをPCから取り外します。ランタイムチェックのインターバル時間を10秒に設定してあるので、10秒後に右のメッセージが表示されます。



コードメータCM-Stick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CM-Stick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]
１個のコードメータCM-Stickを使って複数のPC上で同時にプログラムを使用する（１個のキーによる使い回し行為）というライセンス違反を防ぐためにも、このランタイムチェック機能のご使用をお勧めいたします。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CM-Stickにユニットカウンタを設定します。

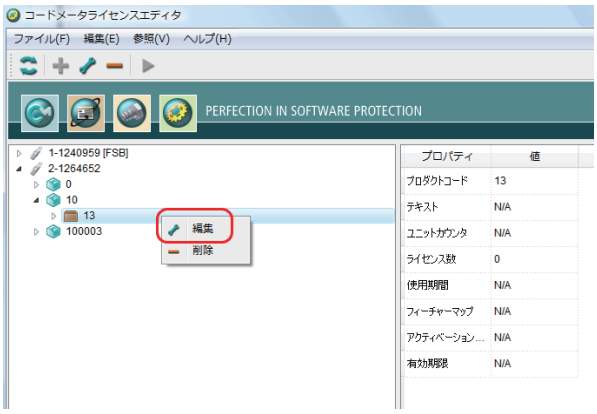
現在のCM-Stickのファームコード = 10、プロダクトコード=13には、ユニットカウンタが設定されていません。("n/a"の状態）この状態では、プログラムの使用回数は無制限になります。

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

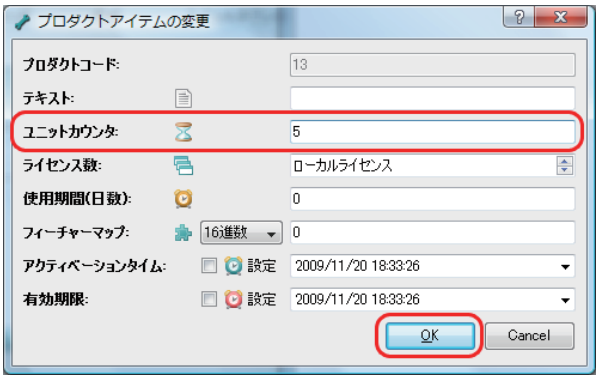
コードメータFSBとCM-StickをPCに装着して、スタートボタンから[すべてのプログラム]-[CodeMeter]-[Tools]-[CodeMeter License Editor]を選択し、コードメータライセンスエディタを起動します。

CM-Stick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。



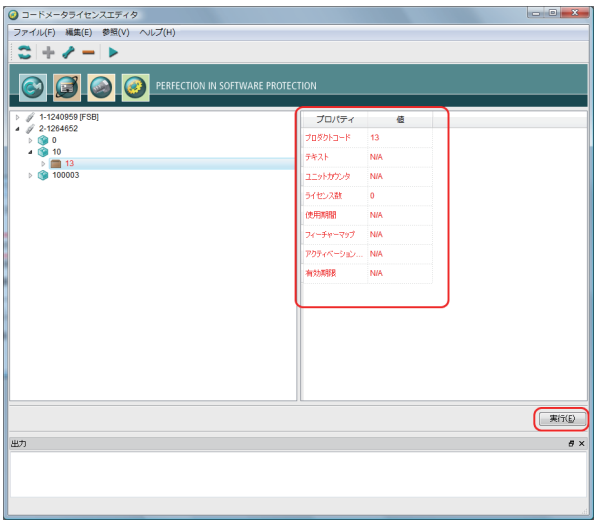
②ユニットカウンタを設定する

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。



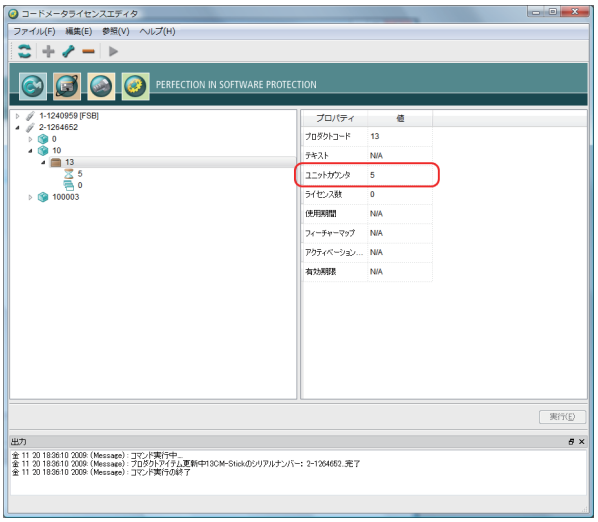
③「実行」ボタンをクリックする

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」ボタンをクリックします。



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。

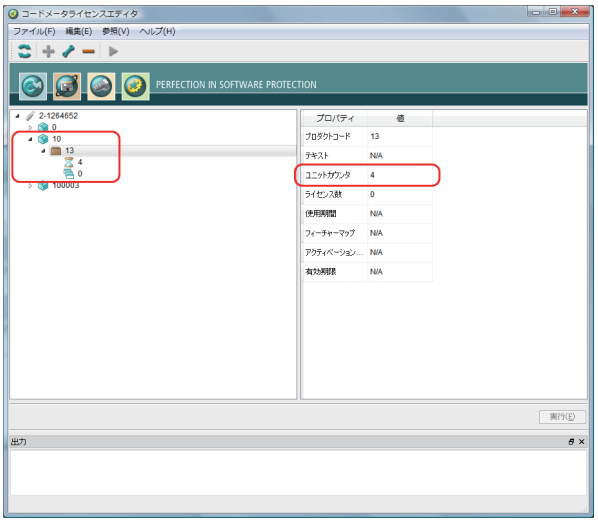
⑤ sample.exe を起動しユニットカウンタを確認

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）に利用すると非常に効果的です。



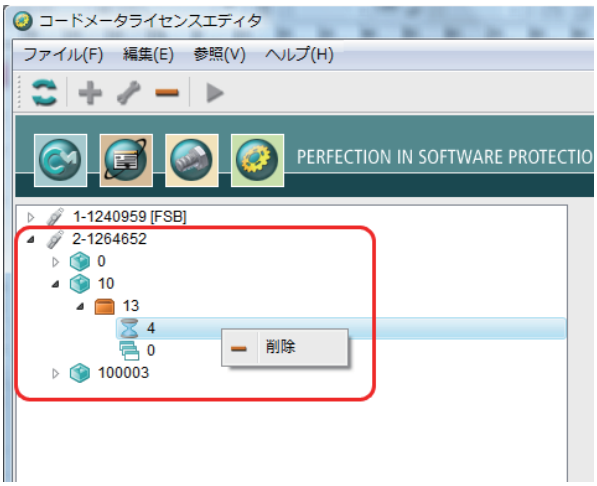
3-6. 使用有効期限 (Expiration Time) を設定したプロテクトを行う

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CM-Stick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。

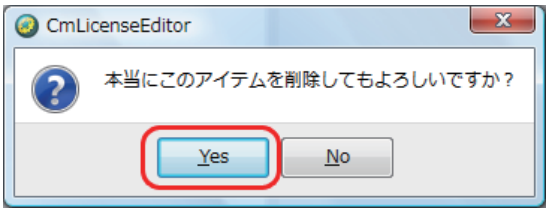
①ユニットカウンタを削除する

コードメータFSBとCM-Stick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。



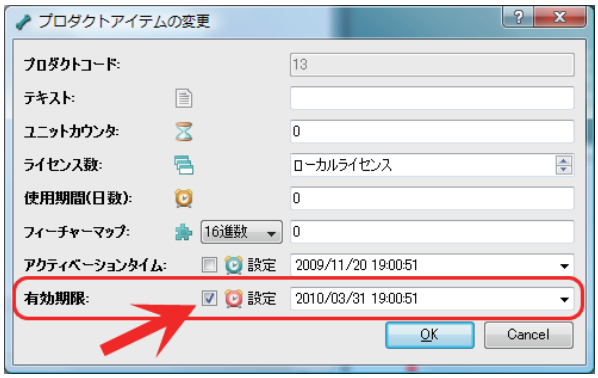
ユニットカウンタのアイコンが変わります。

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。

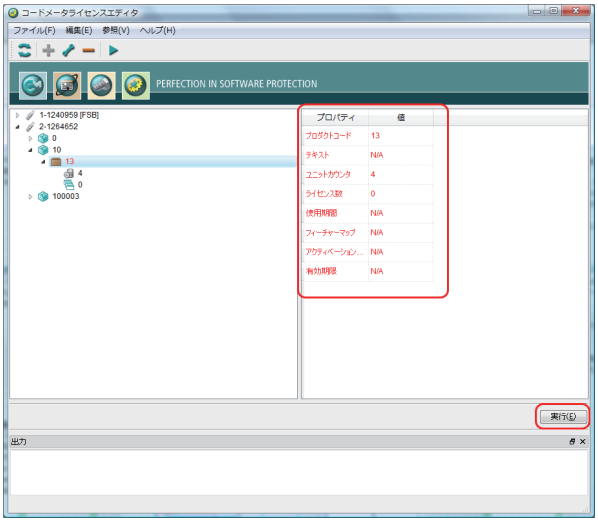
[NOTE]
"13"（プロダクトコード）を確実に選択してから右マウスをクリックしてください。

「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。



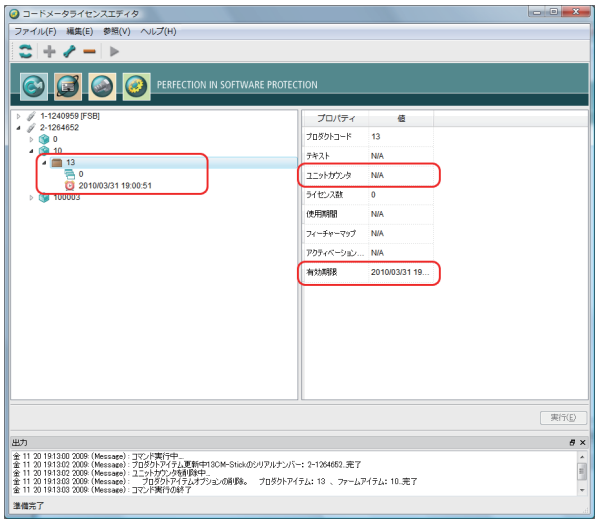
③「実行」ボタンをクリックして登録する

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、CM-Stick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2010年3月31日のCM-Stick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、CM-Stick内に刻まれたBox時間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。Box時間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。

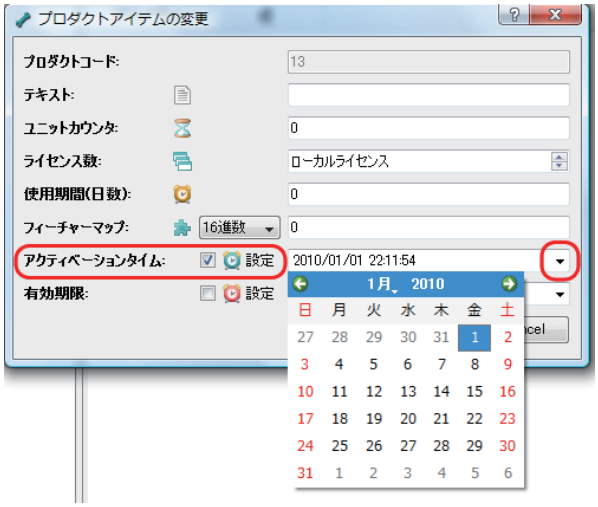
ファームコード＝10/プロダクトコード=13に対して、アクティベーションタイム（使用開始日）=2010年1月1日、使用有効期限=2010年3月31日を設定します。

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CM-Stick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。

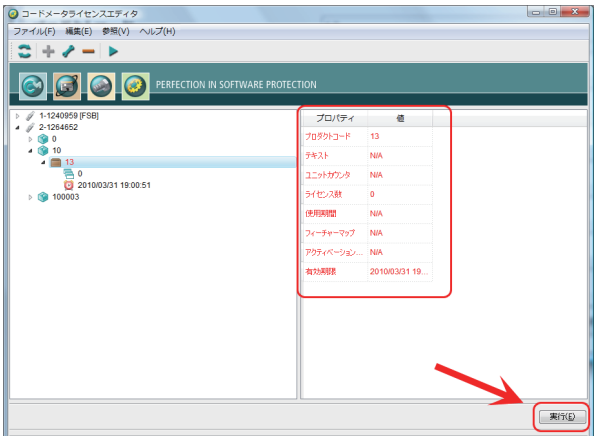
プロダクトコード"13"を選択し、右マウスをクリックし、「編集」メニューをクリックします。

「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、"2010/01/01"を設定し、「OK」ボタンをクリックします。



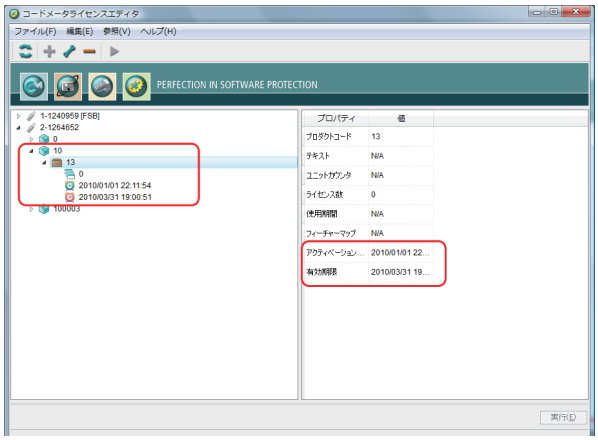
②「実行」ボタンをクリックして登録する

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、CM-Stick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。

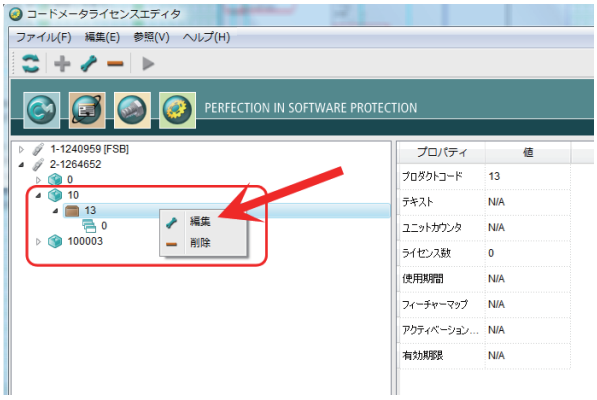
これで、使用開始日2010年1月1日、使用有効期限2010年3月31日のCM-Stickが作成できました。sample.exeは、使用開始日から使用有効期限までの期間で使用できますが、それ以外の期間では起動しません。

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。

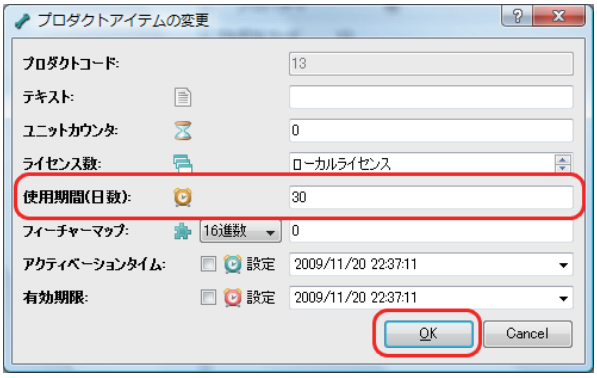
① 使用期間 (Usage Period）を設定する

コードメータFSBとCM-Stick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。



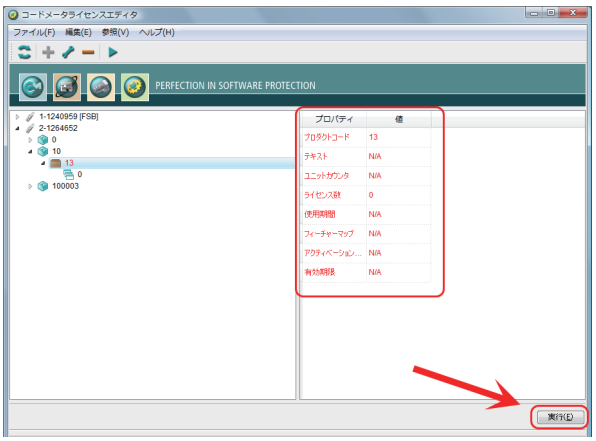
プロダクトコード"13"を選択し、右マウスをクリックし、「編集」メニューをクリックします。すでに、有効期限やアクティベーションタイムが登録されている場合は、初めに削除しておいてください。

「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。



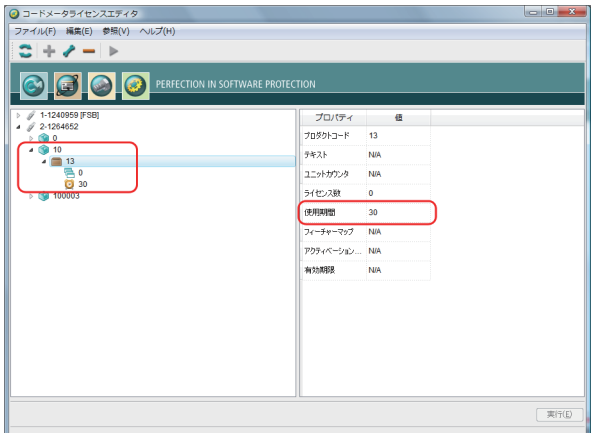
②「実行」ボタンをクリックして登録する

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、CM-Stick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。

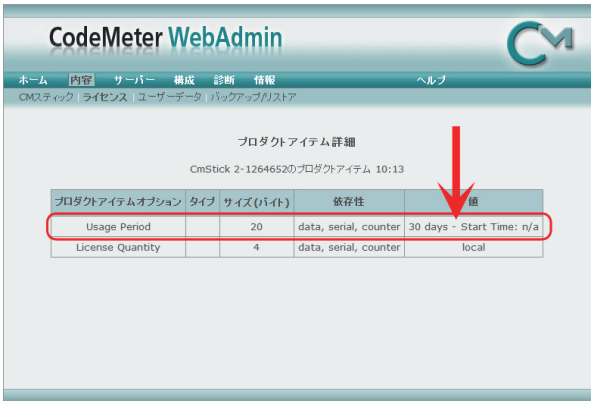


これで、「使用期間=30日間」というCM-Stickが作成されます。WebAdminから、プロダクトコード=13をクリックして確認してください。

Usage Period（使用期間）のValueの欄に

30 days - Start Time: n/a

と登録されています。



"Start Time: 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Start TimeとしてCM-Stickに自動的に登録されます。そして、その時刻(Start Time)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、Start Time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料金の課金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CM-Stickに対し、コードメータのリモートアップデート機能を使って、ファイル操作で使用期間の延長または削除を行うことも可能です。

3-9. プロテクトされたプログラムを起動する場合の注意点

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

Chapter 4

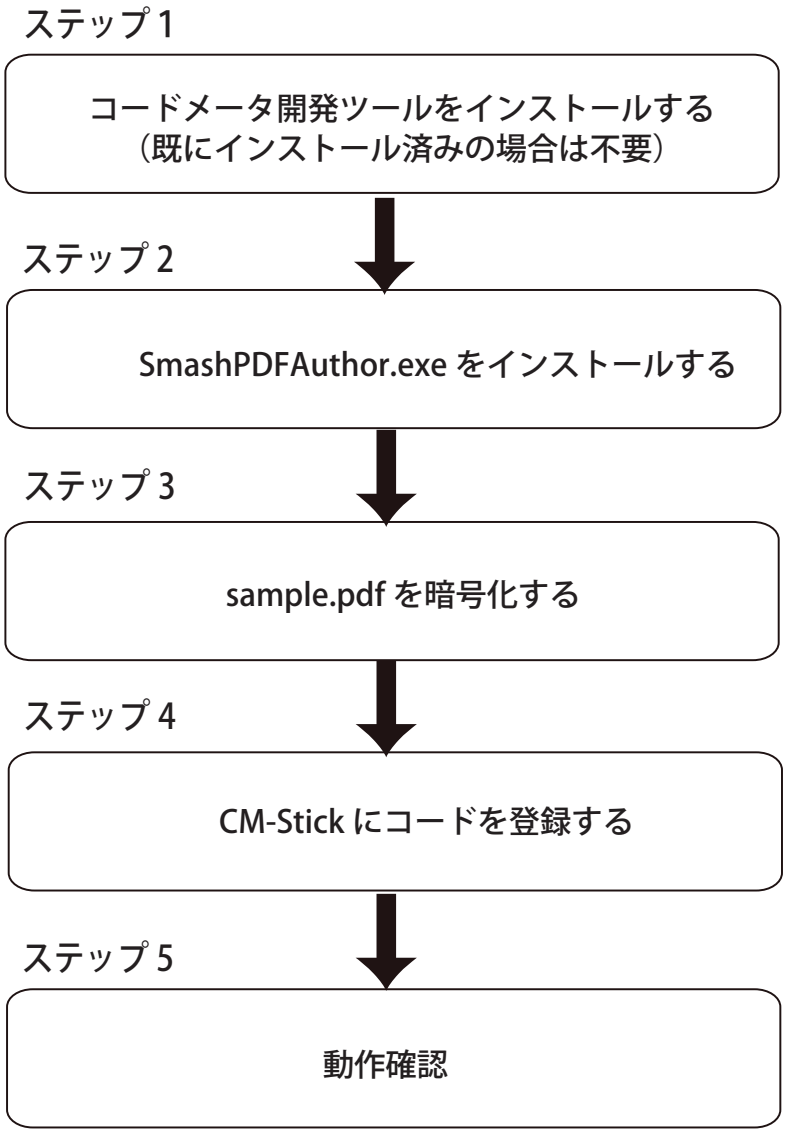
Adobe PDF ファイルにプロテクトをかける

4-1. 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。



4-2. 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 Abode Acrobat 6/7/8/9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 2000/XP/Vista/7 (32bit/64bit)上で行います。

[NOTE]
暗号化作業は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、Adobe AcrobatおよびAdobe Readerの両方で開くことができます。

4-3. 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「4-4. 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。

4-4. SmartShelter|PDFAuthor をインストールする

SmartShelter|PDFAuthorをインストールします。SmartShelter|PDF Authorは、コードメータCD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/

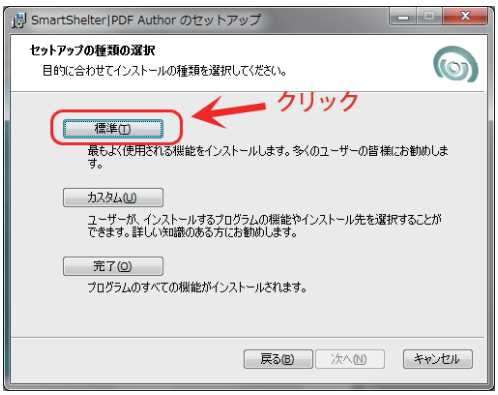
① SmashPdfAuthor.exe を起動する

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。

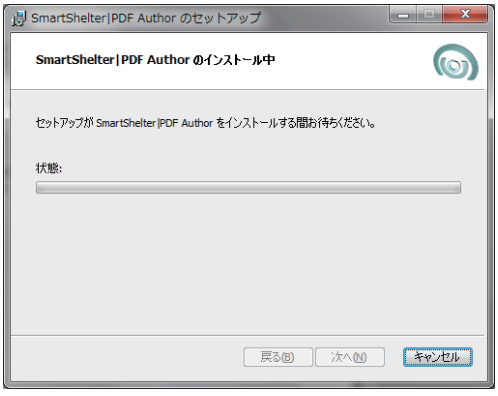


③ 「インストール」 ボタンをクリックする

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。



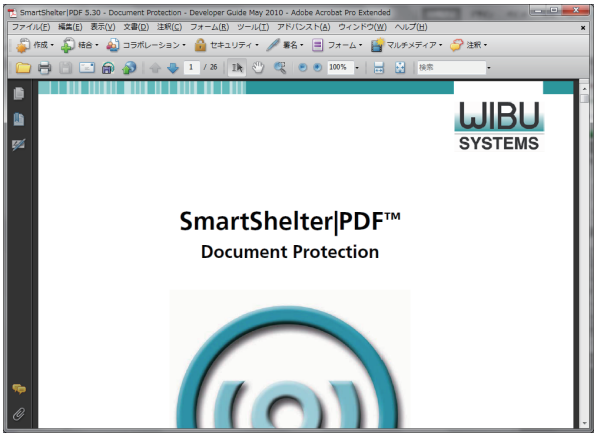
5 正常にインストールされた

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。

4-5. 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は、必ずコードメータFSBが必要になります。



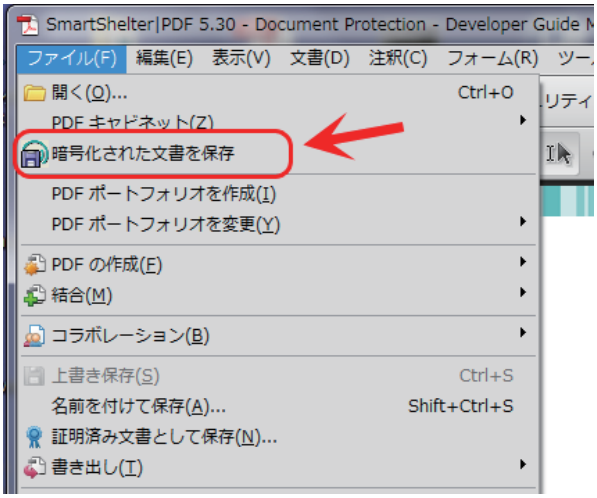
② Adobe Acrobat を起動する

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 9の画面で説明します。
Acrobat 6/7/8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

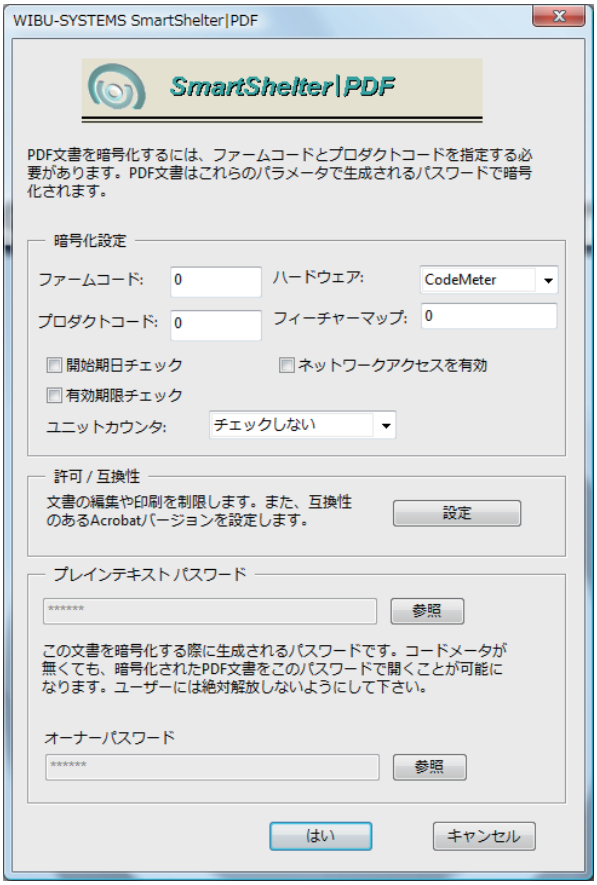
[ファイル]メニューから、「暗号化された文書を保存」メニューをクリックします。



[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とプロダクトコードをもとにハッシュ値を生成し、そのハッシュ値をベースに暗号化されます。

従い、暗号化仕様は、常に貴社独自の暗号化仕様になります。

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。

⑦ プロダクトアイテムオプションを設定する

必要に応じて、プロダクトアイテム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。

有効期限チェック

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。

ネットワークアクセスを有効

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。

ユニットカウンタ

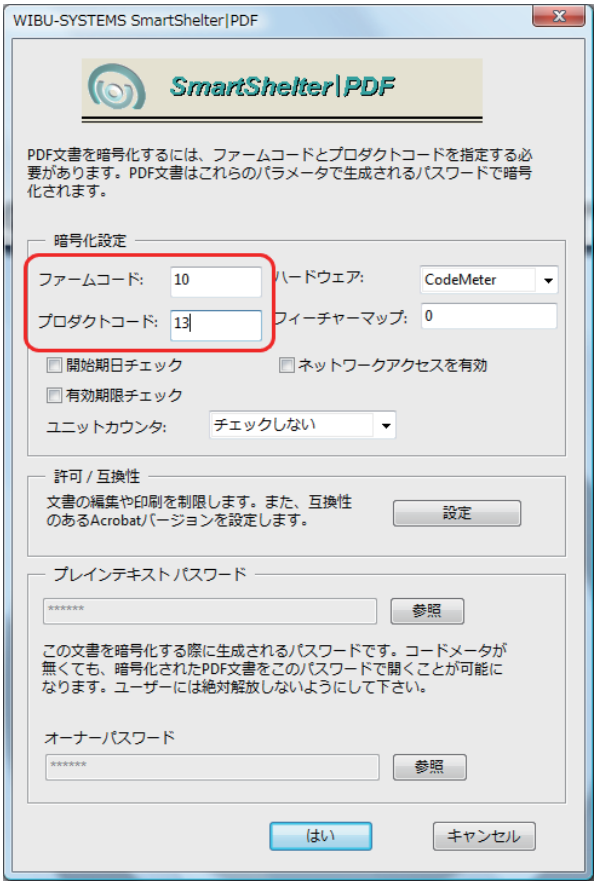
暗号化したPDFファイルの閲覧回数または印刷回数を設定します。閲覧または印刷を行うたびに、ユニットカウンタが 1 つずつ減ります。とくにユニットカウンタを使用しない場合は、"チェックしない"を選択します。

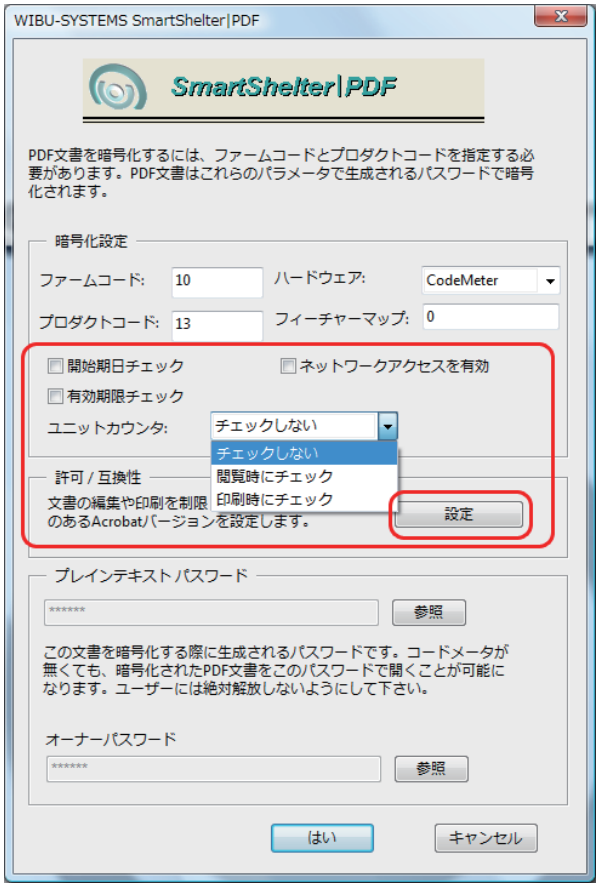
（3 つの選択モード）

- **チェックしない**
- **閲覧時にチェック**
- **印刷時にチェック**

[NOTE]

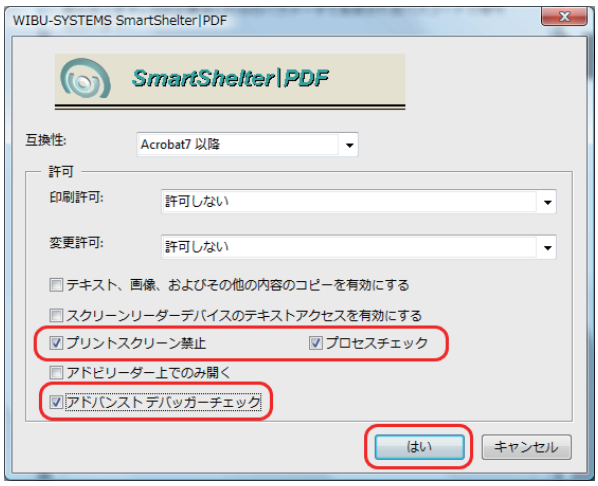
実際のファームコード、プロダクトコード、開始期日、有効期限、ユニットカウンタの数値はCM-Stickに登録します。





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、プロダクトアイテムオプションを入力したら、セキュリティオプション設定を行います。中欄の「設定」ボタンをクリックします。



⑨ オプション項目を設定する

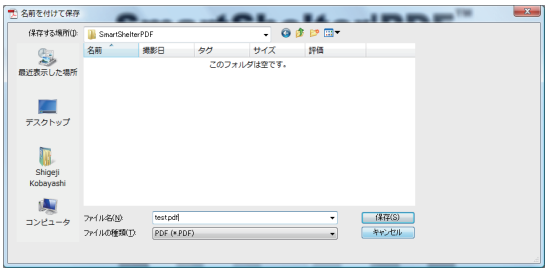
貴社のセキュリティニーズに従い、オプション項目を設定します。

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ハイレベルで防止する「アドバンスト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。

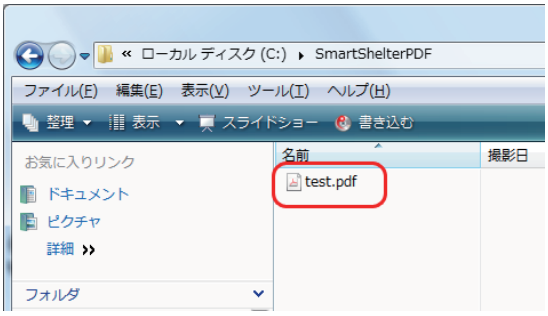
チェックを入れたら、「はい」をクリックし、暗号化設定画面に戻ります。

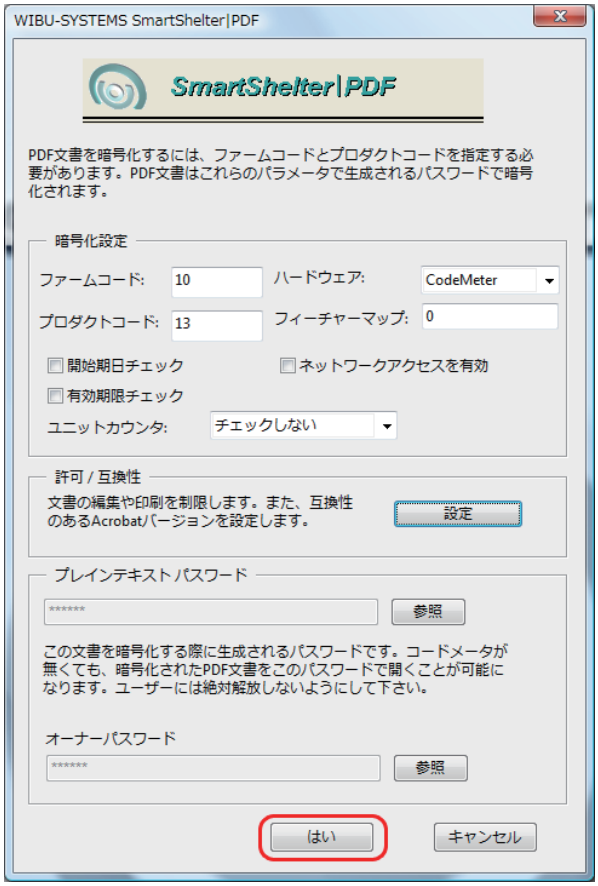
⑩ 暗号化処理を開始する

暗号化設定画面の「はい」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。

4-6. 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。

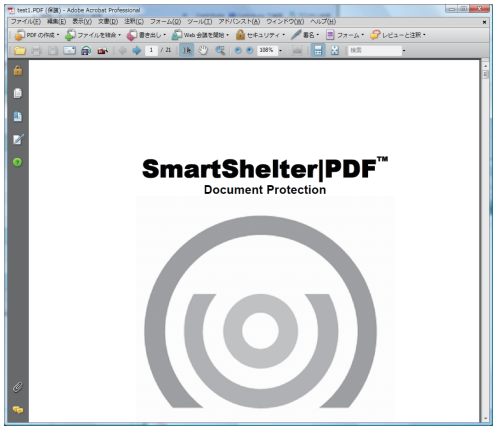
[NOTE]
ファームコードやプロダクトコードなどを登録する方法は、CM-Stick, CM-Stick/M2GB, CM-ExpressCard, CM-PCCard, CM-CFCard, CM-SDCard, CM-Micro SDCardすべて共通です。

4-7. 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。

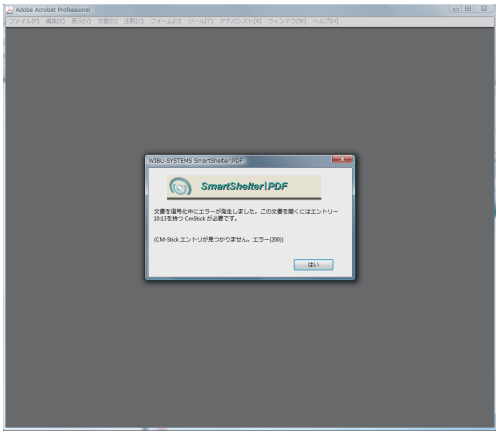
① CM-Stick が装着されていると開く

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。

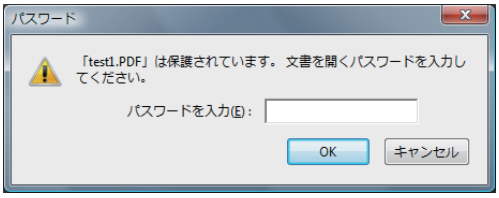


② CM-Stick がないと開かない

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。



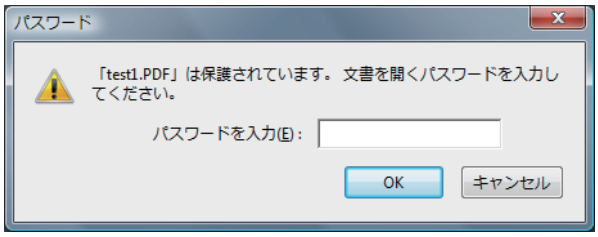
エラー画面で「はい」ボタンをクリックすると、パスワード入力画面が表示されますが、これはCM-Stickが無くても暗号化されたPDFを開くことができる「プレインテキストパスワード」または「オーナーパスワード」を入力するための画面です。通常は使用しませんので、「キャンセル」ボタンをクリックして画面を閉じます。



このパスワード画面を表示させないようにするには、「4-8. パスワード画面を表示させないようにする方法」をご参照ください。

4-8. パスワード画面を表示させないようにする方法

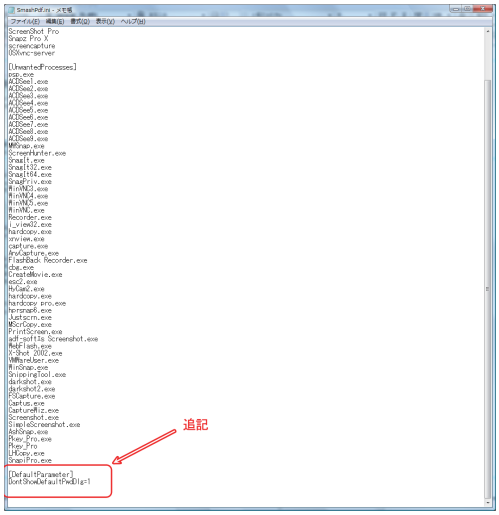
デフォルトの状態でPDFを暗号化した場合、CM-Stickが装着されていないと、最後にパスワード入力を要求する画面が表示されます。これは、CM-Stickが無くても暗号化されたPDFを開くことができる「プレインテキストパスワード」または「オーナーパスワード」を入力するための画面です。



この画面を表示させないようにするには、SmashPdf.iniファイルに下記のコマンドを追加します。

[DefaultParameter]
DontShowDefaultPwdDlg=1

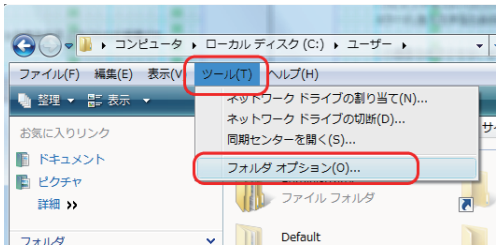
SmashPdf.iniファイルをメモ帳などで開き、上記２行を追記してください。(半角英数文字)追記後、PDFファイルを暗号化処理すると、パスワード画面が表示されないようになります。

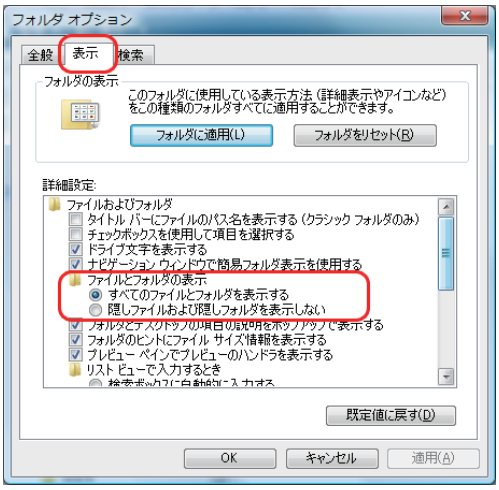


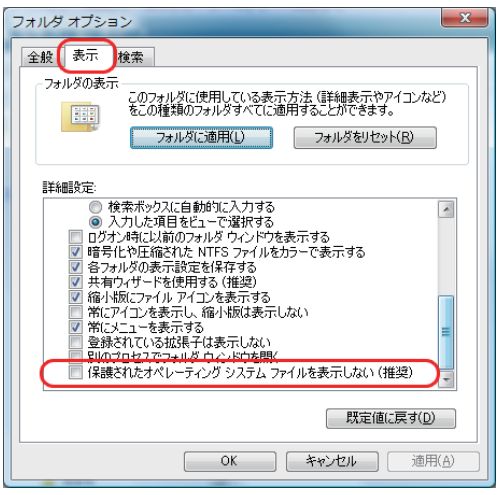
SmashPdf.iniファイルは、
Windows XPの場合
c:¥Documents and Settings¥All Users¥Application Data¥WIBU-SYSTEMS¥SmartShelter PDF

Windows Vista/7の場合
c:¥ユーザー¥All Users¥WIBU-SYSTEMS¥SmartShelter PDF

に存在します。上記フォルダが表示されない場合は、エクスプローラ上の「ツール」/「フォルダオプション」をクリックし、「表示」タブを選択し、「すべてのファイルとフォルダを表示する」にチェックを入れ、「保護されたオペレーティングシステムファイルを表示しない(推奨)」のチェックをはずしてください。また、編集権限が無いと、「アクセス拒否」によりSmashPdf.iniの編集ができませんので、その場合は、管理者権限等で編集作業を行ってください。







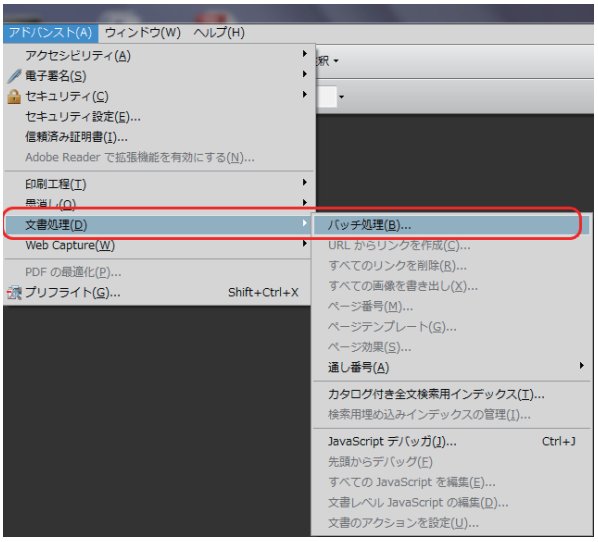
[NOTE]
パスワード画面の非表示の設定は、SmartShelterPDFランタイムキットがインストールされているAcrobatまたはAcrobat Reader上でのみ有効です。SmartShelterPDFランタイムキットがインストールされていないAcrobatまたはAcrobat Reader上で開くと、パスワード画面が表示されます。これは、Adobe社とのライセンス契約によるもので、パスワード画面を表示させる必要があるためです。

4-9. 複数の PDF ファイルを一括して暗号化する

複数のPDFファイルを一括して暗号化する場合は、Acrobatのバッチ処理機能を利用します。このバッチ処理機能を利用するには、Acrobatのプロフェッショナル版が必要です。Acrobatのスタンダード版にはバッチ処理機能がありませんので、この作業はできません。

①「バッチ処理」メニューを選択する

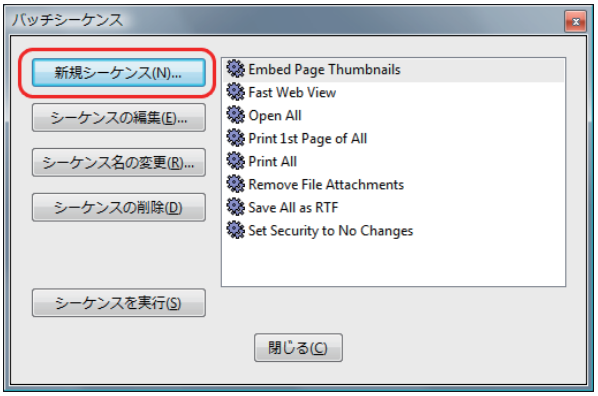
Acrobatの「アドバンスト」/「文書処理」メニューから「バッチ処理」を選択します。Acrobat6/7/8の場合、メニュー画面が多少異なりますが、「バッチ処理」を選択してください。

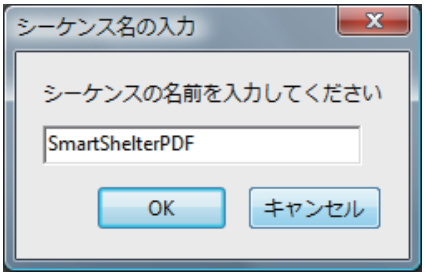


「バッチ処理」をクリックすると、バッチシーケンス画面が表示されます。このバッチシーケンスにPDF暗号化処理を登録します。

②「新規シーケンス」をクリックする

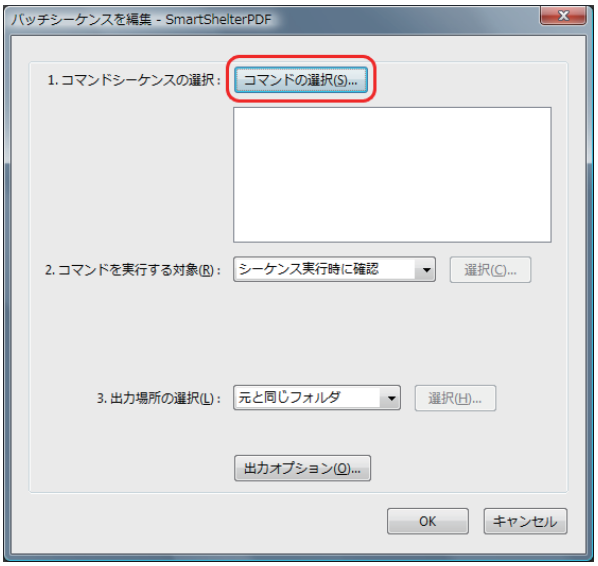
「新規シーケンス」をクリックすると、シーケンス名の入力画面が表示されますので、任意の名前を入力します。ここでは、SmartShelterPDFと入力します。





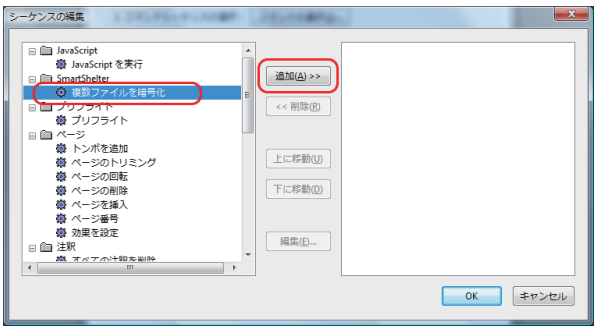
③「コマンドの選択 (S)」をクリックする

「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。



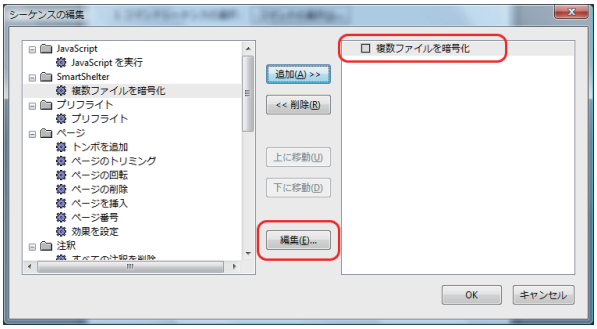
④「複数ファイルを暗号化」を選択する

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。



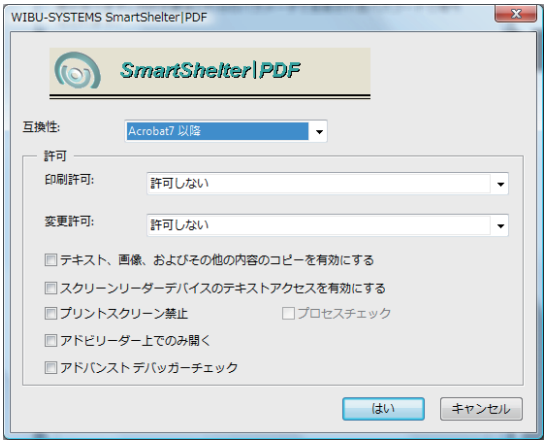
⑤「編集 (E)」ボタンをクリックする

「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。

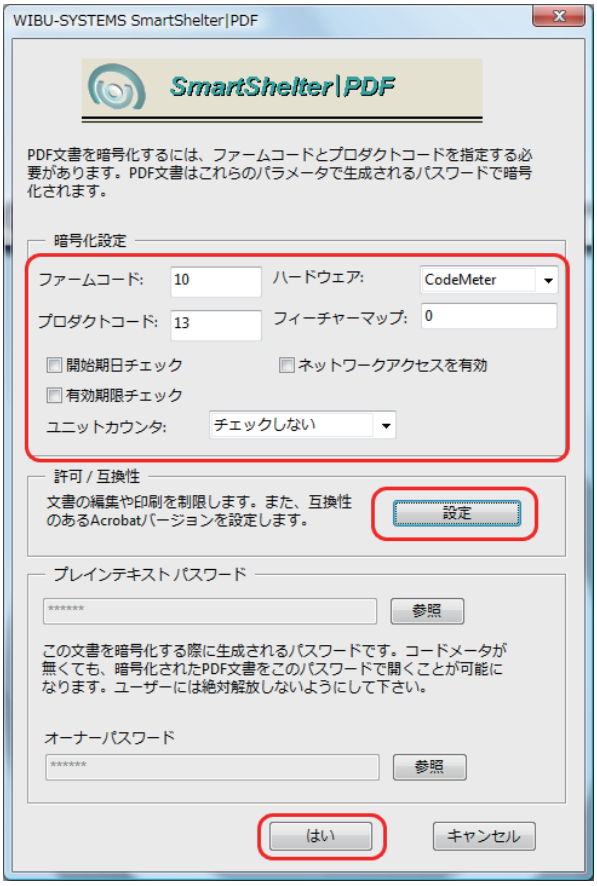


⑥ 暗号化内容を設定する

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。オプション設定ボタンをクリックし、必要に応じてオプション項目を設定します。

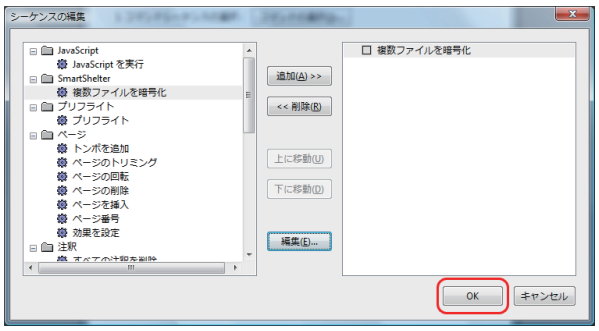


暗号化設定が完了したら、「はい」ボタンをクリックします。



⑦「OK」ボタンをクリックする

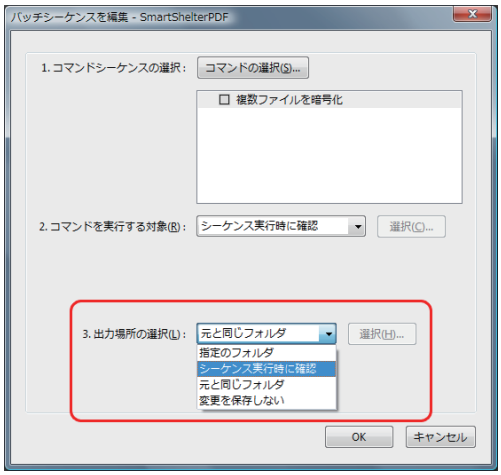
シーケンスの編集画面に戻り、「OK」ボタンをクリックします。

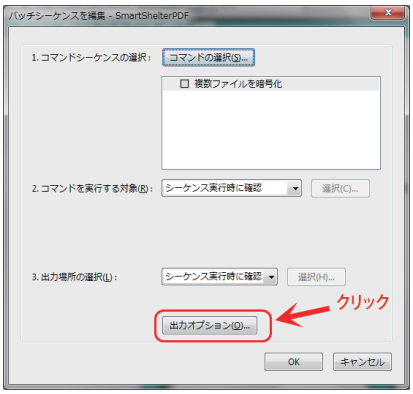


⑧ バッチシーケンスを編集する

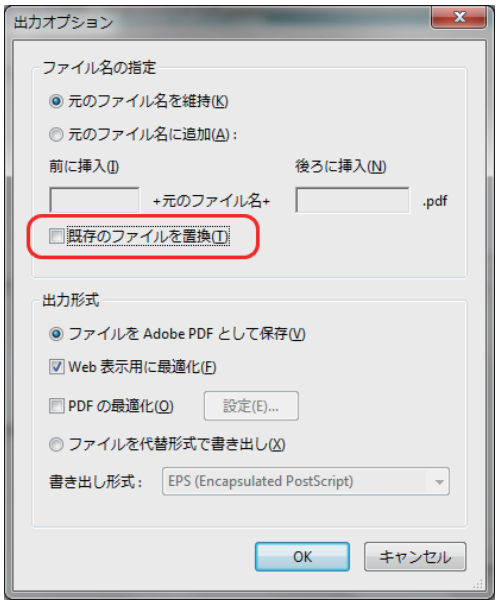
「バッチシーケンスを編集」画面の「出力場所の選択」で「シーケンス実行時に確認」を選択します。「元と同じフォルダ」を選択すると、オリジナルファイルが暗号化ファイルに上書きされるのでご注意ください。

また、念のため、「出力オプション」を選択し、「既存のファイルを置換(T)」のチェックを外しておくと、上書き保存の防止になります。なお、バッチシーケンス編集についての詳しい操作方法はAcrobatのマニュアル等をご参照ください。





「OK」ボタンをクリックすると、バッチシーケンスが登録されます。

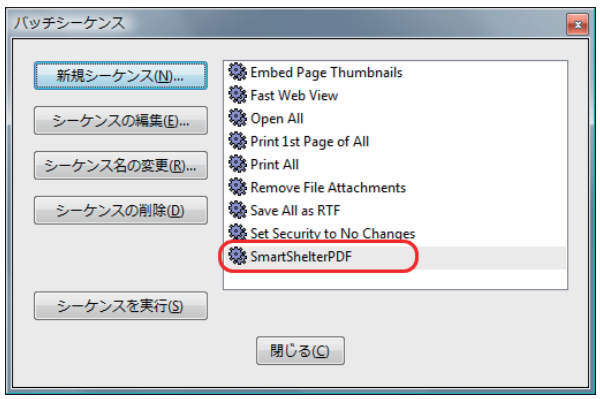


⑨ SmartShelterPDF が登録された

「バッチシーケンス」画面に「SmartShelterPDF」シーケンスが登録されたことを確認します。

「閉じる」ボタンをクリックすると、バッチ処理が終了します。

登録するバッチシーケンスは、暗号化内容に応じて複数作成することができます。貴社のセキュリティ内容に応じて使い分けてください。



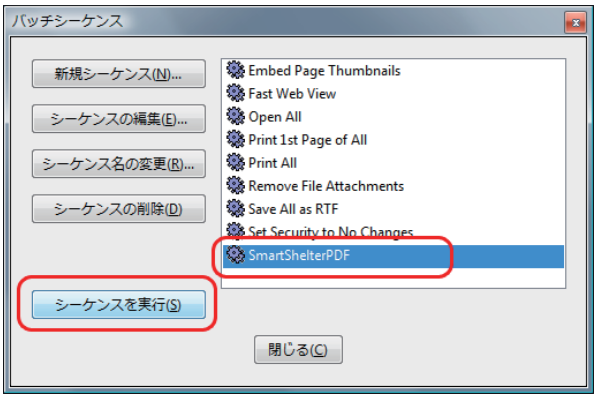
バッチシーケンスを実行して、PDF ファイルを一括暗号化する

① コードメータ FSB を装着する

まず、コードメータFSBをPCに装着します。PDFファイルを暗号化するには、必ずコードメータFSBが必要になります。

② バッチシーケンスを開く

Acrobatを起動し、「アドバンスト」/「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。

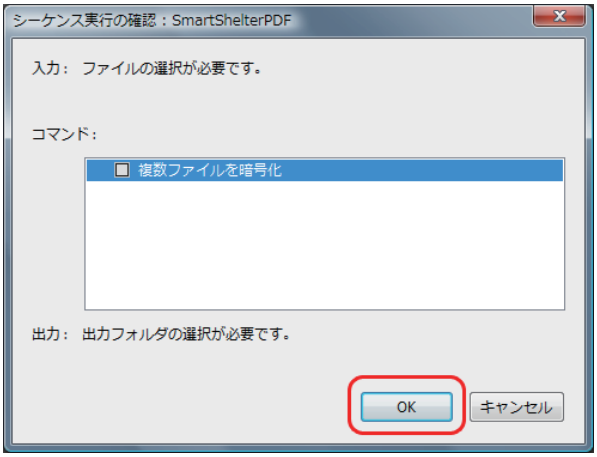


③ バッチシーケンスを実行する

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、指定するファルダに作成するだけです。

4-10. 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。

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。
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。
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 2000/XP/Vista/7 (32bit/64bit) Windows Server 2000/2003/2008 (32bit/64bit)
Adobe Acrobat 6/7/8/9 (32bit/64bit) またはAdobe Reader 6/7/8/9 (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。)

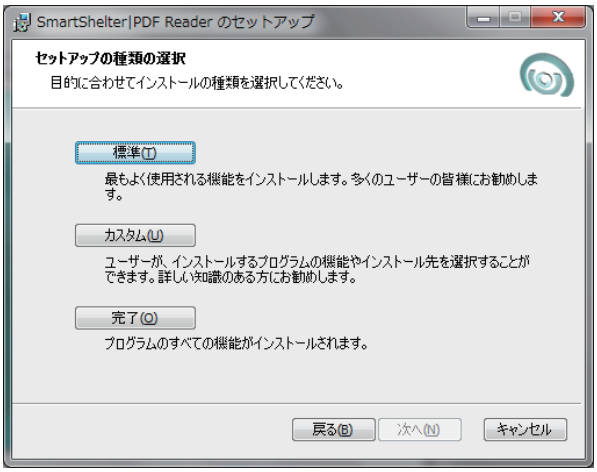
ユーザー用プログラム "SmashPdfRdr.exe" について

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。

Chapter 5

自動暗号化ツール AxProtector について

5-1. 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。

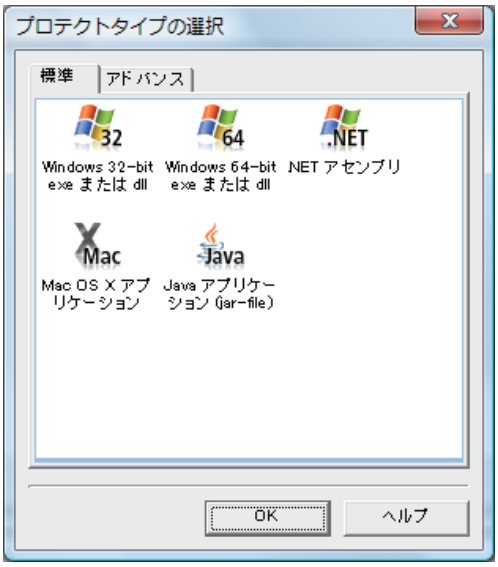
AxProtectorの対象になるファイルは以下のとおりです。

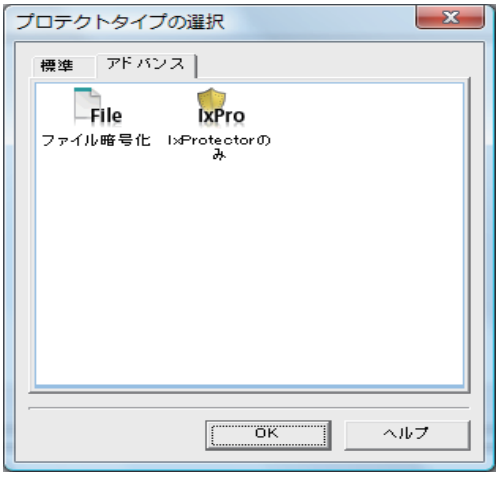
1. Windows32bit 実行形式プログラム (EXE, DLL)
2. Windows64bit 実行形式プログラム (EXE, DLL)
3. .NET アセンブリ実行形式プログラム
4. Mac OS X 実行形式プログラム
5. Javaプログラム

また、アドバンス機能として、

6. ファイル暗号化
7. IxProtectorのみ

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。

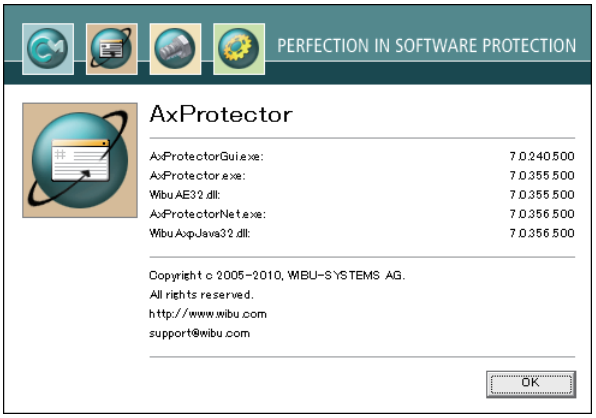
Hotfix について：

.NET プログラムをAxProtectorで暗号化する場合、下記HotfixをOS環境に合わせてインストールする必要があります。インストールする必要があるのは、AxProtectorで暗号化作業を行うPCのみで、暗号化されたプログラムを実行するPC（ユーザー先）には必要ありません。これは、.NET Framework 2.0 Service Pack1における不具合によるもので、詳細につきましてはマイクロソフトサイトKB950986 (**http://support.microsoft.com/kb/950986/ja**) をご参照ください。なお、下記Hotfixは、コードメータCDのTools¥Hotfixフォルダにあります。
または、弊社サイトhttp://www.suncarla.co.jp/download/からダウンロードできます。

Hotfix 401752 (KB950986) Windows 2000/XP 32-bit用 NDP20SP1-KB950986-x86.exe
Hotfix 401752 (KB950986) Windows XP 64-bit用 NDP20SP1-KB950986-x64.exe
Hotfix 401752 (KB950986) Windows Vista 32-bit用 Windows6.0-KB950986-x86.msu
Hotfix 401752 (KB950986) Windows Vista 64-bit用 Windows6.0-KB950986-x64.msu

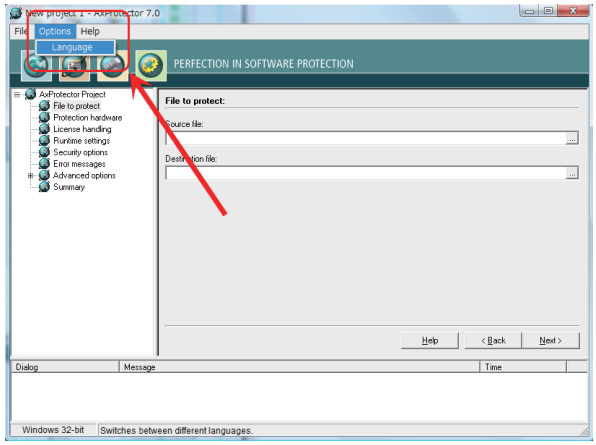
AxProtector 7.00 バージョン

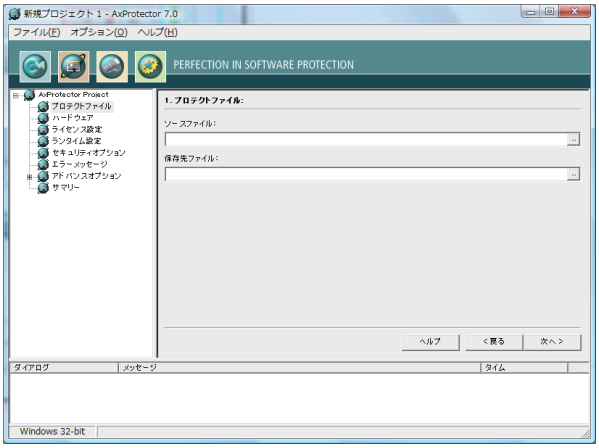
AxProtectorGui.exe 7.0.240.500
AxProtector.exe 7.0.355.500
WibuAE32.dll 7.0.355.500
AxProtectorNet.exe 7.0.356.500
WibuAxpJava32.dll 7.0.356.500



5-2. 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。





5-3. AxProtector のメニュー画面

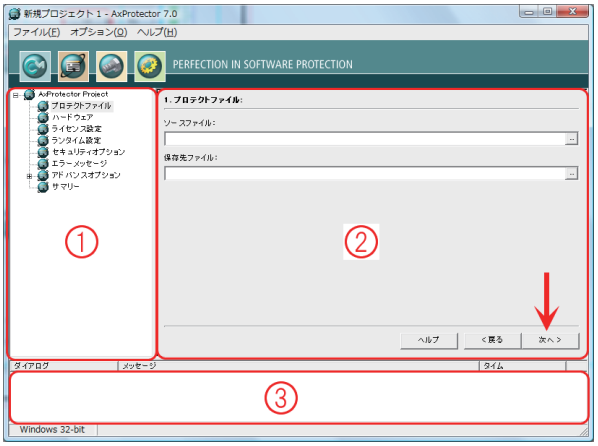
①のペインにはAxProtectorの操作項目が表示されます。各項目を上から順に実行すれば自動的に全てのパラメータを設定できるようになっています。ここで選択した項目に対するパラメータの値は②のペインで入力します。

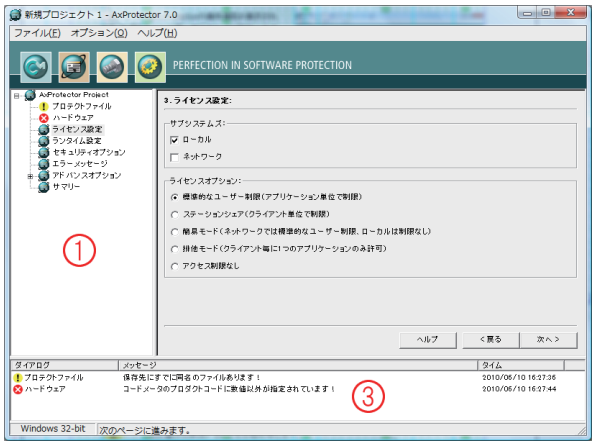
②のペインで実際のパラメータを入力、設定します。入力が終了したら [次へ]ボタンをクリックすれば、次のプロテクト項目の設定画面に切り替わります。この操作は、①のペインで１つ下のプロテクト項目を選択したのと同じ動作になります。

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伴ってリアルタイムに表示されます。

尚、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。

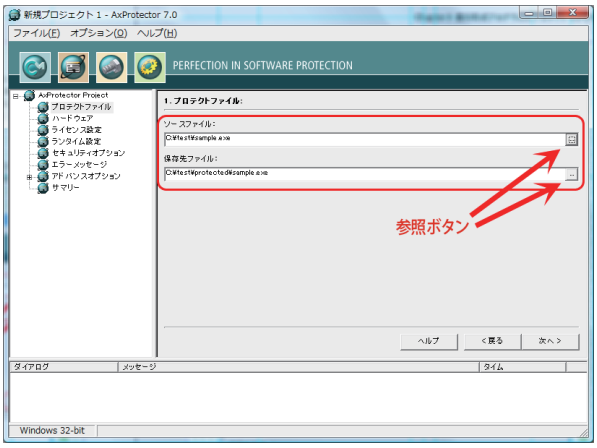
・・・ 警告を表します。警告の内容によっては無視しても構わない場合がありますが、できるだけ警告を解消することをお勧め致します。

・・・ 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。

5-4. 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。



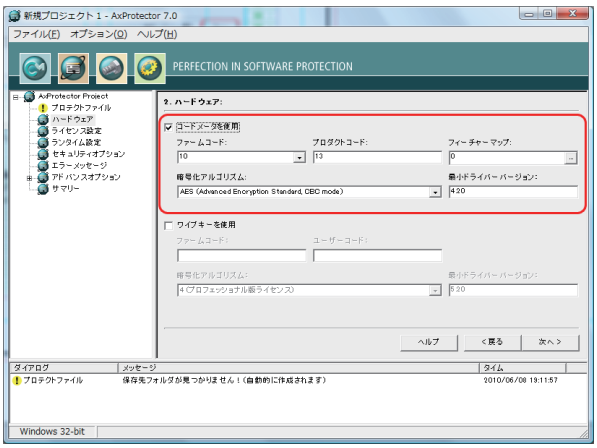
ソースファイル:
暗号化する前のオリジナルファイル名を指定します。右部の参照ボタンからファイルを指定することもできます。

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

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

2. ハードウェア

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、コードメータ以外にワイブキーを使用することができます。



☐ コードメータを使用
コードメータ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
ワイブキーを使用する場合はここにチェックを入れます。

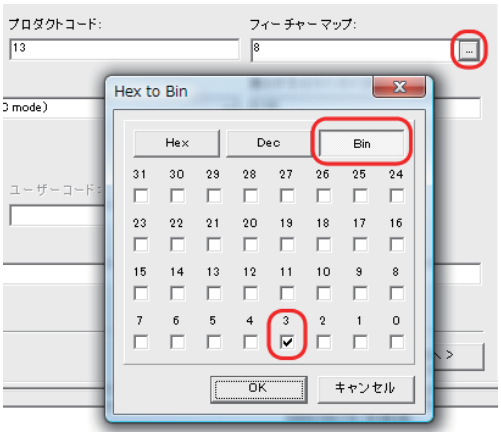
[NOTE]
コードメータとワイブキーを同時に指定することも可能です。その場合、暗号化されたプログラムはコードメータまたはワイブキーのどちらかが見つかると動作します。

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、0 ～ 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。

0x00000000000000000000000000001000(Bin) = 8 (Dec)

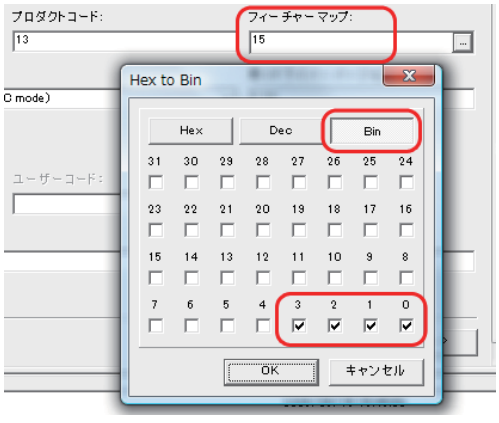
ファームコード=10、プロダクトコード=13、フィーチャーマップ=8で暗号化されたプログラムを起動するには、同じコード（ファームコード=10、プロダクトコード=13、フィーチャーマップ=8）を持つCM-Stickが必要になります。

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

0x00000000000000000000000000001111 = 15 (Dec)

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



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

0x00000000000000000000000000011111 = 31 (Dec)

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

暗号化アルゴリズム：
AES (Advanced Encryption Standard) 128ビットを指定。

最低限必要なドライバーバージョン：
使用するコードメータランタイムキットの最低バージョンを指定します。

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。
（１PC ＝ 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)
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がネットワーク(LAN)上で見つかれば、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。

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）、ただちにアプリケーションを終了させず、許可した回数だけアプリケーションを続行させる機能です。例えば、ここで３（デフォルト）を設定すると、３回まではランタイムチェックエラーになっても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強制終了します。

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停止させたい場合は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し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：

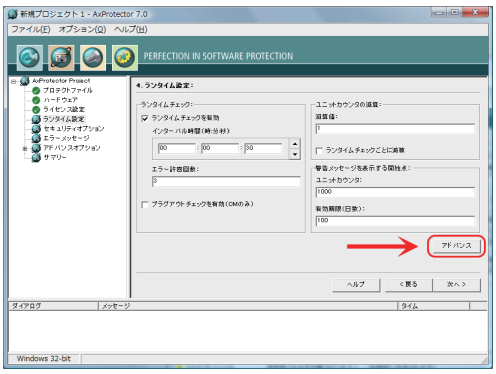
ユニットカウンタや有効期限（日数）の残りが一定の値を下回ると、警告メッセージを表示させるオプションです。

☐ **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）

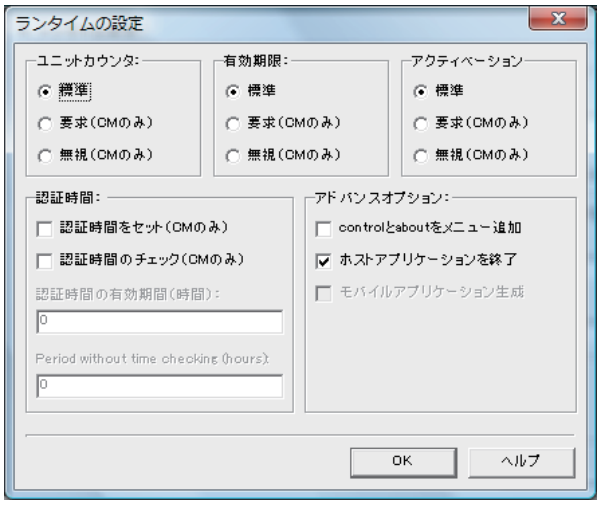
☐ **有効期限（日数)：**
有効期限まであと何日になると警告メッセージを出すかの開始点を設定します。警告メッセージは有効期限に達するまで、アプリケーションを起動するたびに表示されます。出力される警告メッセージは有効期限警告用メッセージです。
（デフォルト値は100日：有効期限（日数）が100になると警告メッセージを表示し始める）

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：



○標準

コードメータキー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（CMのみ）

ユニットカウンタを必ずチェックする。もし、コードメータキー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。
（コードメータのみ対応）

○無視（CMのみ）

ユニットカウンタを無視する。コードメータキー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。ユニットカウンタ機能なしの状態になる。
（コードメータのみ対応）

有効制限：

○標準

コードメータキー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（CMのみ）

有効期限を必ずチェックする。もし、コードメータキー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。（コードメータのみ対応）

○無視（CMのみ）

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

アクティベーション（使用開始日）：

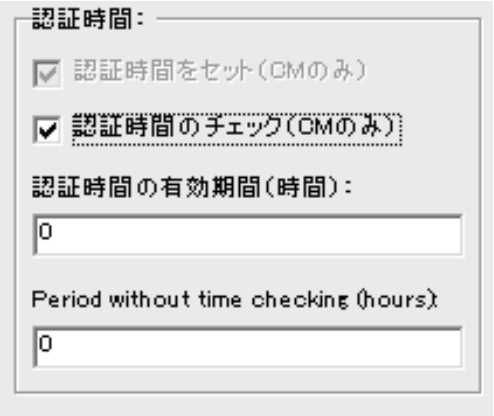
○標準

コードメータキー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（CMのみ）
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。（コードメータのみ対応）

○無視（CMのみ）
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。アクティベーションタイムなしの状態。
（コードメータのみ対応）

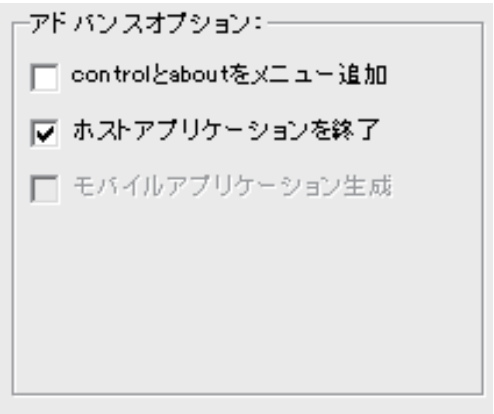
認証時間：



□認証時間をセット(CMのみ)
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タイムサーバーにアクセスし、コードメータキーの認証時間とボックス時間を更新します。（コードメータのみ対応）

□認証時間のチェック（CMのみ）
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、コードメータキーの認証時間と比較します。もし、「認証時間の有効時間（時間）」で設定された時間よりも差がある場合（コードメータキーの認証時間が遅い場合）はプログラムの起動を中止します。（コードメータのみ対応）

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）

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。(ただし、「スタティックコードの変更」と「拡張スタティックコードの変更」は同時に指定できません)

□リソースセクションの暗号化
リソースを暗号化します。

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

□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。

□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」と同時には指定できません。

アンチデバッグの設定

アンチデバッギングの設定です。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差し障りが生じる場合は、以下を参考にチェックをはずしてください。

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

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

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

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

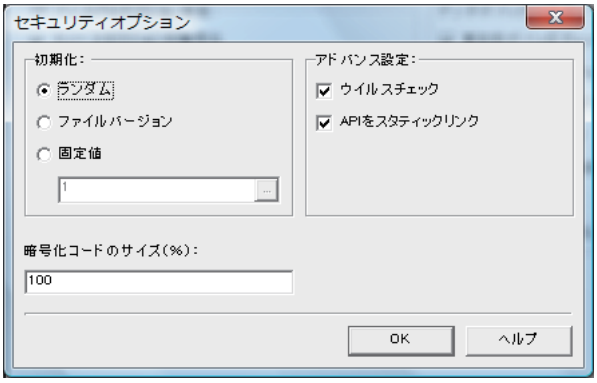
□一般的なデバッガ検知
デバッガがアプリケーションにアタッチされないようなメカニズムを追加します。

□仮想マシン検知
アプリケーションがバーチャルマシン上で起動できないようにします。

□デバッガを検知したらハードウェアキーをロック
デバッカの動作を検知した場合に、コードメータキー（ハードウェア）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（ハードウェア）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。

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。



初期化：

○ ランダム
ランダムデータ生成の初期値にランダムデータを使用します。

○ ファイルバージョン
ランダムデータ生成の初期値にファイルのバージョン情報を使用します。

○ 固定値
ランダムデータ生成の初期値に使用する固定値を設定します。暗号化を2回行なった場合、2回とも同じ結果が得られます。

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。ウイルス感染の疑いのある場合、プログラム起動時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。

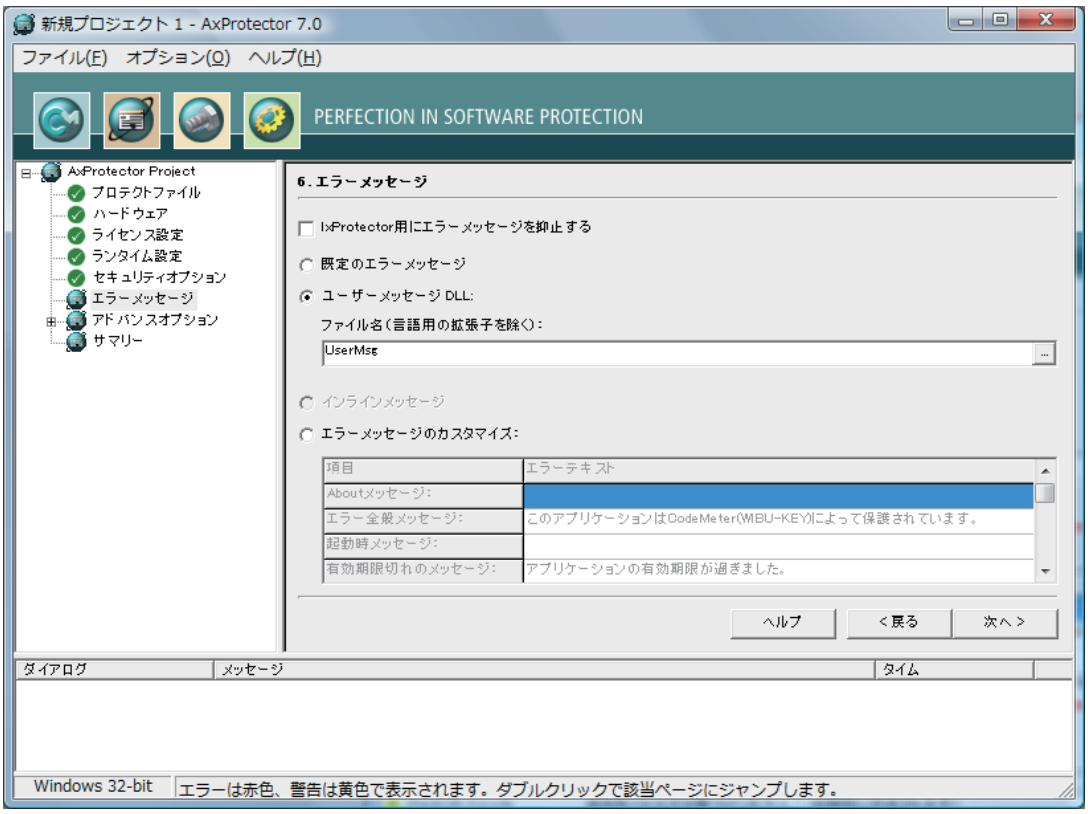
暗号化コードのサイズ（%）：

暗号化するコードサイズを%にて設定します。

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用にエラーメッセージを抑止する**

◯ **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“UserMsgJp.ini” ファイルの内容を直接編集します。デフォルトの状態で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"UserMsgUs.ini"が作成されます。この"UserMsgUs.ini"をコピーし、"UserMsgJp.ini"とリネーム後、メモ帳などのエディタで編集してください。編集した“UserMsgJp.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。

【UserMsgJp.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)

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）

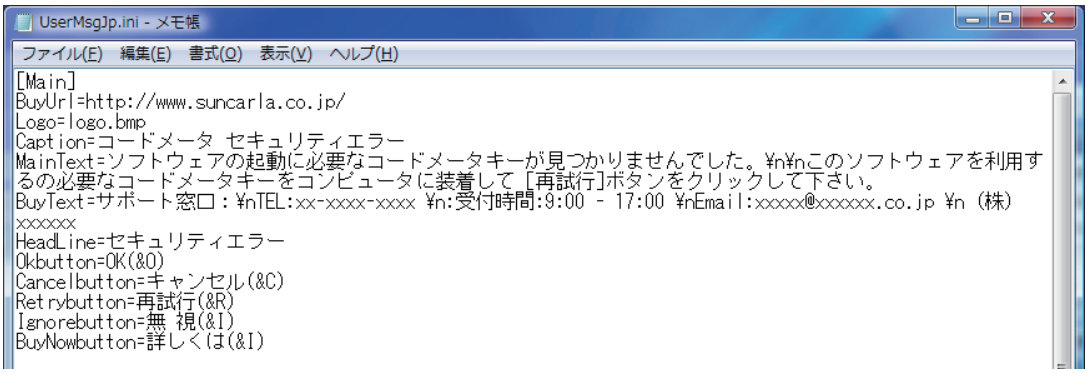
DaysMax = プログレスバーに表示する有効日数最大数（例：100）

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:）

【操作例】

1.「UserMsgJp.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。

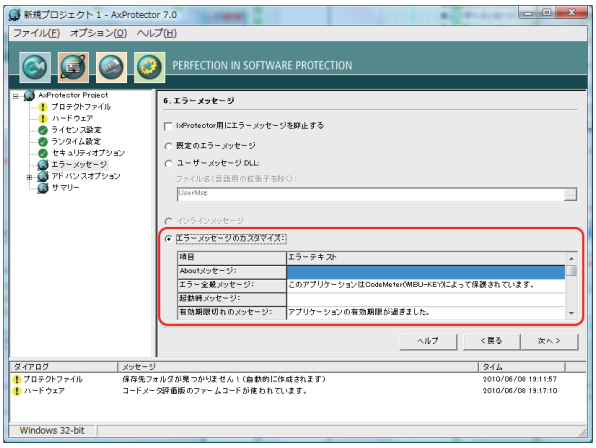


2. 暗号化されたプログラムと同じフォルダに、"UserMsgUs.dll" と "UserMsgJp.ini"および使用した画像ファイル(BMP)をおき、コードメータキーを装着しない状態でプログラムを実行するとエラーメッセージが表示されます。

3. 暗号化されたプログラムを配布する際は、必ず"UserMsgUs.dll" と "UserMsgJp.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 "UserMsgJp.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。

[NOTE]
次にあげる0x5C を含む文字において文字化けが発生しますので、これらの文字を使用する場合は、該当文字の後ろに ¥ を追加してエスケープしてください。

文字化けが発生する文字は以下のとおりです。

―	ソ	㏂	Ⅸ	噂	浬	欺	圭	構	蚕	十	申	曾	箪	貼	能	表	暴	予	禄
兔	喀	媾	彌	拿	杤	歓	淦	畚	秉	綵	臀	藹	觸	軆	鐔	饅	鷭	偬	砬

(例)
"予め申し上げます。"　という文字列の場合、予 と 申 が上記のグループに含まれるため該当する文字の直後に半角 ¥ マークを付けて "予¥め申¥し上げます。" とします。

○ 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
UserMessageDE.ini
UserMsg.bmp

日本語メッセージを作成するには、UserMessage.iniをコピーして、UserMessageJA.iniとリネームし、内容を編集します。

[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。（xxxxJp.iniでなくxxxxJA.ini）
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"UserMsgJp.ini"です。また、.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。）

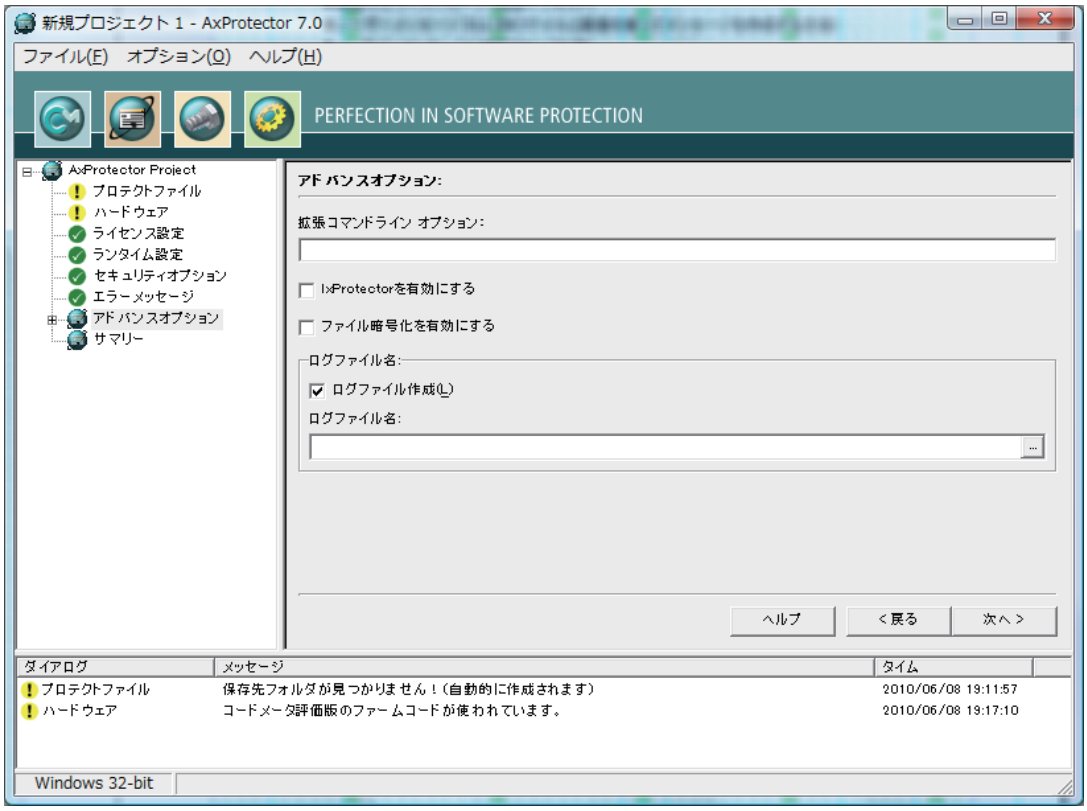
.NET アセンブリの場合 -->UserMessageJA.ini
Windows32/64アプリケーションの場合-->UserMsgJp.ini

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を有効にする

IxProtector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」を実現する新しい機能です。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)

[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。

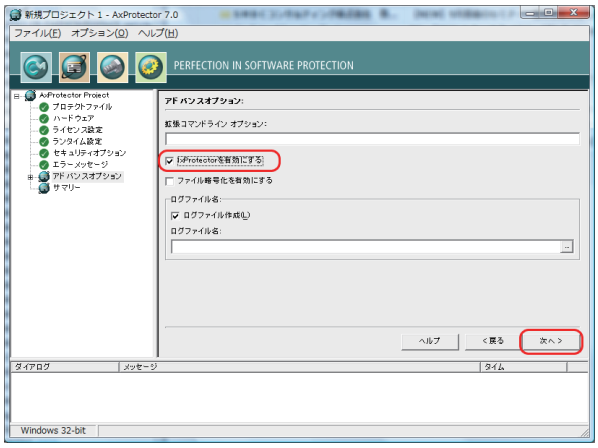
☐ ログファイル作成

暗号化処理のログファイルを作成します。

7-1. IxProtectorの使い方

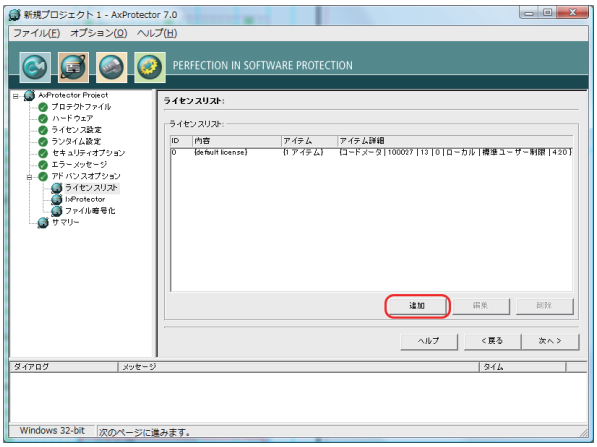
① IxProtector を有効にする

アドバンスオプション画面で、「IxProtector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。



③ ライセンス内容を登録する

「ライセンスリストの追加」画面でライセンス内容を登録します。

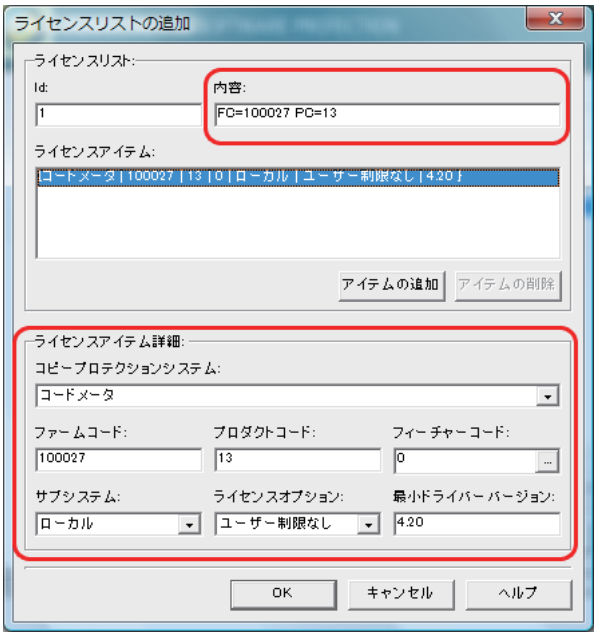
Id:
ライセンスIDを設定する。ライセンスIDは連番で自動表示されます。

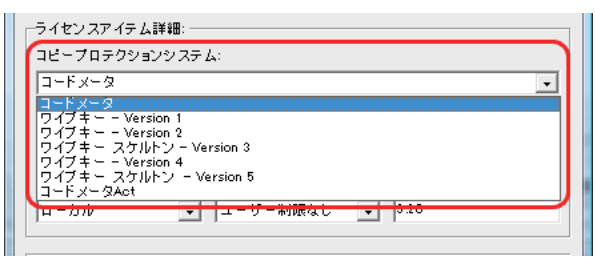
内容:
ライセンス内容を記載します。

コピープロテクションシステム:
使用するハードウェアキーを指定します。コードメータ(CodeMeter)、ワイブキー(Wibukey)、コードメータAct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。

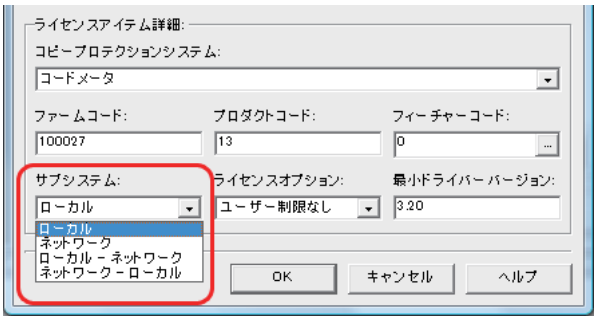




フィーチャーコード:
ライセンスIDに割り当てるフィーチャコード（フィーチャーマップ）を設定します。

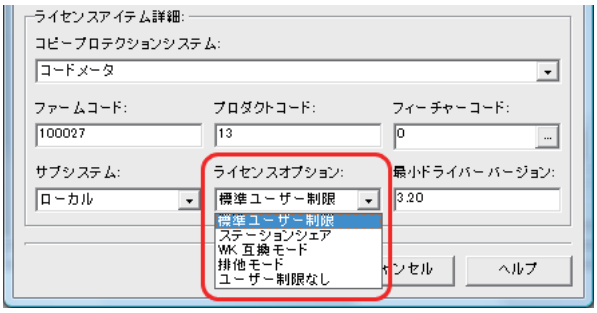
サブシステム:
サブシステムを割り当てます。

"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）
"ローカルーネットワーク"
（最初にローカルアクセス、次にネットワークアクセス）
"ネットワークーローカル"
（最初にネットワークアクセス、次にローカルアクセス）

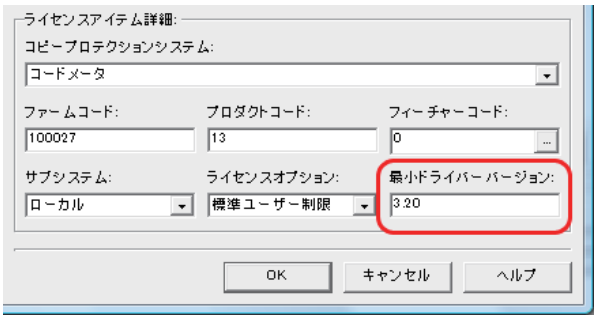


ライセンスオプション:
ライセンス形態を割り当てます。

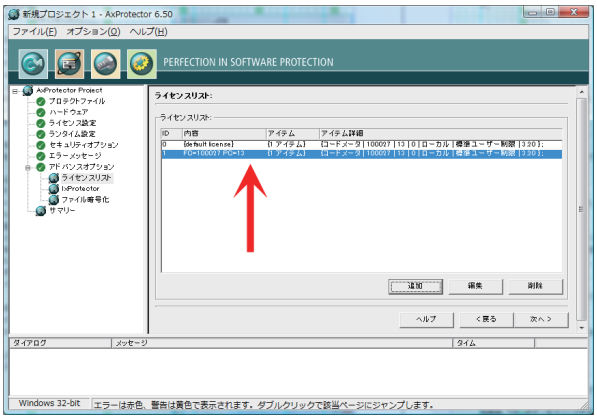
"標準ユーザー制限"
（アプリケーション単位で制限）
"ステーションシェア"
（クライアント単位で制限）
"WK 互換モード"
（ネットワークでは標準制限、ローカルは制限なし）
"排他モード"
（クライアントごとに 1 つのアプリケーションのみ許可）
"ユーザー制限なし"
（ユーザー無制限で使用可能）



最小ドライバーバージョン:
コードメータを動作させる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。

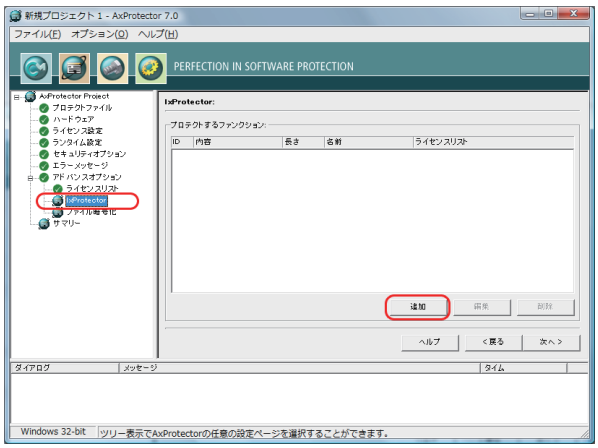


「OK」ボタンをクリックするとライセンス内容がライセンスアイテムに追加登録されます。



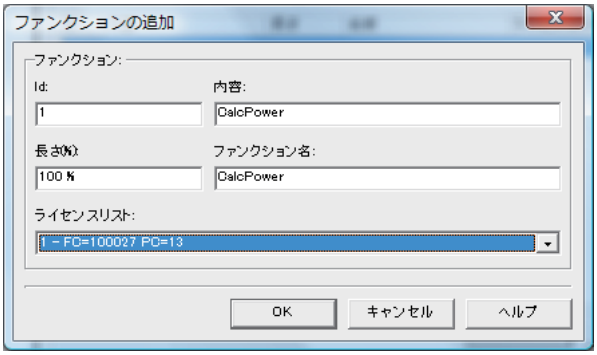
④ IxProtector を登録する

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



⑤ ファンクション名を登録する

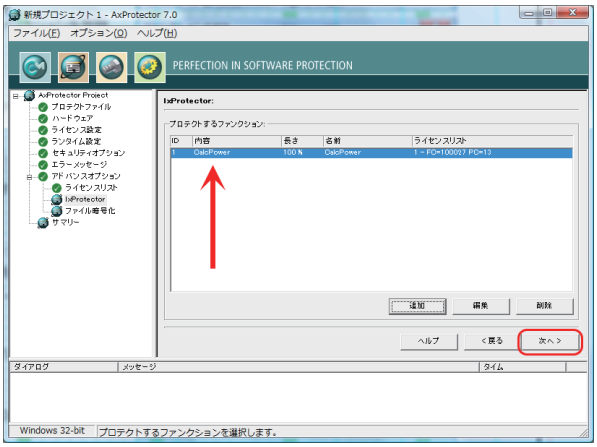
「ファンクションの追加」画面でファンクションを登録します。



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

内容:
ファンクションの内容を登録します。

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



ファンクション名:
ソースコードの中で実際に使われているファンクション名を正確に入力します。
(例)
CWupiCalculatorDlg::CalcSimpleOperation
OnButtonCalcMemClear
CWupiCalculatorDlg::CalcAngle

ライセンスリスト:
ライセンスリストから使用するライセンスを指定します。

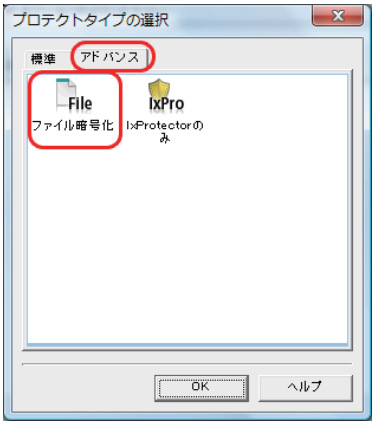
登録後、「OK」ボタンをクリックすると、IxProtectorにファンクションが登録されます。
「次へ」ボタンをクリックして次に進めます。

7-2. ファイル暗号化の使い方

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

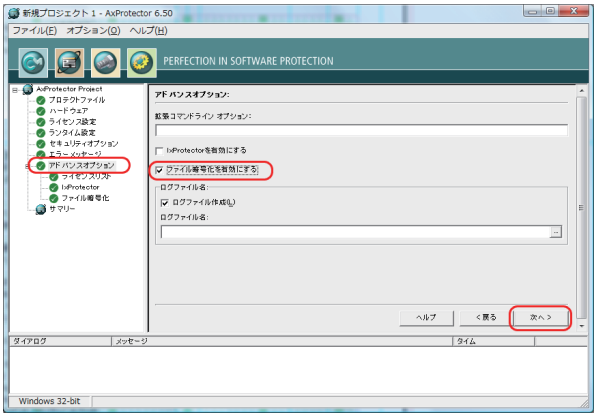
なお、データファイル自身はAxProtectorの"ファイル暗号化"で最初に暗号化しておきます。

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

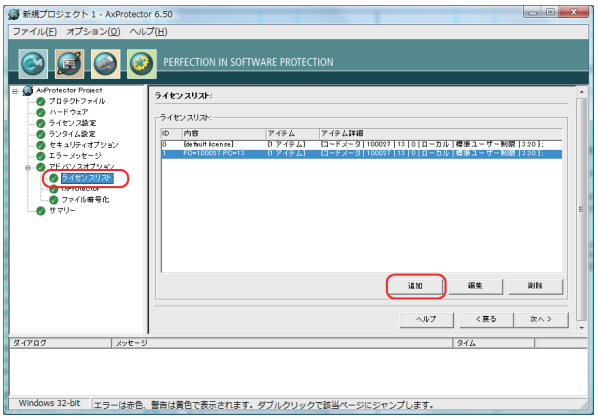
「アドバンスオプション」画面で、「ファイル暗号化を有効にする」にチェックを入れ、「次へ」をクリックする。



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。

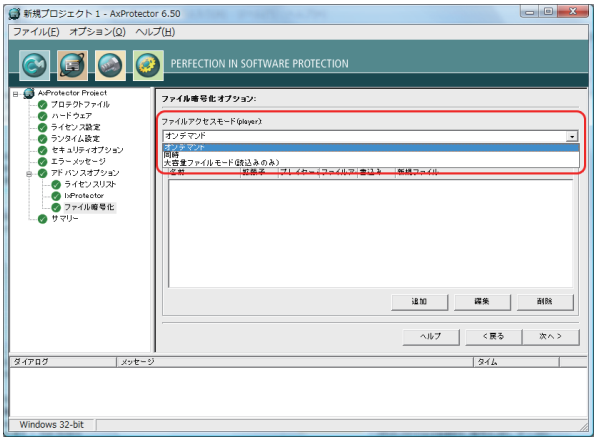
ライセンスリストに登録されているライセンスは、IxProtectorとファイル暗号化の両方で使用可能です。IxProtectorを使用しない場合は、IxProtector画面がグレーアウトしますので、「次へ」ボタンをクリックして次に進めます。



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（一括）
大容量ファイルモード（読込みのみ）



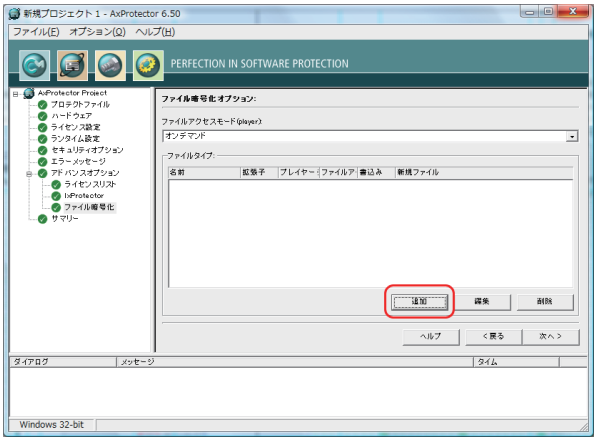
オンデマンド:
「オンデマンド」モードでは、プログラム（Player)は、ファイル全体のメモリーを確保しますが、読み込むのは必要な部分だけです。読み込みと復号化は4kバイトブロック単位で行い、必要な時に必要なブロックをロードし復号化します。一度復号化されたブロックは、メモリー上に残るため、2回目以降はメモリー上のブロックが使用されます。この「オンデマンド」モードはロードされるファイル全体のメモリーが必要になりますが、一度復号化されたデータを再利用できるため、パフォーマンスに優れます。このモードは、データファイルを読み書き(Read/Write)する場合に適します。

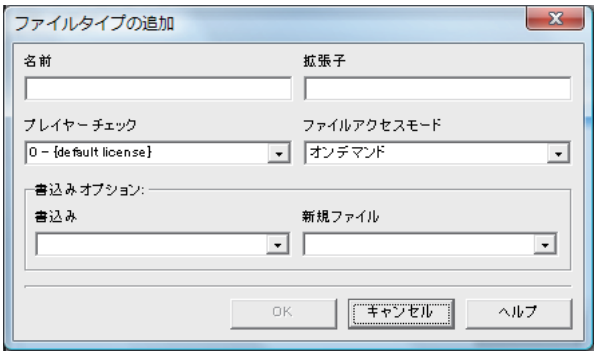
同時（一括）:
「同時（一括）」モードでは、プログラム（Player)は、ファイル全体のメモリーを確保し、ファイル全体をメモリーに読み込んでから一度に復号化します。復号化されたデータはメモリー上に残り、必要な都度メモリー上から読み出されます。ファイル全体のメモリーが必要になりますが、復号化済のデータを利用できるためパフォーマンスに優れます。「オンデマンド」モードと異なる点は、最初のアクセス時にファイル全体を復号化するための時間が必要となることです。このモードは、「オンデマンド」モード同様、データファイルを読み書き（Read/Write)する場合に適します。

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（読込みのみ）」から選択します。

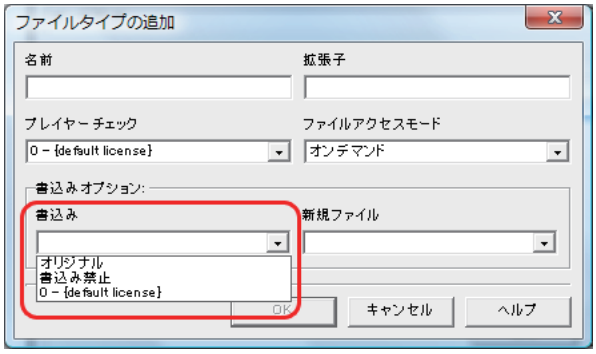
書込みオプション:

書込み:
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。

書込み禁止
ファイルへの書込みを禁止する。

0 - {default license}
デフォルトライセンス仕様にて保存する。



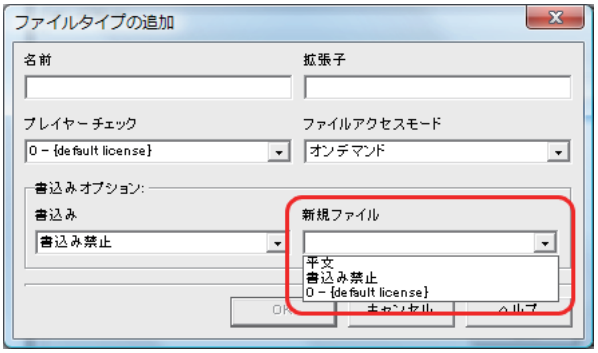
(License List)
指定したライセンス仕様で保存する。ライセンスリストで追加されたライセンスが表示されます。

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。

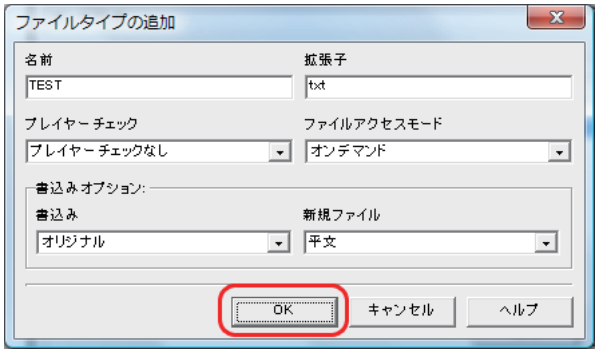
書込み禁止
新規ファイルは作成しない。新規保存できない。

0 - {default license}
デフォルトライセンス仕様にて作成保存する。

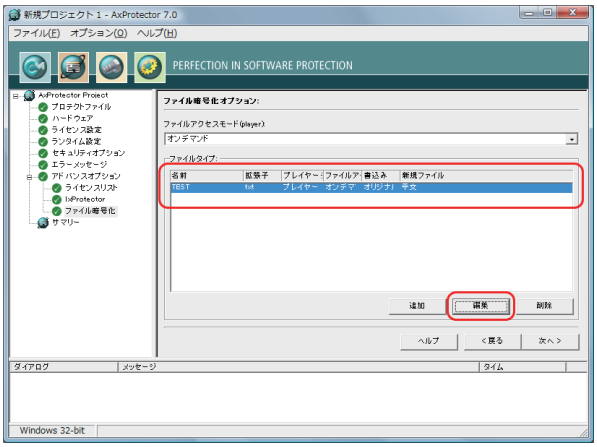


(License List)
指定したライセンス仕様で作成保存する。指定したライセンス仕様で保存する。ライセンスリストで追加されたライセンスが表示されます。

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。

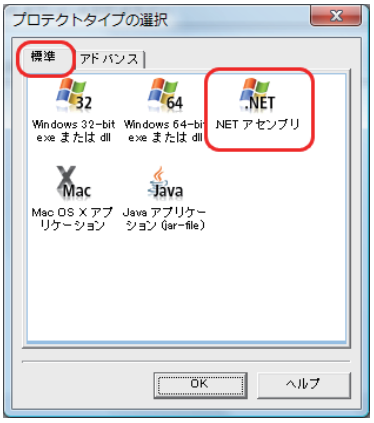


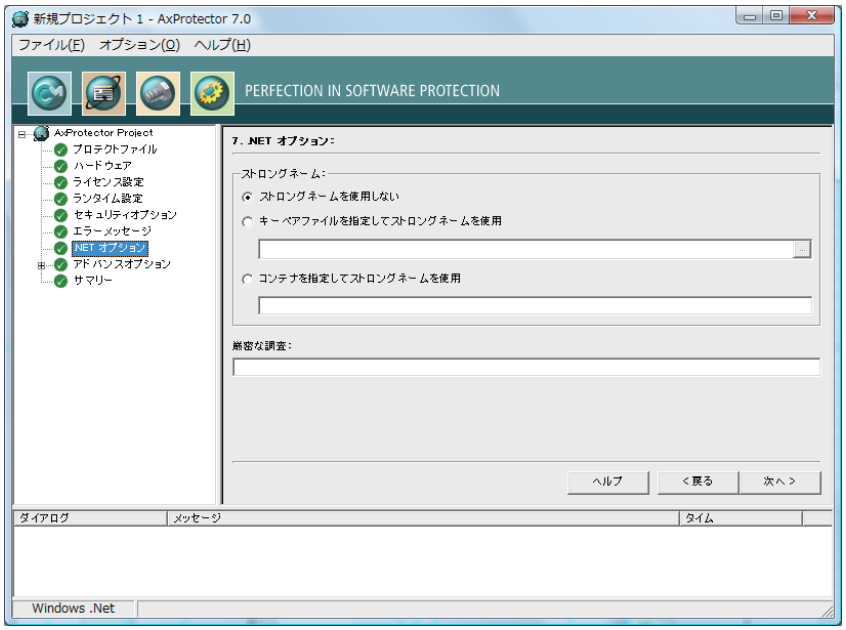
なお、登録したファイルタイプは「編集」ボタンをクリックすることで編集が可能です。



7a. .NET オプション：（.NET アセンブリの場合）

AxProtector起動時に、「プロテクトタイプの選択」画面で「.NETアセンブリ」を選択した場合、この「.NETオプション」画面が表示されます。





ストロングネーム：

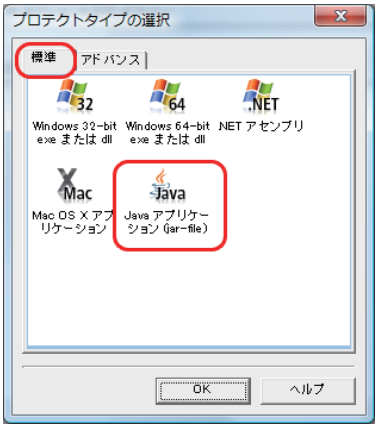
○ ストロングネームを使用しない
ストロングネームを使用しない場合に選択します。

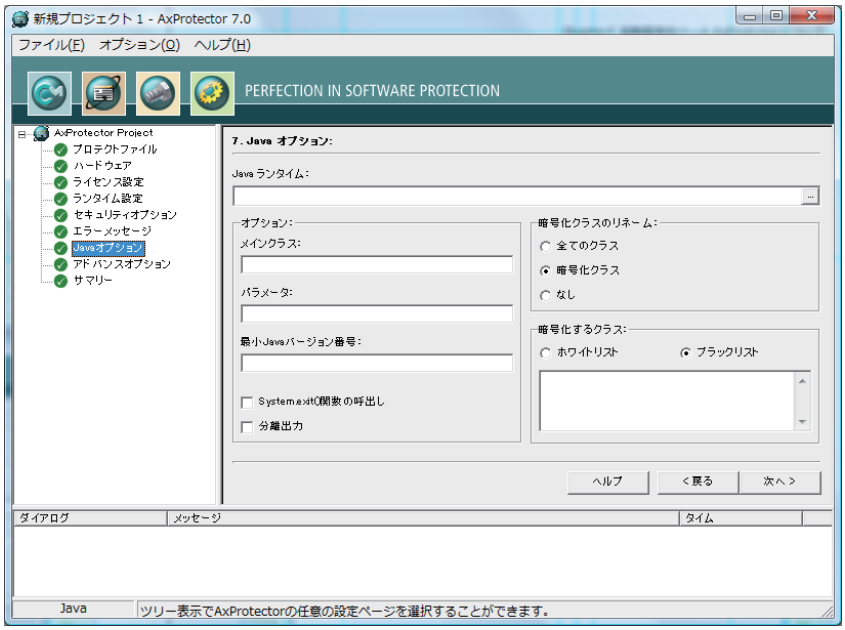
○ キーペアファイルを指定してストロングネームを使用
ストロングネームをキーファイルから使用する場合にチェックし、任意のキーファイルの保存先を指定します。

○ コンテナを指定してストロングネームを使用
ストロングネームをCSPコンテナから使用する場合にチェックし、任意のキーコンテナ名を指定します。

7b. Java オプション（Java アプリケーションの場合）

AxProtector起動時に、「プロテクトタイプの選択」画面で「Java アプリケーション(jar-file)」を選択した場合、この「Javaオプション」画面が表示されます。





Javaランタイム：
使用するJavaランタイムを選択することができます。

オプション：
メインクラス：
起動するMain Classを指定することができます。マニュフェストでMain-Classを指定していない場合、必ずこのオプションでMain-Classを指定してください。

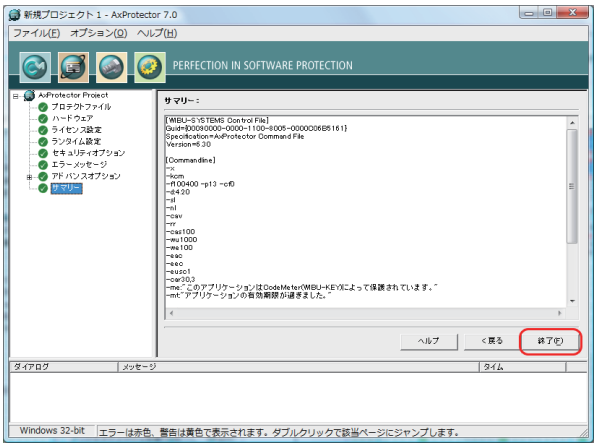
パラメータ：
Main-Classのための引数を指定することができます。

最低限必要とされるJavaバージョン番号：
使用するJava Versionの下限を定義することができます。

□ System.exit()関数の呼び出しでアプリケーションを終了
System.exit()関数を呼び出すことにより、アプリケーションを終了します。

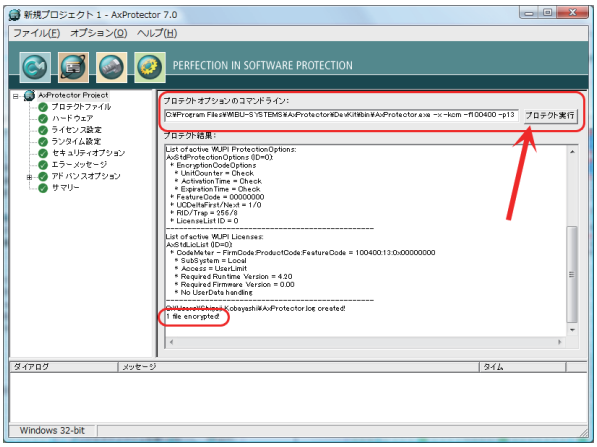
8. サマリー：

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



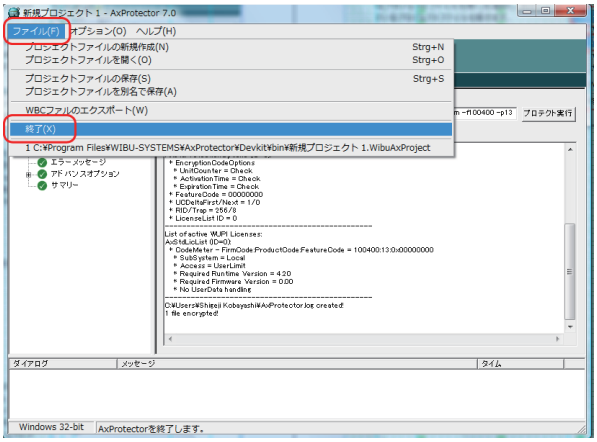
設定に問題がなければ、右下の「終了」ボタンがアクティブになります。もし、設定に問題がある場合は、「終了」ボタンが非アクティブの状態になります。

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

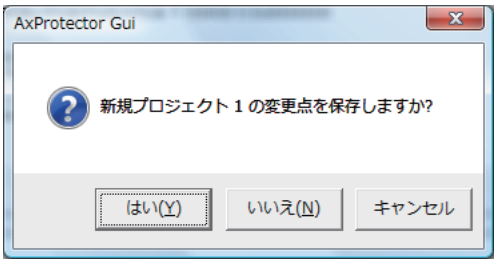


また、プロテクトオプションのコマンドラインからは、設定したオプションパラメータが表示されます。このコマンドラインを直接入力して、オプションパラメータを変更および追加をして暗号化処理を行うことも可能です。右部の「プロテクト実行」ボタンをクリックすると暗号化処理が開始されます。

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



閉じる際、プロジェクトファイルを保存するかの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。

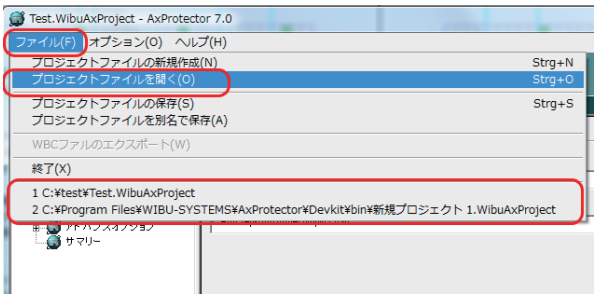


9. プロジェクトファイルを開く

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。

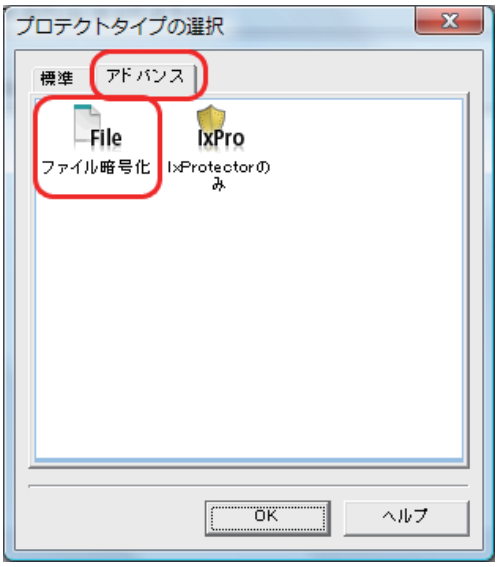
また、直近に使用したファイルが最下行に表示されますので、そこからファイルを開くこともできます。



5-5. データファイルを暗号化する

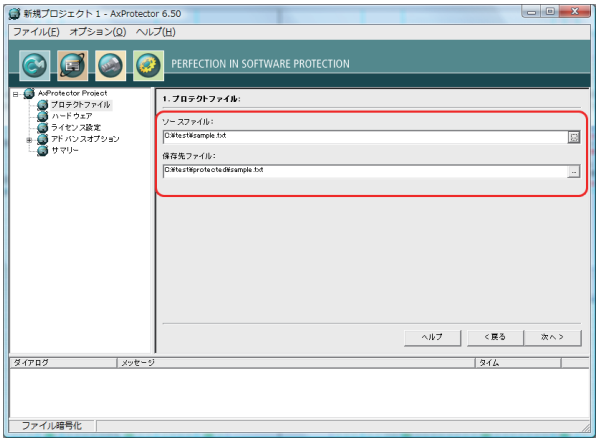
データファイルを暗号化する場合は、AxProtectorの"ファイル暗号化"機能を使用します。AxProtector起動後、「プロテクトタイプの選択」で「アドバンス」タブを選択し、「ファイル暗号化」を選択して"OK"ボタンをクリックするか、「ファイル暗号化」のアイコンをダブルクリックします。

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
Excel, Word, PowerPointなどのOffice系ファイル
その他のドキュメントファイル
です。



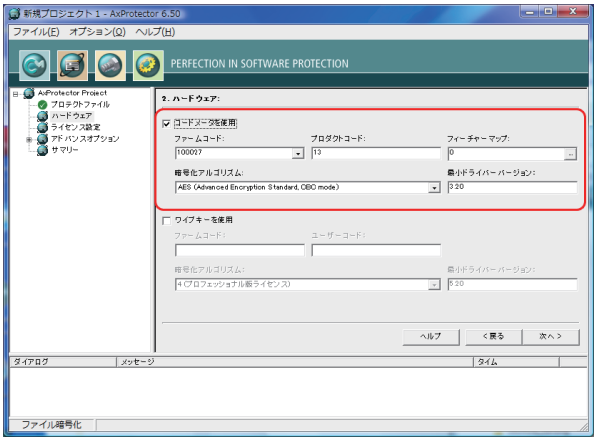
① ファイル名を入力する

「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。



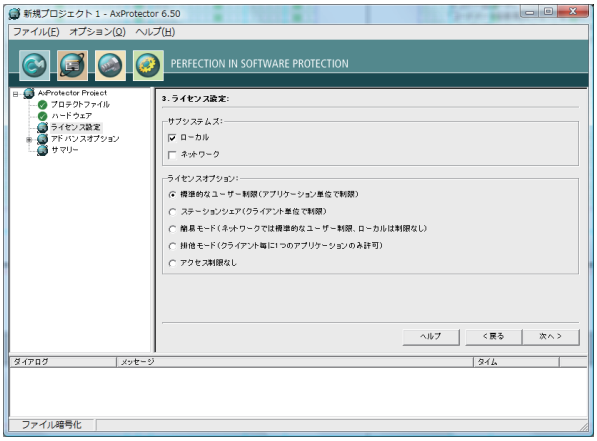
② ファームコード、プロダクトコードを入力する

「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、最小ドライバーバージョン（ランタイムバージョン）を設定します。



③ ライセンス設定を行う

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

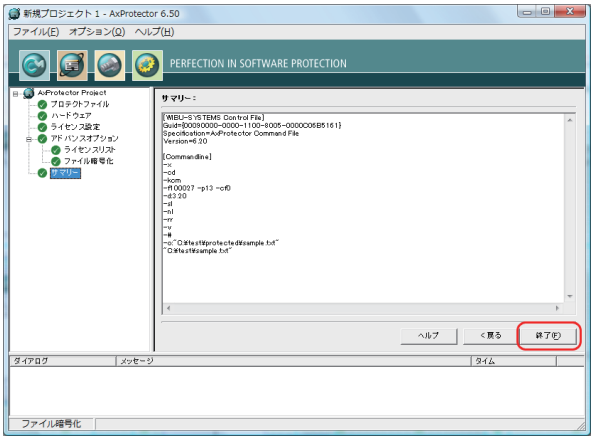


④ アドバンスオプション

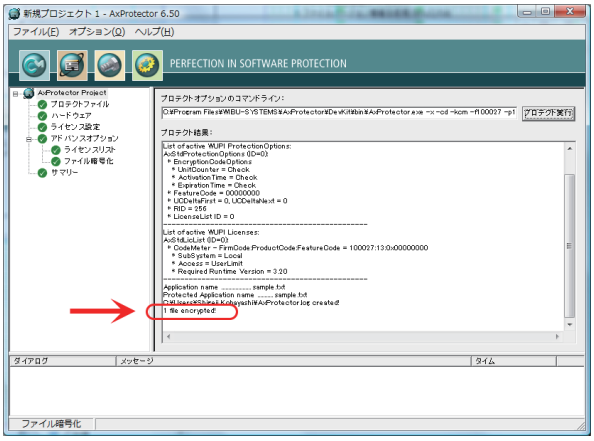
アドバンスオプション画面では特に設定する項目はありません。また、次のライセンスリストおよびファイル暗号化オプションでも設定する項目はありません。「次へ」ボタンで進めます。

⑤ 暗号化処理を開始する

サマリー画面で「終了」ボタンをクリックすると暗号化処理が開始されます。



サマリー画面で"1 file encrypted!"のメッセージが表示されれば暗号化に成功したことになります。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。



5-6. コマンドラインでの使用方法

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

axprotector.exe options filespec[/s]

以下は、オプションパラメータ設定の内容です。

基本設定：

/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: デバッガ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ーを指定します。

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

警告の開始点
E: 有効期限(Expiration Time)の開始点を指定<t>
U: ユニットカウンタ(Unit Counter)の開始点を指定<c>

Chapter 6

IxProtector/WUPI について

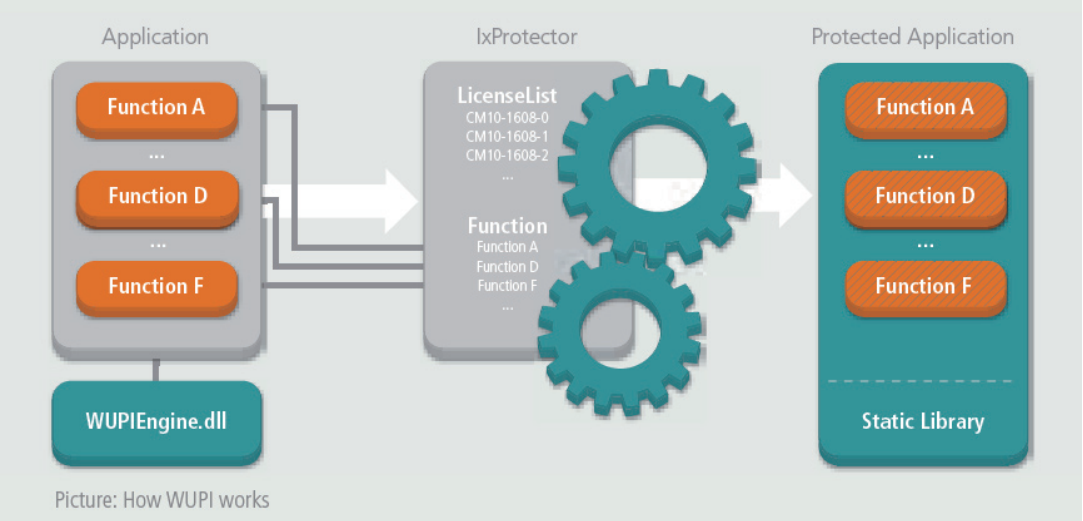
6-1. IxProtector とは

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。



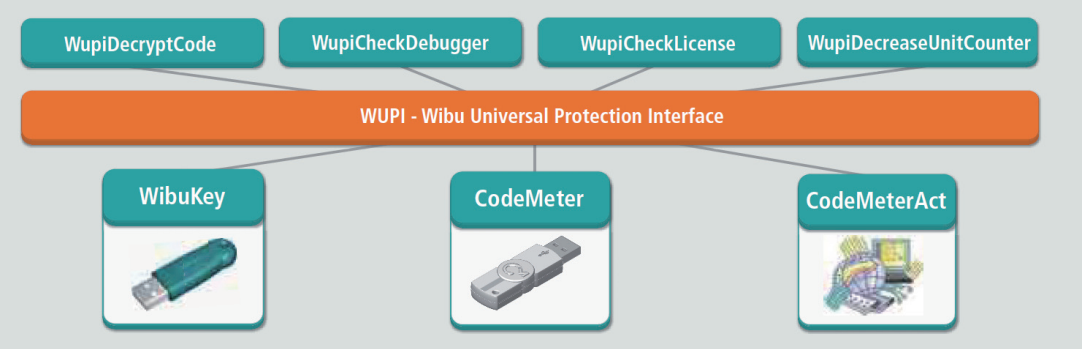
Picture: How WUPI works

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。

6-2. WUPI ファンクションについて

WUPI（ウーピィ）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。



6-3. 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。

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。

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。

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。

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。

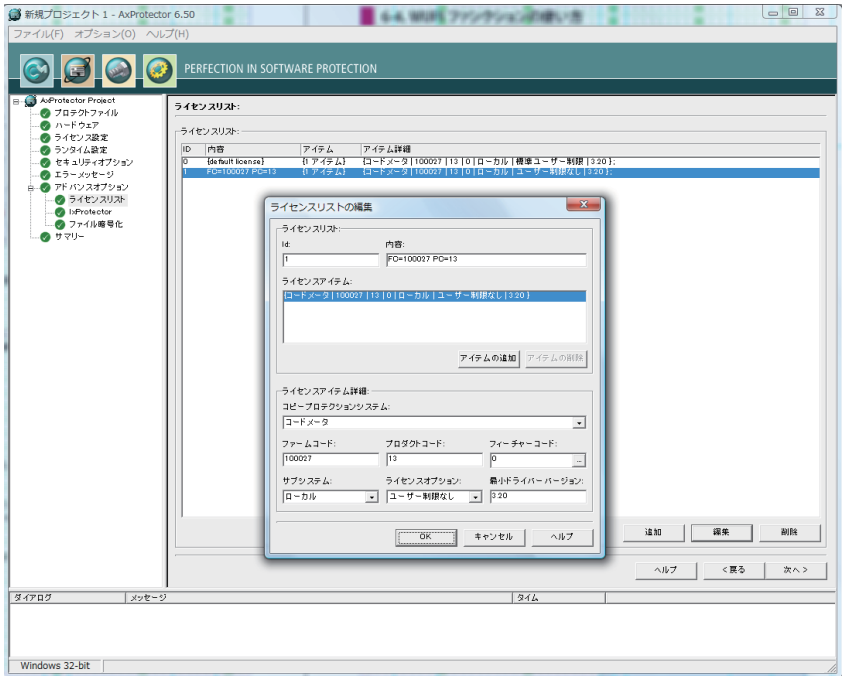
WupiGetLastError
直前に実行したWUPIファンクションのエラーコードを返します。

6-4. 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。

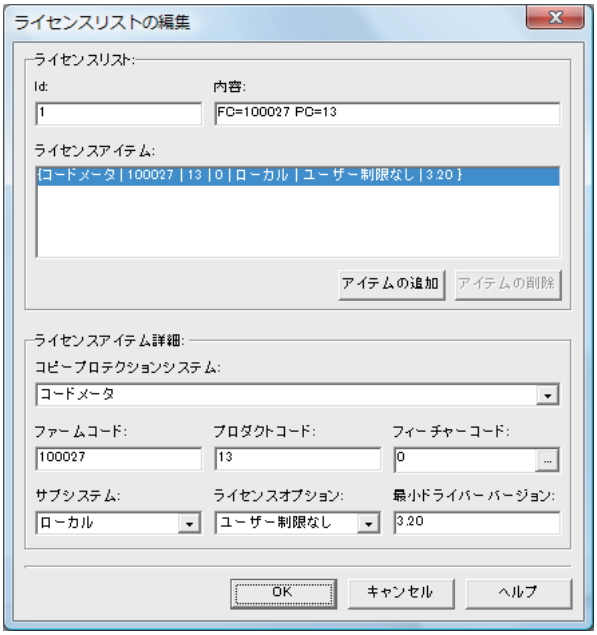


① ライセンスリストを作成する

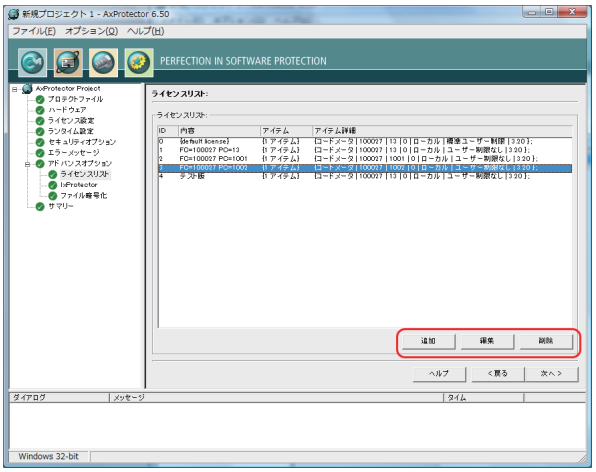
「ライセンスリストの編集」画面で、インデックスベースのライセンスを作成します。"Id=1"は"Index=1"を意味します。"内容"には、ライセンス名称を明記します。

"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。

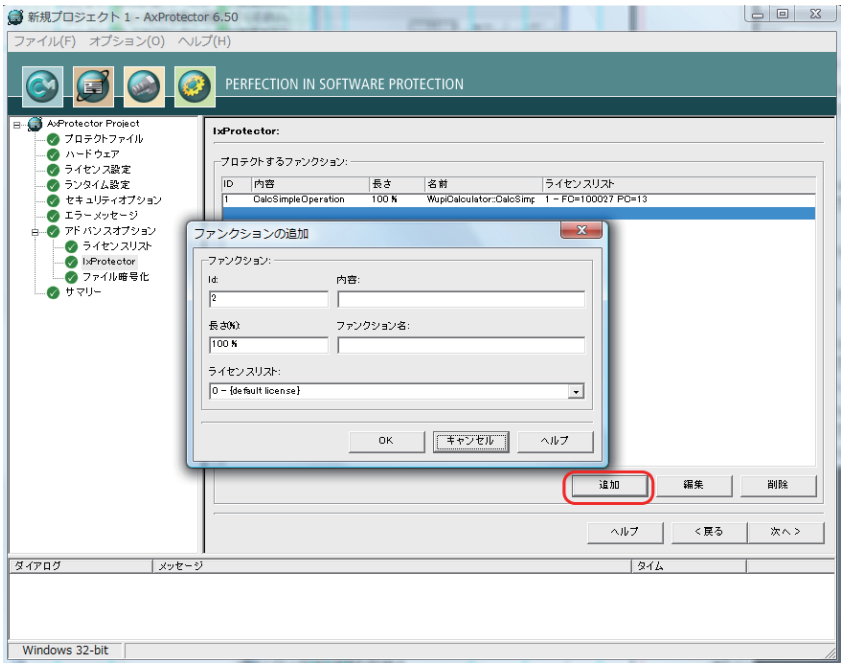
ここで作成されたライセンスリストが、WUPIファンクションから参照されます。

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。

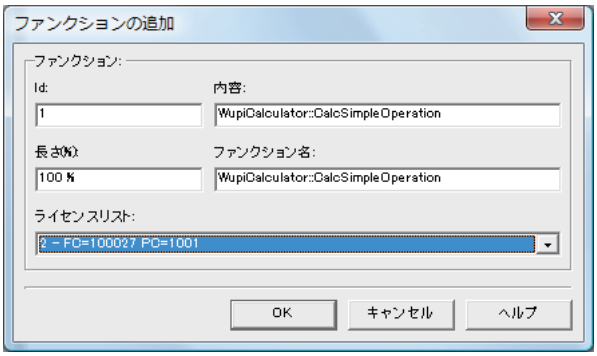
2. ファンクションとライセンスリストを割り当てる

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。



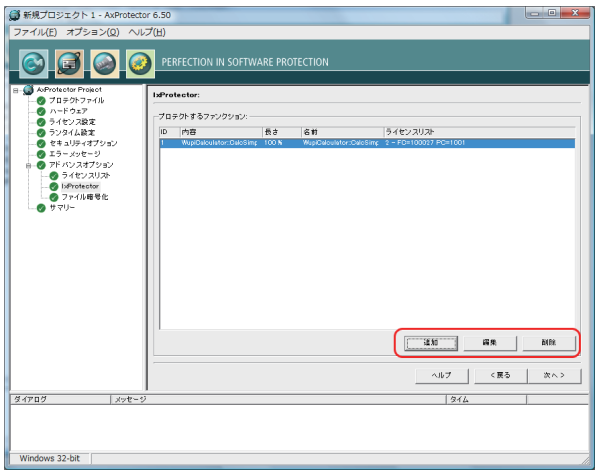
① ファンクション定義を行う

「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、

WupiEncyptCode(1)

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

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

6-5. WUPI ファンクション詳細

WupiAllocateLicense

[内容]
指定したライセンスリストのライセンスを割り当てます。

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

[Syntax]
int WupiAllocateLicense (int iLicenseList);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。

[リターン値]
エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiFreeLicense

[内容]
割り当てたライセンスを解放します。

[Syntax]
int WupiFreeLicense (int iLicenseList);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。

[リターン値]
エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiGetHandle

[内容]
エントリのネイティブハンドルを返します。
CodeMeterの場合はHCMSysEntryに、WIBU-KEYの場合はHWKBENTRYに指定されます。

[Syntax]
WupiGetHandle (int iLicenseList);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。

[リターン値]
エラーの場合は0を返します。

WupiEncryptionCode

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

[Syntax]
int WupiEncryptCode (int iFunction);

[パラメータ]
iFunction
ファンクションインデックスを参照します。

[リターン値]
エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiDecryptCode

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

[Syntax]
int WupiDecryptCode (int iFunction);

[パラメータ]
iFunction
ファンクションインデックスを参照します。

[リターン値]
エラーの場合はFALSE(0)、それ以外はTRUE(1)を返します。

WupiDecreaseUnitCounter

[内容]
CodeMeterの場合はUnit Counter（ユニットカウンター）、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。

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。

WupiGetLicenseType

[内容]
ライセンスタイプを返します。

[Syntax]
int WupiGetLicenseType (int iLicenseList);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。

[リターン値]
UpiLicenseNone
ライセンスタイプは内部で設定されている。
UpiLicenseWibuKey
ライセンスタイプはWIBU-KEY。
UpiLicenseCodeMeter
ライセンスタイプはCodeMeter。
UpiLicenseCodeMeterAct
ライセンスタイプはCodeMeterAct。

WupiCheckDebugger

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

[Syntax]
int WupiCheckDebugger (int iLicenseList, int nLevel);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。
nLevel
アンチデバッグレベルを指定します。

Level 0: デバッガーチェックをしない。
Level 1: 簡易デバッガーチェック。
Level 2: カーネルデバッガーチェック。
Level 4: アドバンスデバッガーチェック。
Level 8: IDEデバッガーチェック。デバッガーが完全に使用できなくなる。
Level 16: デバッガーを検知するとCodeMeterまたはWIBU-KEYのエントリーを無効化（アクセス不可）する。
Level 64: 仮想マシン上での動作をチェックする。プロテクトされたアプリケーションが仮想マシン上で動作することを禁止する。

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

WupiCheckLicense

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

[Syntax]
int WupiCheckLicense (int iLicenseList);

[パラメータ]
iLicenseList
ライセンスリストインデックスを参照します。

[リターン値]
エラーの場合FALSE(0)を返し、それ以外はTRUE(1)を返します。

WupiGetLastError

[内容]
直前に実行したWUPIファンクションのエラーコードを返します。

[Syntax]
int WupiGetLastError();

[リターン値]
エラーコードを返します。

Error Codes

Errorcode: UpiErrorDebuggerDetected

デバッガーによる解析行為が検知されました。(-5)

Errorcode: UpiErrorFunctionNotFound

指定されたファンクションポイントが見つかりませんでした。エラー(-3)

Errorcode: UpiErrorInfoNotAvailable

WupiQueryInfoファンクションで取得した情報は無効です。(-7)

Errorcode: UpiErrorLicenseModuleNotLoaded

ライセンスモジュールがロードできません。(-6)
このエラーは、WIBU-KEYの場合WkWin32/64.dll、CodeMeter/CodeMeterActの場合WibuCm32/64.dllが見つからないか、またはインストールされていない場合に発生します。

Errorcode: UpiErrorLicenseNotFound

指定されたライセンスが見つかりませんでした。エラー(-2)

Errorcode: UpiErrorNoDefaultLicense

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

Errorcode: UpiErrorNoError

エラーは発生しませんでした。(0)

Errorcode: UpiErrorRuntimeTooOld

インストールされているランタイムは古いバージョンです。エラー(-4)

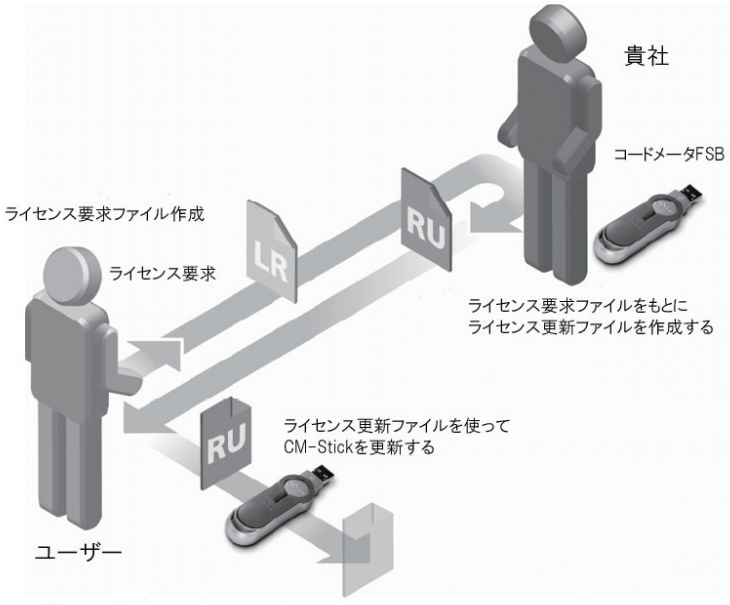
Chapter 7

リモートアップデート機能について

7-1. 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。

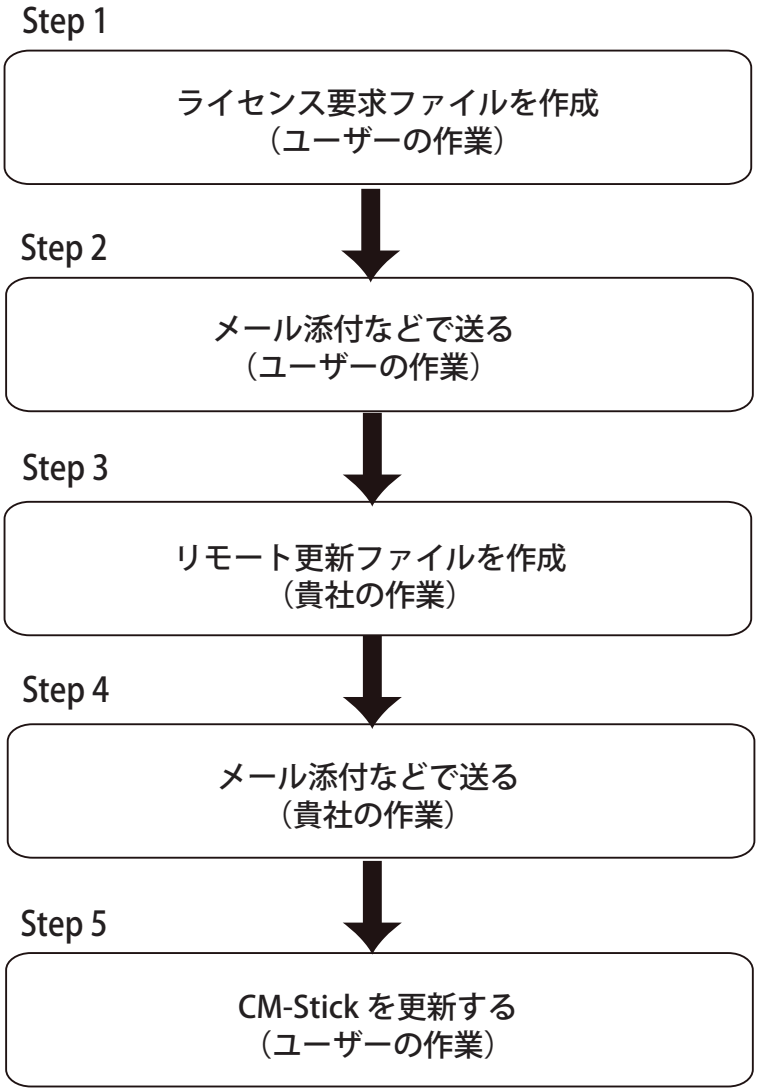


ユーザーに配布したライセンス更新ファイルは指定したコードメータキー（CM-Stick)に対して１回だけ使用可能です。同じコードメータキー（CM-Stick)に対して２回以上使用したり、別のコードメータキー（CM-Stick)に使用することができないため、確実にセキュリティを確保できます。

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。

7-2. 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。



7-3. ライセンス要求ファイルの作成（ユーザー側）

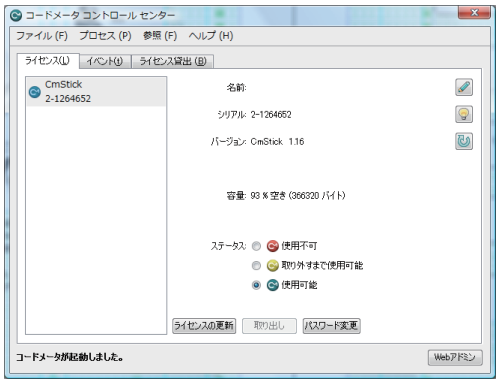
ライセンス要求ファイルは、コードメータコントロールセンターから作成します。

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。

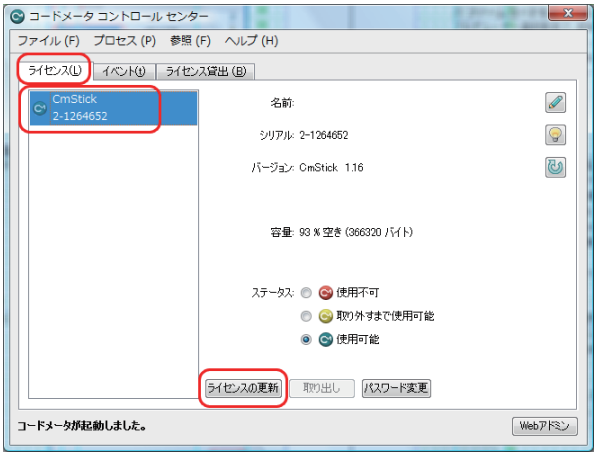
② コードメータコントロールセンターを開く

コードメータコントロールセンターを開きます。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。



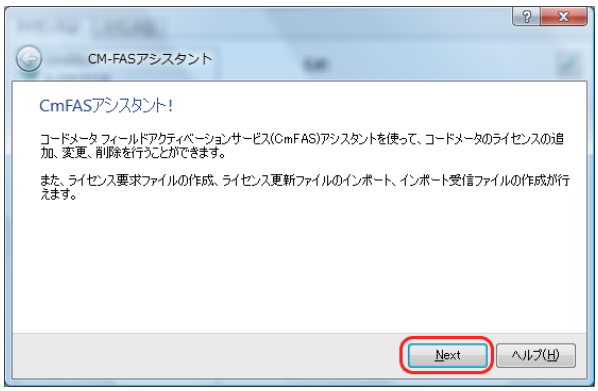
③「ライセンスの更新」をクリックする

「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。



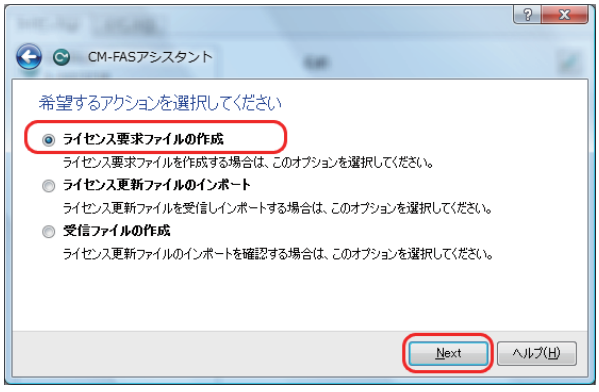
④ CM-FAS アシスタント画面が表示される

「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

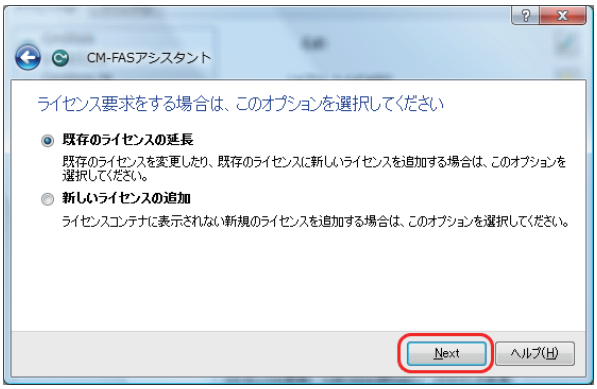
希望するアクションを選択する画面で、「ライセンス要求ファイルの作成」を選択します。



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。

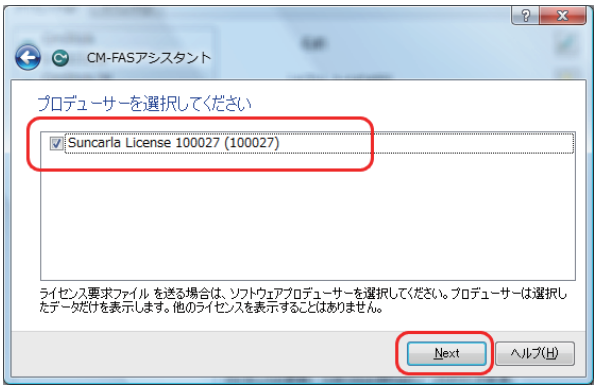
「新しいライセンスの追加」とは、まだ登録されていない新規のファームコードを追加するオプションです。

[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

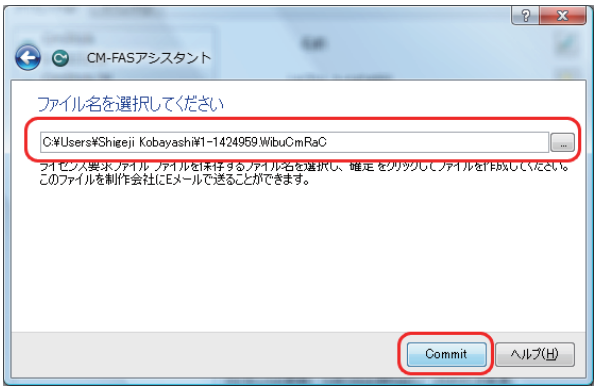
プロデューサー選択画面で、更新を要求するファームコードを選択し、「Next」ボタンをクリックします。異なるファームコードが複数表示されている場合は、更新を要求するファームコードを選択してください。



⑧ 保存先を指定する

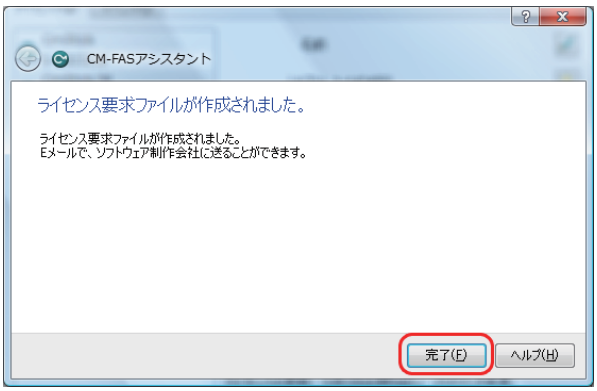
ライセンス要求ファイルの名前と保存先を指定します。ライセンス要求ファイルの拡張子は ".WibuCmRaC"です。拡張子は変更しないでください。また、ファイル名の左部には、CM-Stickのシリアル番号が表示されます。ファイル名は、デフォルトのままご使用されることをお勧め致します。

設定後、「Commit」ボタンをクリックします。



⑨ ライセンス要求ファイルが作成される

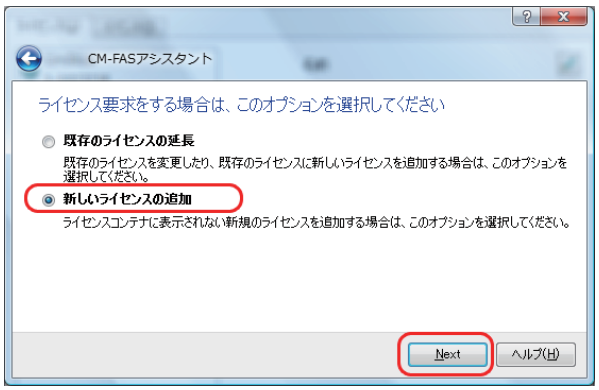
"**⑧保存先を指定する**"で指定したフォルダにライセンス要求ファイルが作成されます。エクスプローラを使って確認してください。



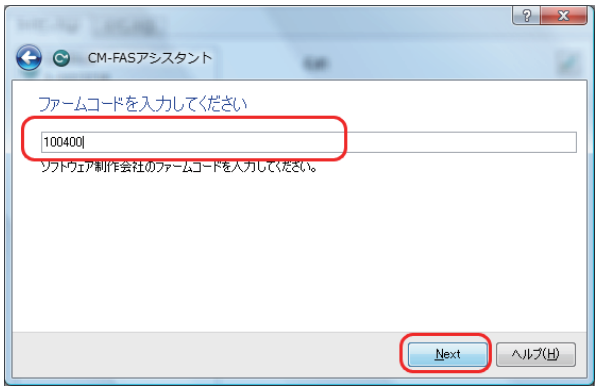
「完了」ボタンをクリックすると、コードメータコントロールセンター画面に戻ります。

▷ 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。

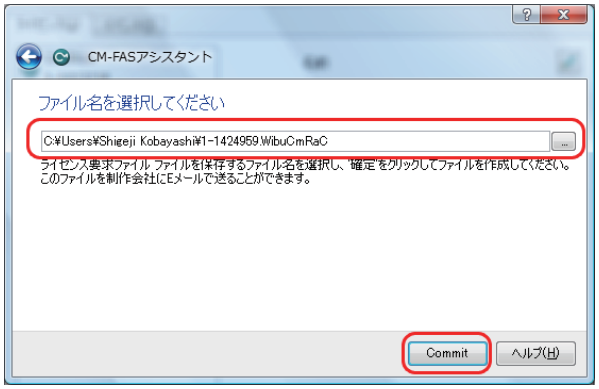


ファームコード入力画面で、要求するファームコードを半角数字で入力し、「Next」ボタンをクリックします。(ここでは、ファームコード= 100400を要求)



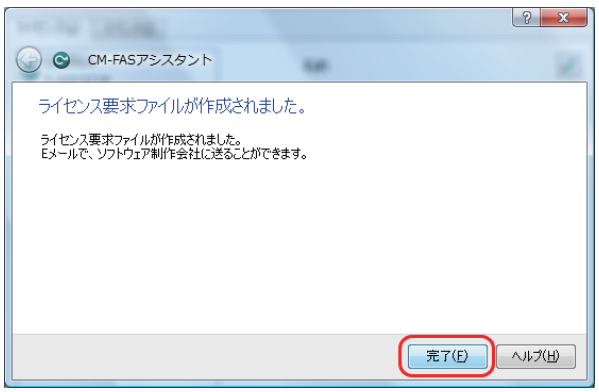
ライセンス要求ファイルの名前と保存先を指定します。ライセンス要求ファイルの拡張子は".WibuCmRaC"です。拡張子は変更しないでください。また、ファイル名の左部には、CM-Stickのシリアル番号が表示されます。ファイル名は、デフォルトのままご使用されることをお勧め致します。

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。



7-4. 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。

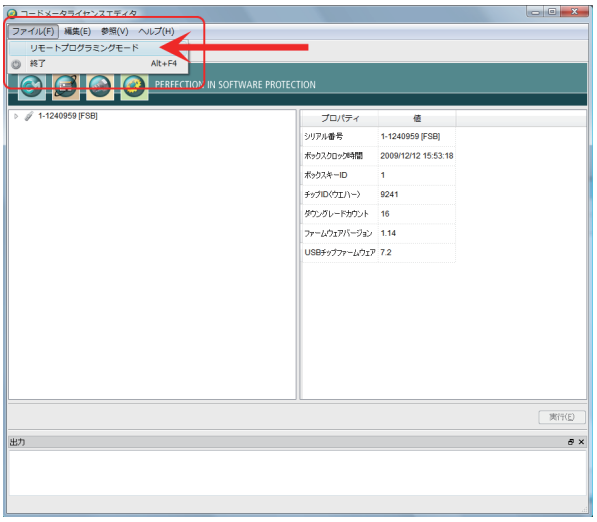
- コードメータライセンスエディタを使用する方法
- CmBoxPgm.exeを使用する方法（コマンドライン環境）

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。

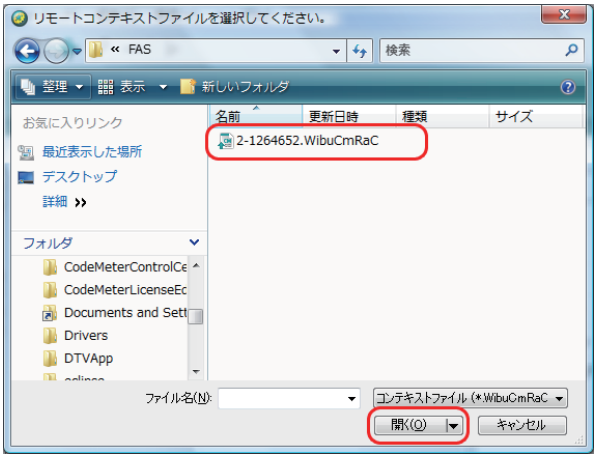


② コードメータライセンスエディタを起動

【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。



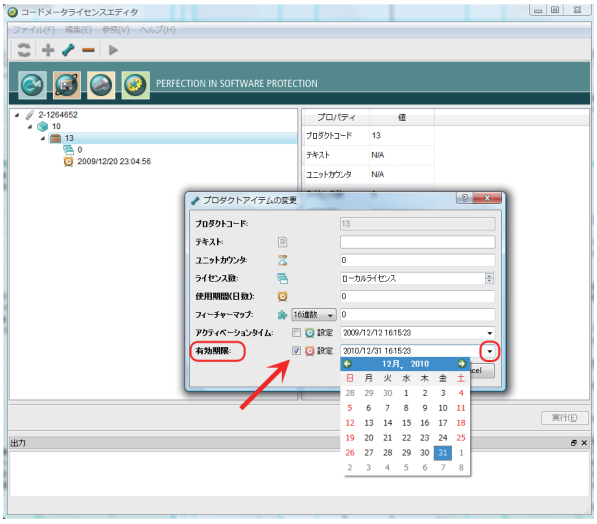
④ ライセンス要求ファイルの内容を確認

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。

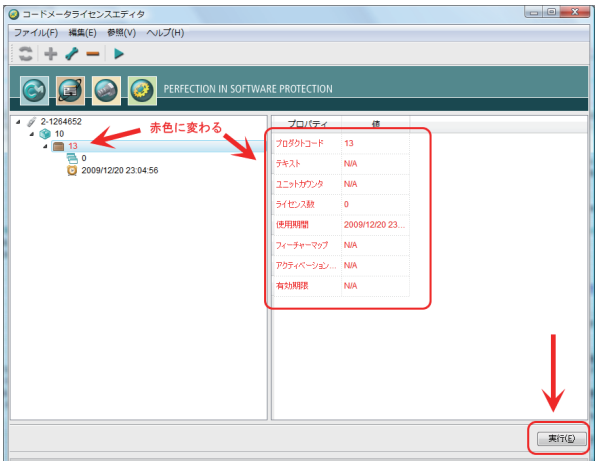
⑤ ライセンス更新ファイルを作成する

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。

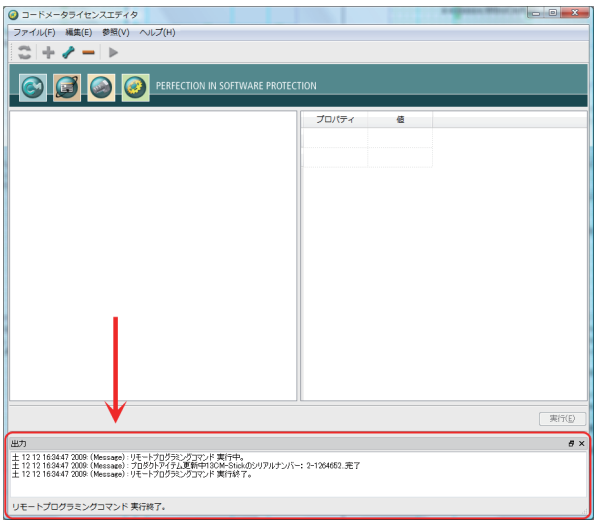
プロダクトアイテムの変更画面で更新する内容を編集します。ここでは、使用有効期限を2010年12月31日に変更します。



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。

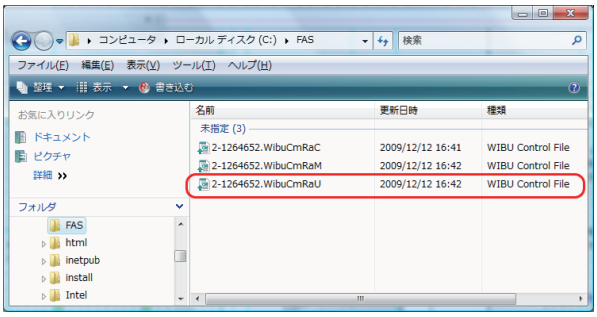


⑥ ライセンス更新ファイルを確認する

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。

7-5. CM-Stick を更新する（ユーザー側）

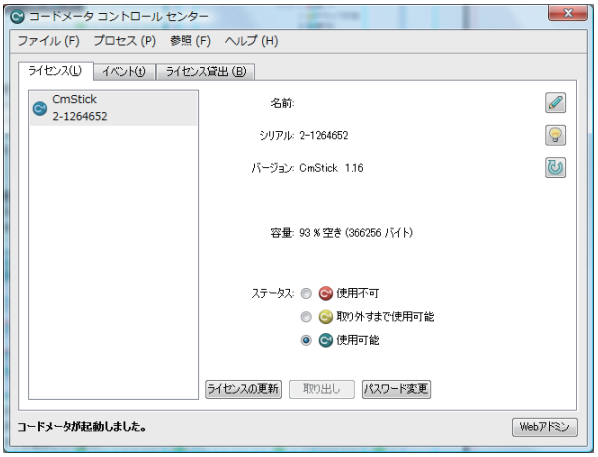
ライセンス更新ファイル"2-1264652. WibuCmRaU"を使って、CM-Stickの内容を更新します。CM-Stickの更新は、コードメータコントロールセンターを使用します。

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。

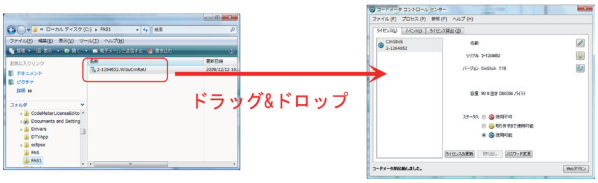
② コードメータコントロールセンターを開く

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。



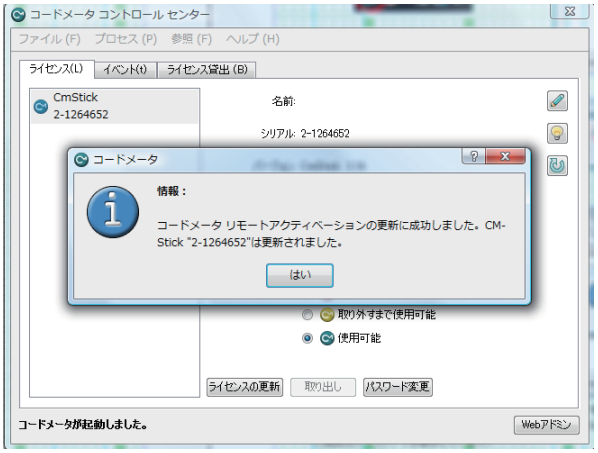
③ ライセンス更新ファイルをドラッグする

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。



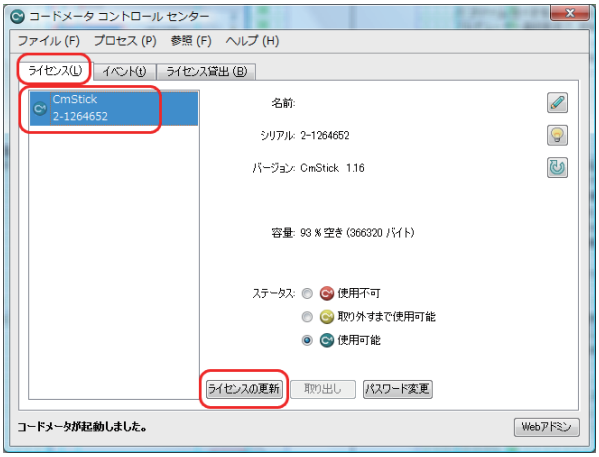
④ CM-Stick が更新された

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

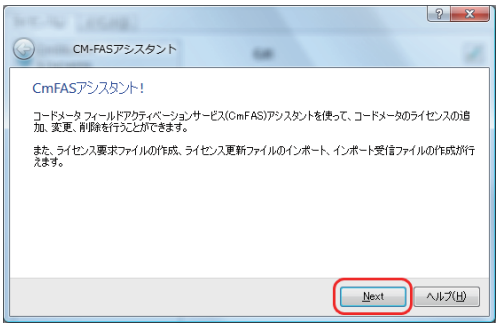
装着するCM-Stickは、ライセンス要求ファイルを作成したときのCM-Stickである必要があります。別のCM-Stickを装着してもシリアル番号が一致しないため更新作業は行われません。

② コードメータコントロールセンターを開く

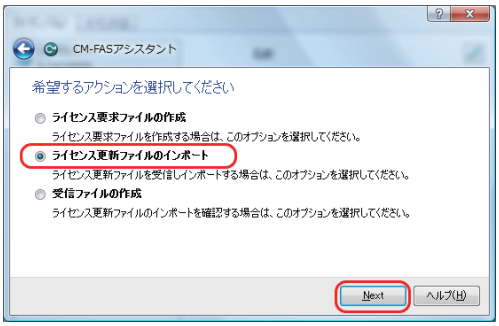
画面右下のコードメータコントロールセンターのアイコンをクリックして、コードメータコントロールセンターを開きます。アイコンが見つからない場合は、【スタート】→【すべてのプログラム】→【CodeMeter】→【CodeMeter Control Center】から起動します

③「ライセンスの更新」ボタンをクリックする

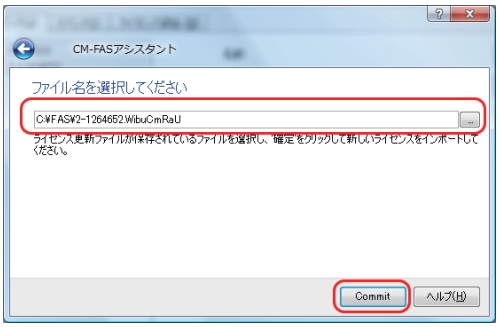
「ライセンスの更新」ボタンをクリックします。
CM-FASアシスタントが起動しますので、メッセージに従って進めます。



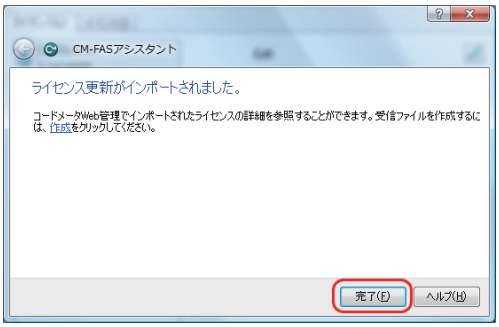
「ライセンス更新ファイルのインポート」を選択し、"Next"ボタンをクリックします。



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。

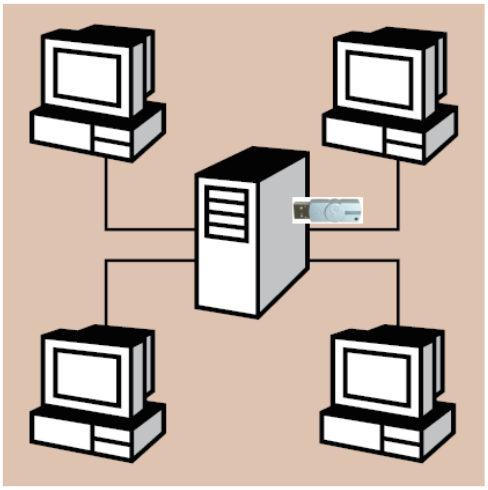


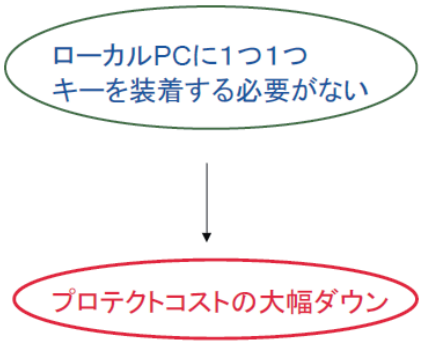
Chapter 8

ネットワーク機能について

8-1. 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。

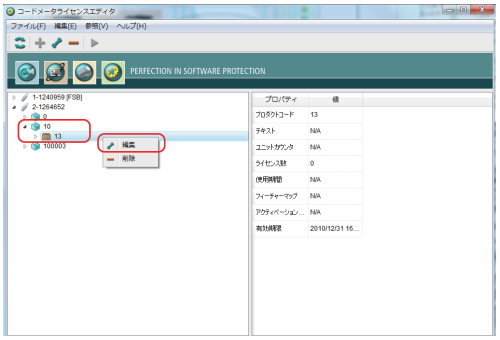
8-2. ネットワークカウンターの登録方法

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

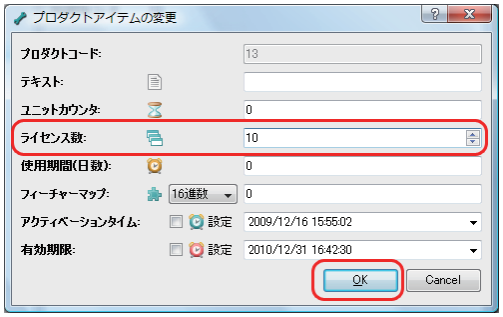
① CM-Stick と CM-FSB を PC に装着する

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。

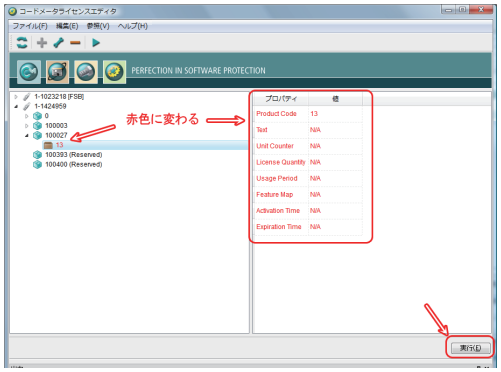


③「ライセンス数」を設定

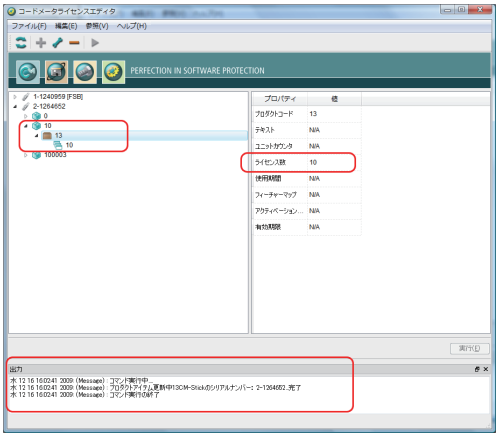
「プロダクトアイテムの変更」画面で、「ライセンス数」に許可するネットワークライセンス数を入力します。ここでは、10を入力し、OKボタンをクリックします。

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。

8-3. コードメータサーバーの起動方法

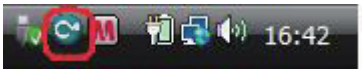
コードメータサーバーは、WebAdminのネットワーク設定から行います。コードメータランタイムキットをインストールするとこの機能も自動的にインストールされます。コードメータサーバーは、必ずしもネットワークを実際に管理するサーバーを指定する必要がなく、クライアントPCの１台をコードメータサーバーにすることが可能です。どれをサーバーにするかは、アクセス負荷を考慮の上、指定してください。

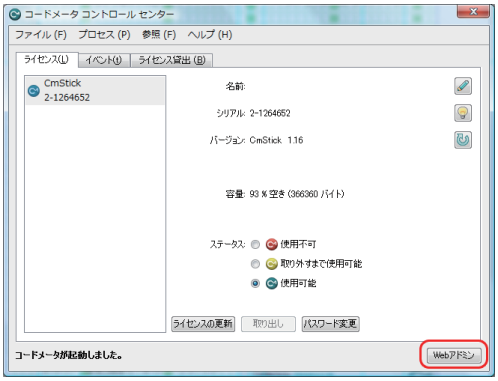
① コードメータサーバーに CM-Stick を装着します。

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。

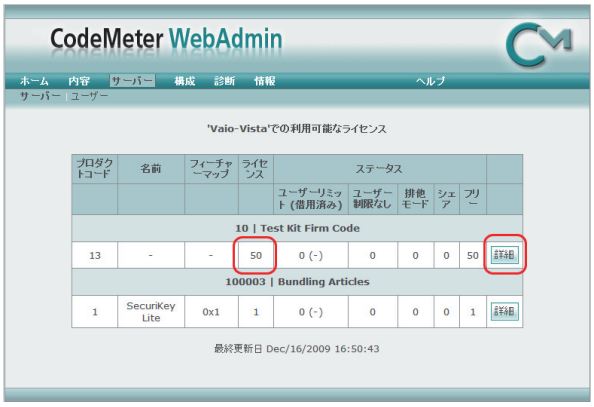
③ネットワークサーバーを設定する。

「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。



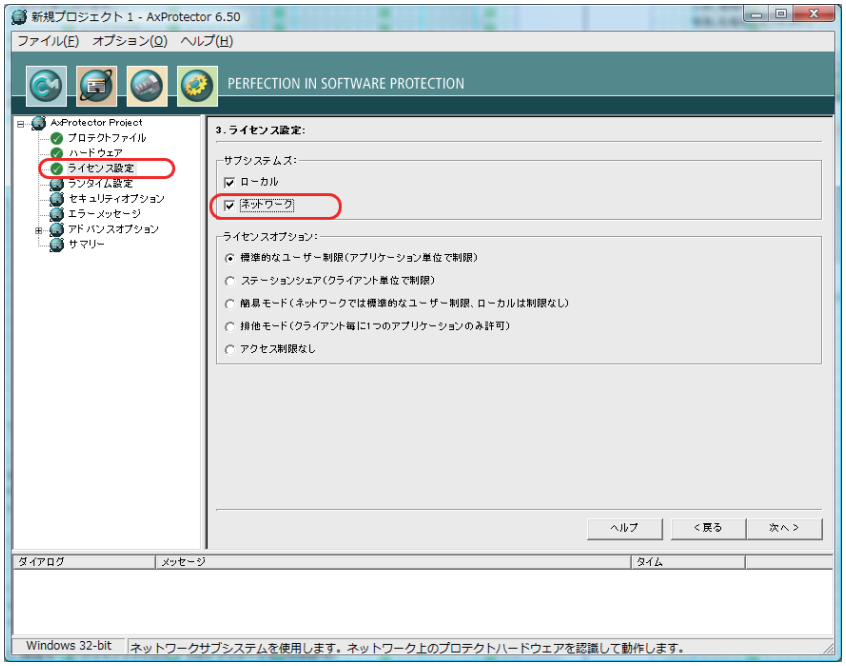
8-4. ネットワーク対応型プロテクトの作成方法

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「3. 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「4. 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「3. 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、ニーズ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。

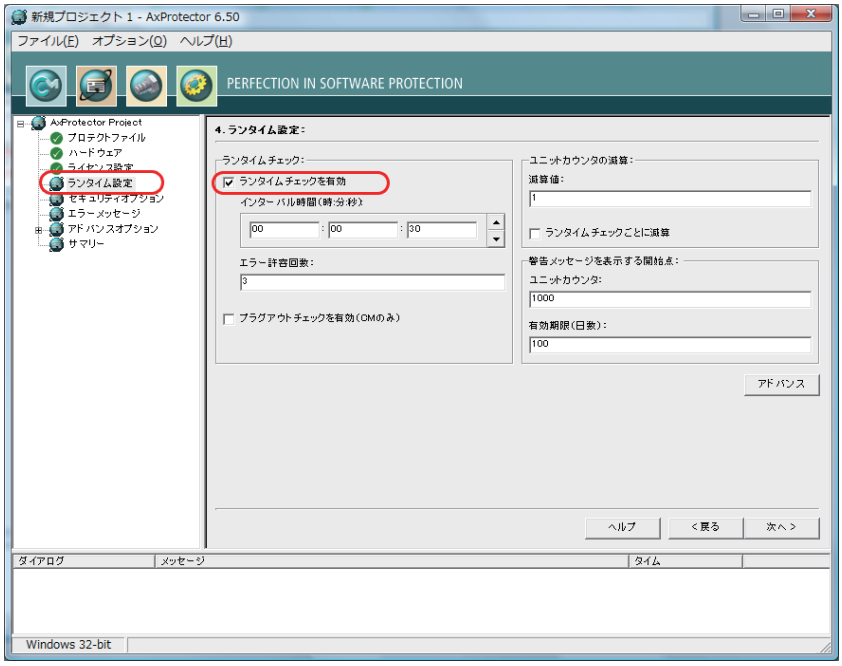
○ 簡易モード（ネットワークでは標準的なユーザー制限、ローカルは制限なし）
ネットワーク上のコードメータキーに対しては「標準的なユーザー制限」として動作しますが、ローカルマシン上のコードメータキーに対しては制限がありません。

○ 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かれば、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「4. ランタイム設定」で「ランタイムチェックを有効」にチェックを入れます。この項目を指定しないと、アプリケーションが起動したあとにサーバークラスタ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 9

コードメータ アイデンティティ（Web 認証）について

9-1. コードメータアイデンティティ (CmIdentity) とは

コードメータアイデンティティ(CmIdentity)とは、CM-Stickを使ってウェブサーバーへのアクセス認証（Web認証）を行う機能です。CM-Stickというハードウェアデバイスを利用するため、単なるIDとパスワードによるアクセス認証よりも、よりセキュアな認証が実現できます。

コードメータアイデンティティは、次のようなニーズに役立ちます。
▷特定のWebサイトへアクセスできるユーザーを限定化したい。
▷SaaSやASPなどのクライアントライセンスを確実に制限管理したい。

9-2. コードメータアイデンティティの優れた点

コードメータアイデンティティは、スマートカードやUSBトークンなどの一般的なハードウェア認証デバイスとは異なり、コードメータアイデンティティ独自の優れた機能を有しています。一般的なWeb認証との違いとコードメータアイデンティティの優位性をご説明します。

ユーザー ID とパスワード認証との違い

従来のWebサイトとソフトウェアは、主にユーザーIDとパスワードによって保護されてきました。この方法は構築が簡単で、全てのWebブラウザに対応し、サービスサイトに簡単に実装することが可能です。

しかし、ユーザーIDとパスワードを使ったやり方には多くの欠点があります。
一例として・・・
・ユーザーIDとパスワードの組み合わせは、簡単にコピーされて異なるクライアントからも使われます。従って、実際には個々にライセンスを購入する必要があるにもかかわらず1つのライセンスを不正に共有（使い回し）することができます。
・単純なパスワードは簡単に破られます。逆に有用なパスワードは複雑で覚えておくのが困難です。また、アクセスのたびにキーボードからパスワードを入力する作業は、ユーザーにとって負担となります。
・サーバーサイトではアクセスの権限を細かく解析し制御しなければなりません。例えば、1 年間に限定したアクセス許可を与える場合、あるいは一定の回数だけアクセスを許可する場合、というような機能を複合的に実現するためには、サービス側で膨大な利用者データを用意する必要があります。

コードメータアイデンティティの場合、アクセスのベースとなるCM-Stickハードウェアは複製することができません。ユーザーは空のCM-Stickを購入することはできますが、固有のライセンス情報を設定できるのはコードメータFSBを所有している開発会社だけです。

また、コードメータアイデンティティは標準的なウェブサイトに入力する特有のパスワードを必要としません。ユーザーが知っていなければならない唯一のパスワードは、CM-Stickそのものの使用を禁止／許可する時に必要となるパスワード（PINコード）だけです。このパスワードはハードウェアに基づいており、シンプルでありながら高いセキュリティ強度を備えています（例えば4～6桁の数字でさえすでに比較的安全なレベルにあります）。

サーバーサイトへの細かなアクセス権限はCM-Stickによってコントロールされます。従い、サーバーサイトではアクセス許可のための膨大な利用者データを用意する必要はありません。

ソフトウェアベースの認証キーとの違い

ユーザーIDとパスワードの代わりに、ActiveXやJavaプラグインを使ったクライアント用セキュリティソフトウェアでアクセスする方法があります。それらは、単純なキー認証から複雑な認証スキーマまで構築することができます。

ただし、この方式の問題点は、そのようなソフトウェアは、逆アセンブラ、リバースエンジニアリング、デバッギング等によって簡単に解析されてしまうことです。ハッカーは、クライアント認証の手続きを見破って、ハッカー自身のクライアントソフトで正規のクライアントソフトであるかのように振る舞います。

コードメータアイデンティティで使用するCM-Stickは、全ての認証スキーマをハードウェアに実装しています。シークレットキーによる対称暗号、あるいはパブリックキーによる非対称暗号を利用することができますが、いずれのキーもハードウェアに残らないため、逆アセンブラ、リバースエンジニアリング、デバッギングなどによって見つけることはできません。

一般的なハードウェアトークンとの違い

USBドングル、スマートカード、あるいはPCカードのようなハードウェアデバイスにWeb認証のための秘密鍵や公開鍵などのセキュリティ情報を格納し、Webサイトへのアクセスや、Webベースのサービスを制御するハードウェアトークンが存在します。コードメータはすでにこのような機能を搭載しています。さらに、一般的なハードウェアトークンとは異なる次のような追加機能も搭載しています。

追加機能として、

使用有効期限の設定
WebサイトまたはWebサービスにアクセスできる有効期限を設定できます。

使用開始期日の設定
いつからWebサイトまたはWebサービスにアクセスできるかの使用開始期日を設定できます。

使用期間の設定
WebサイトまたはWebサービスが何日間アクセス（使用）可能かの使用期間を設定できます。
（例：30日間有効など）

使用回数の設定
WebサイトまたはWebサービスに何回までアクセスができるかという使用回数を設定できます。
設定できる範囲は1～100万回です。

プロテクトされたプログラムの実行
Webサービスでなく、コードメータでプロテクト処理されたアプリケーションプログラムも同時にプロテクト管理できます。（Web認証とコピープロテクトが同時に実現できる）

更に、これらの機能は、

- サーバー側で管理するのでなく、キー(CM-Stick)をユーザーごとに作り分けることで実現できる。
- １つのWebサイトまたはWebサービスに対して、複数の機能を同時に設定することが可能です。
- 1つのCM-Stickで、数千の異なるWebサイトおよびWebサービスのアクセス認証を確実に行うことが可能です。
- 設定された内容は、Webサイトからリモートで更新することが可能です。

9-3. コードメータアイデンティティのシステム構築

WindowsサーバーとASP.NET
Apache(Windows)
Apache(Linux)

システム構築のサンプルプログラムがコードメータCDのCmIdentityフォルダに格納されています。

9-4. Windows2003 と ASP.NET による構築

IIS と ASP の設定

1.［スタート］-［プログラム(P)］-［管理ツール］-［サーバーの構成ウィザード］（または、「サーバーの役割管理」の「役割を追加または削除する」）で、サーバーの構成ウィザードを起動します。

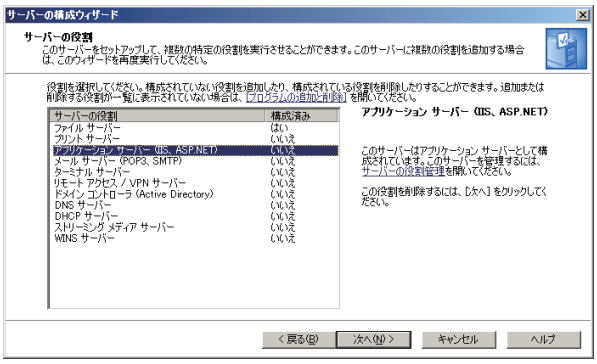
2.「サーバーの構成ウィザードの開始」の画面が表示されたら[次へ(N)]ボタンで先に進めます。

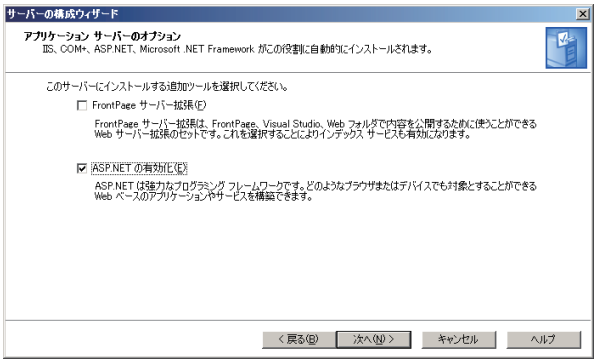
3.「準備作業」の画面が表示されますので必要とされる周辺機器の設定等を確認してください。（Windows Serverのインストールが必要になりますのでドライブにセットしてください。）[次へ(N)]ボタンで先に進めます。

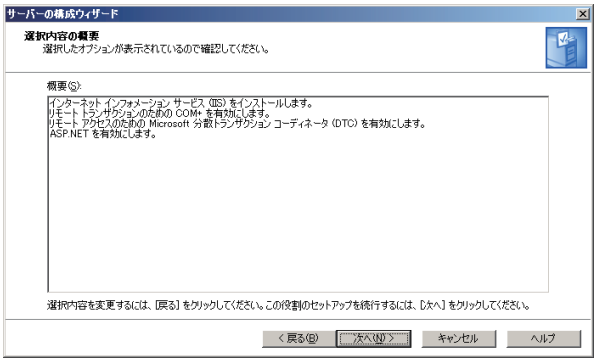
4.ネットワーク構成の確認が行われた後、「サーバの役割」の画面が表示されます。アプリケーションサーバー（IIS、ASP.NET）を選択して[次へ(N)]ボタンで進めます。

5.「アプリケーション サーバーのオプション」の画面が表示されたら、［ASP.NETの有効化(E)］にチェックを入れて[次へ(N)]ボタンで進めます。

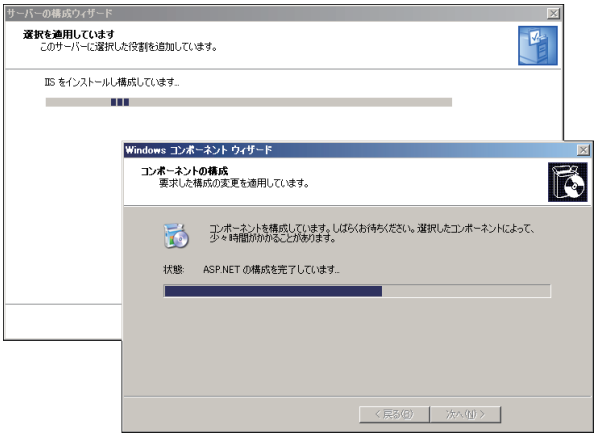
「選択内容の概要」の画面が表示されます。
確認後、[次へ(N)]ボタンで進めます。



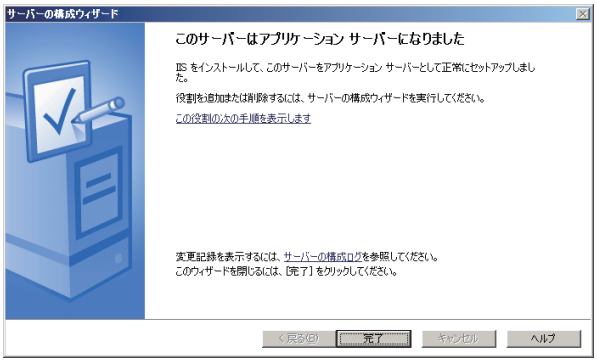




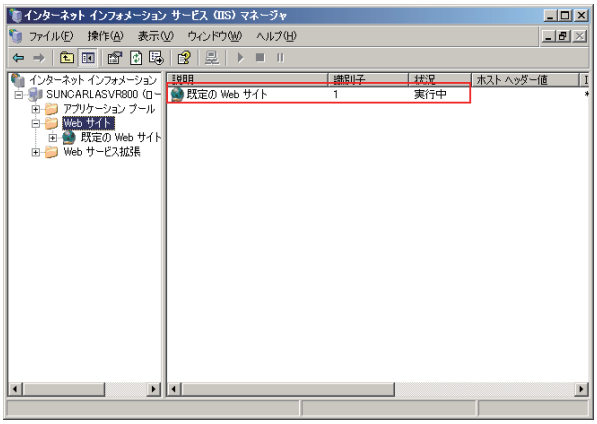
6.インストール作業が開始します。



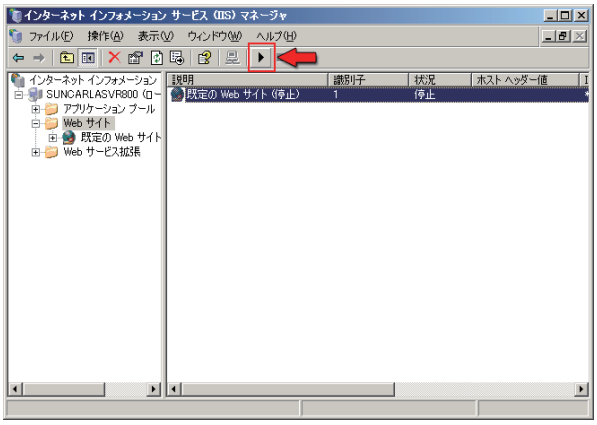
インストールが終了したら[完了]ボタンで終了してください。



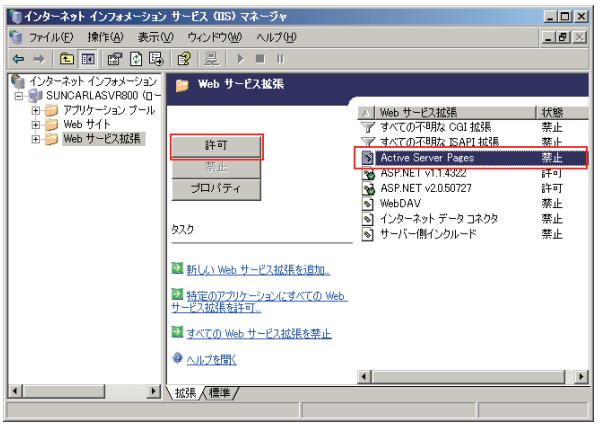
7.次に、IISの設定を行います。
[スタート]-[プログラム(P)]-[管理ツール]-[インターネット インフォメーション サービス(IIS)マーネージャ]でIISマネージャを起動して既定のWebサイトが実行中であることを確認してください。



停止している場合は、既定のWebサイトをクリックしてツールバーの「項目の開始」、または右クリックのコンテキストメニューの「開始(S)」で既定のWebサイトを実行してください。

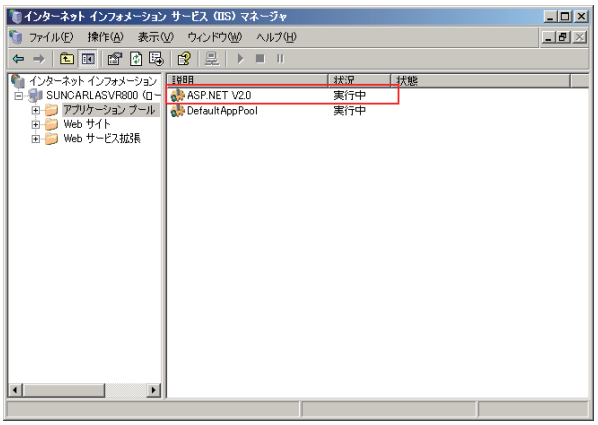


８. 次に、「Webサービス拡張」を選択して Active Server Page の状態を確認します。IISのデフォルト設定では Active Server Page が禁止に設定されています。Active Server Pageを選択して、[許可]ボタンで「許可」に設定してください。



最後に、アプリケーション プールで ASP.NETが実行されていることを確認します。

以上で、IISとASPの設定は完了です。



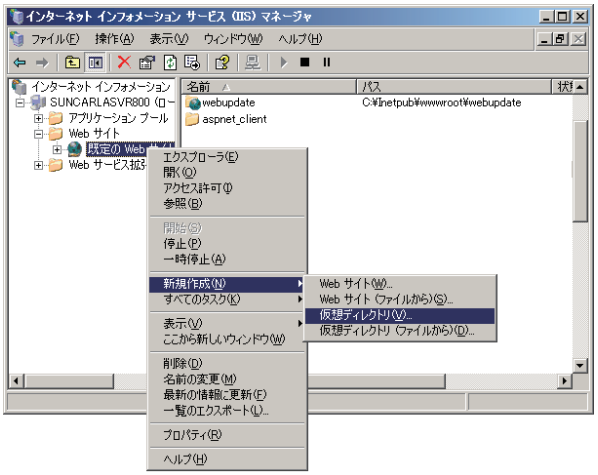
Web サーバーの設定

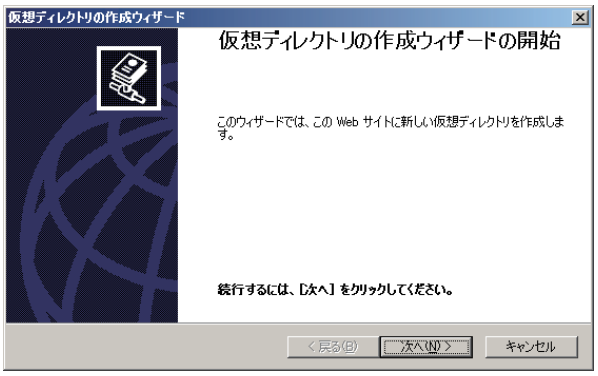
最初に、コードメータランタイムキットをWindows2003サーバーにインストールする必要があります。更に、サーバー側では、.NETでコードメータを使用する必要がありますので、CodeMeter.NETアセンブリ（"WibuCmNET.msi"）をインストールします。

Windows Server 2003のIISとASP用のデモファイルは <CmIdentityASPNET>フォルダの中にあります。サーバーのハードディスク上の適当な場所に <CmIdentityASPNET>フォルダの内容を全てコピーしてください。例では、Dドライブのルートに <CmIdentityASPNET>フォルダを作成してコピーするものとします。(D:\CmIdentityASPNET)

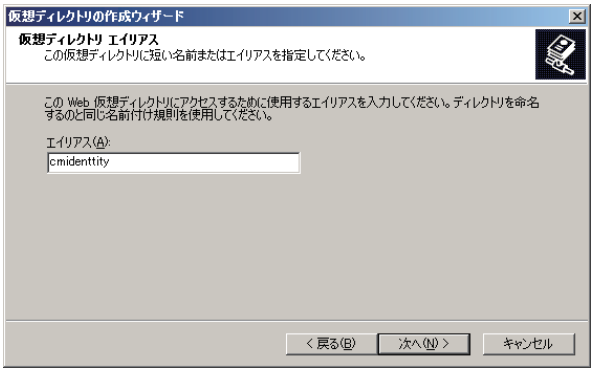
次に、IISの既定のサイトに <CmIdentityASPNET>フォルダを仮想ディレクトリとして登録します。

インターネットインフォメーションサービス(IIS)マネージャを起動して、「既定のWebサイト」を右クリックして、コンテキストメニューの「新規作成(N)」-「仮想ディレクトリ(V)」を選択してください。
仮想ディレクトリの作成ウィザードが起動します。
[次へ(N)]ボタンで先に進めてください。

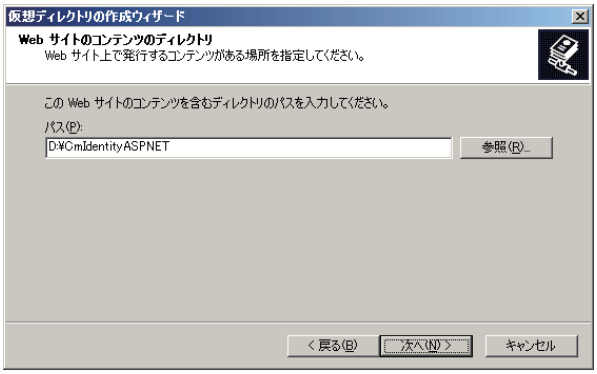




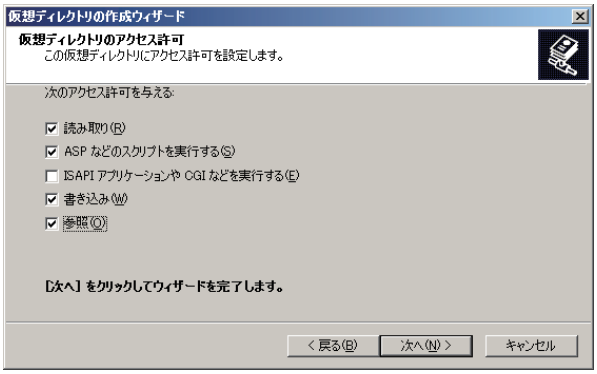
「仮想ディレクトリのエイリアス」画面が表示されます。エイリアスはブラウザから表示する時に指定するサイトの名前になります。ハードディスク上のフォルダ名と同じでなくても構いません。エイリアスを設定して[次へ(N)]ボタンを押してください。(例では、エイリアスを cmidentity とします。)



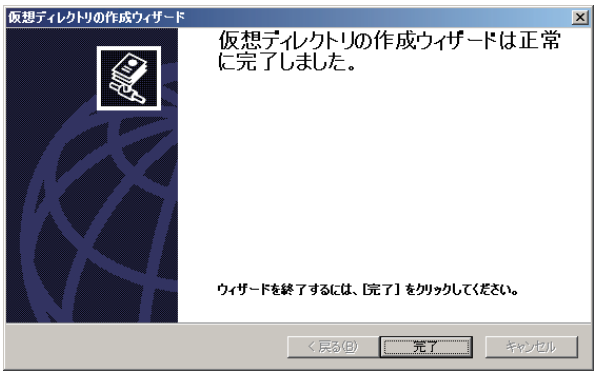
「Webサイトのコンテンツのディレクトリ」画面が表示されます。ハードディスクに保存した<CmIdentityASPNET>フォルダのパスを指定して[次へ(N)]ボタンを押してください。(例では、D:\CmIdentityASPNETです。)



「仮想ディレクトリのアクセス許可」画面が表示されます。読み取り(R)、ASPなどのスクリプトを実行する(S)、書き込み(W)、参照(O) を設定して[次へ(N)]ボタンを押してください。

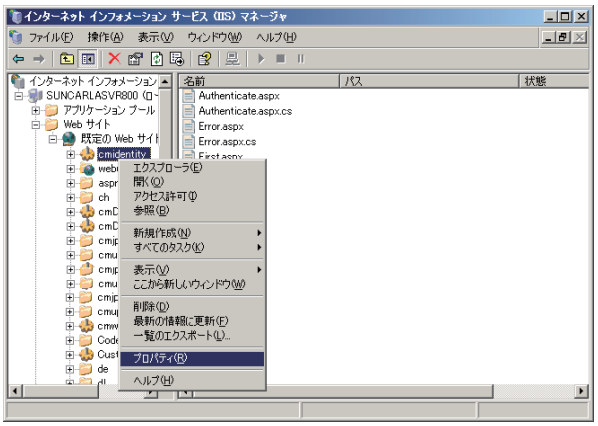


仮想ディレクトリの作成ウィザードの完了画面が表示されたら[完了]ボタンで作業を完了してください。

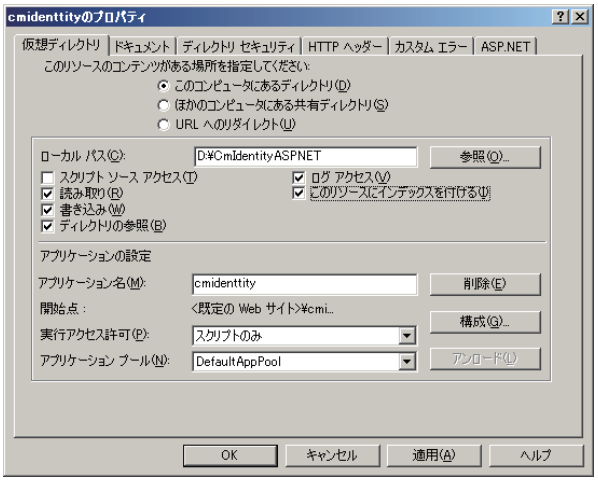


続いて仮想ディレクトリの「アプリケーションの設定」を行います。既定のWebサイトに上記で作成した仮想ディレクトリが追加されているはずです。
(例ではエイリアスに設定したcmidentity が表示されます)

追加した仮想ディレクトリを右クリックしてコンテキストメニューの「プロパティ」を選択してください。



仮想ディレクトリページのアプリケーションの設定を確認してください。



IISとASPの設定は以上です。
次に、コードメータアイデンティティー・デモサイトの設定を行います。
サーバーにコピーしたコードメータアイデンティティーのフォルダ（例では D:\CmIdentityASPNET ）に"Web.Config"ファイルがあります。これをテキストエディタで開きます。

以下のエレメントを検索してください。
　　<add key="LogPath" value="c:\"/>
これはコードメータアイデンティティのログ保存パスです。
IISが書込み許可を所有する有効なディレクトリを設定してください。（例. D:\CmIdentity_LOG ）
ログ保存フォルダが正しく設定されていないと最初のテストでエラーが表示されますのでご注意ください。

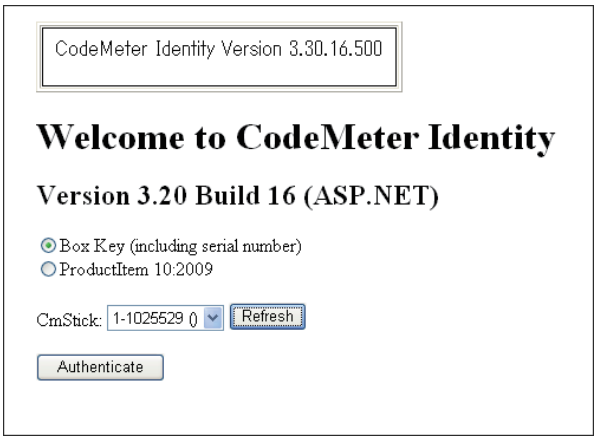
次に、以下のエレメントを検索してください。
<add key="FirmCode" value="10"/>
<add key="ProductCode" value="2009"/>
これはこのデモサイトで使用するCM-Stickのエントリ情報です。必要に応じて変更することができます。

次に、以下のエレメントを検索してください。
<add key="Codebase_ActiveX32" value="Download/WibuCmId32.cab#version=3,20,15,500"/>
<add key="Codebase_ActiveX64" value="Download/WibuCmId64.cab#version=3,20,15,500"/>
<add key="Codebase_JavaApplet" value="Download"/>
これらはクライアントサイトからの要求に応じてダウンロードされるコンポーネントです。
インストール先フォルダの <Download>フォルダにあります。

動作テスト

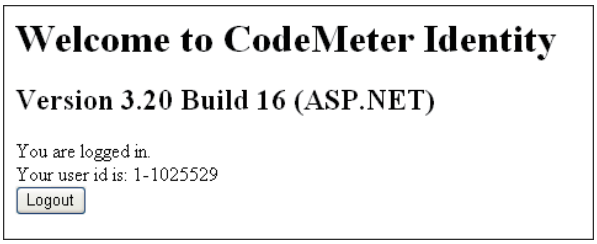
コードメータアイデンティティの動作テストを行います。

クライアントPCにCM-Stickを装着して、Webブラウザでhttp://<server name>/cmidentity/ にアクセスします。(例. http://localhost/cmidentity/)
右の画面が表示されます。



2つのモードが、あります:
認証方法の 1 つは、CM-Stickの内部キー(シリアルを含む)を使用します。
もう 1 つの方法は、ファームコードとプロダクトコードを使用します。
どちらの方法も試すことができます。

[Authenticate]ボタンをクリックしてCM-Stickを介した認証が正しく行われると、右のようなログインメッセージが表示されます。
Box Key によって認証を行った場合、Your user id is: にはCM-Stickのシリアル番号が表示されます。
ProductItem によって認証を行った場合、Your user id is: には ファームコードとプロダクトコードが表示されます。



9-5. Apache(Windows) による構築

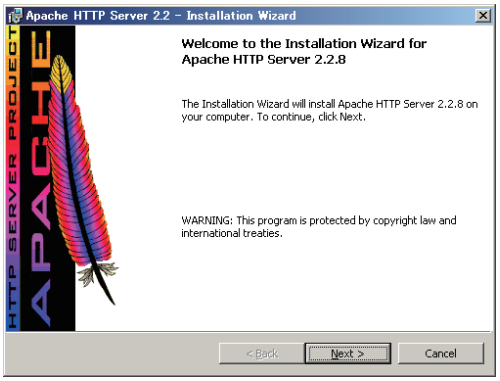
Windows上で稼働するApache上でコードメータアイデンティティのシステムを構築する説明です。

Apache のインストール

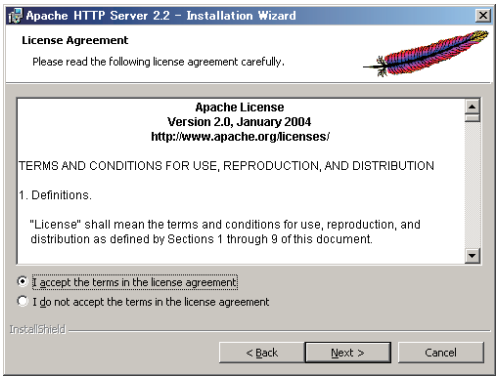
最初にWindows版Apacheをインストールします。すでに、Windows版Apacheがインストールされている場合は、次へ進んでください。尚、最新のApacheは、下記サイトからダウンロードできます。

http://www.apache.org/

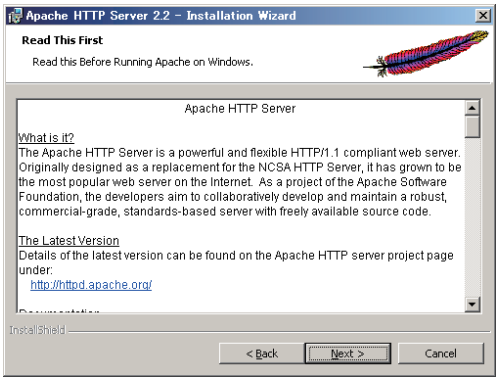
Apacheのインストーラを起動します。インストーラーの起動画面が表示されたら[Next]ボタンで進めてください。



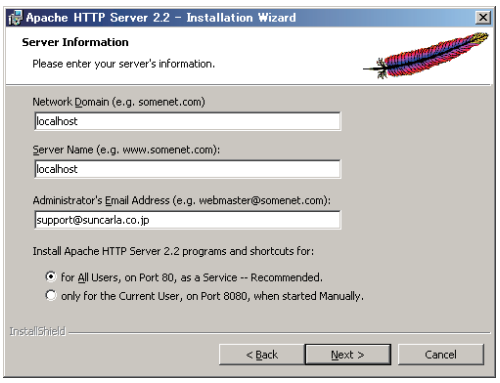
ライセンス契約の画面が表示されます。
I accept the terms in the license agreement をチェックして　[Next]ボタンで進めてください。



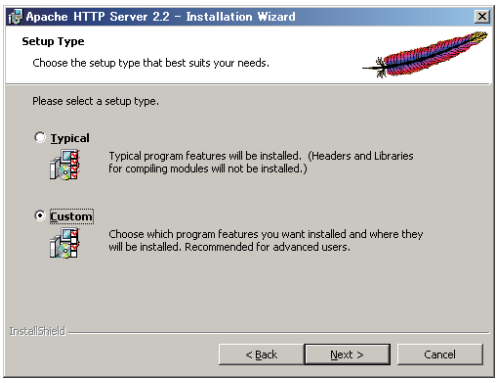
説明画面が表示されます。[Next]ボタンで進めてください。



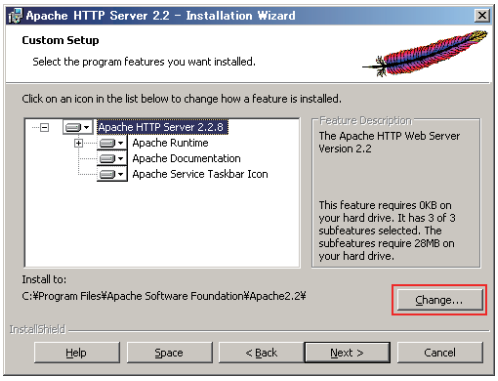
サーバー情報の設定画面が表示されます。
例では、Server Name にlocalhost を設定しています。また、HTTPサーバーのインストール方法はポート80でサービスとしてのインストールを選択しています。必要に応じて設定を変更してから[Next]ボタンで進めてください。



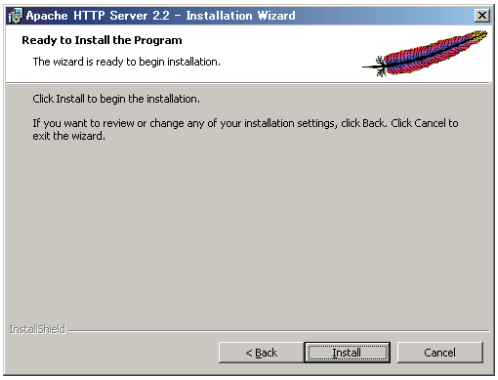
セットアップ方法の画面が表示されます。
例ではインストール先フォルダを変更できるCustom を選択しています。Typicalを選択すると標準的な設定で、Program Filesの下にインストールされます。いずれかを選択して[Next]ボタンで先に進めてください。



Custom を選択した場合、[Change...]ボタンで任意のインストール先フォルダを指定することができます。必要に応じてインストール先フォルダを設定してから [Next]ボタンで進めて下さい。

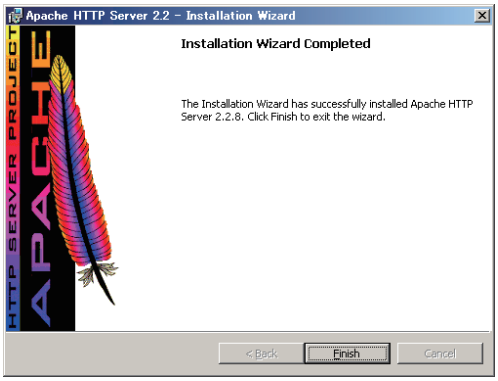


インストールの確認画面が表示されます。[Install]ボタンをクリックするとインストールが始まります。



インストールの終了画面が表示されればApacheのインストールは完了です。[Finish]ボタンで終了してください。

お使いのWebブラウザーで http://localhost/ にアクセスして、Apacheのトップページが正しく表示されることをご確認ください。



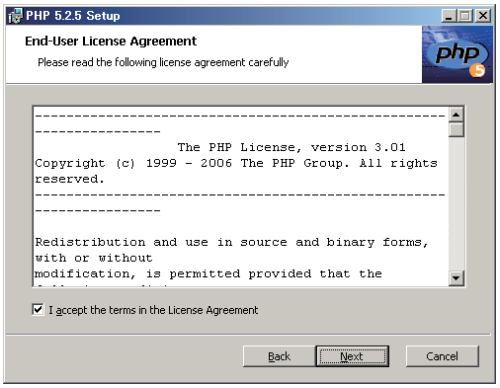
PHP のインストール

次に、PHPをインストールします。すでに、PHPがインストールされている場合は、次へ進んでください。
尚、最新のPHPは、下記サイトからダウンロードできます。

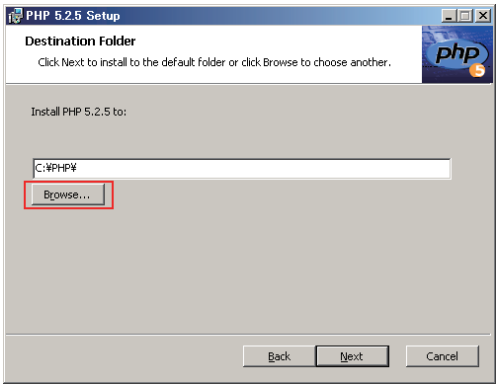
http://www.php.net/

PHPのインストーラを起動します。インストーラの起動画面が表示されたら[Next]ボタンで進めてください。

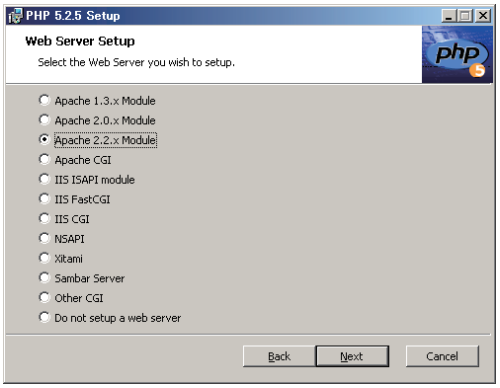
ライセンス契約の画面が表示されます。
I accept the terms in the License Agreement をチェックして[Next]ボタンで進めてください。



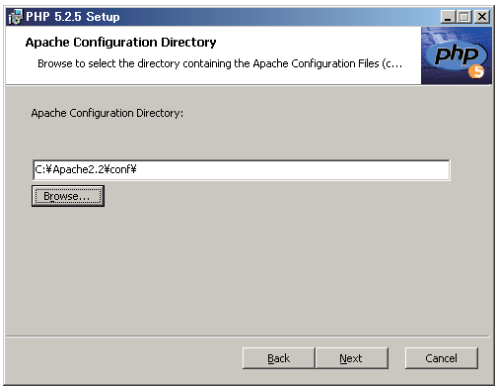
セットアップ先フォルダの設定画面が表示されます。[Browse...]ボタンでインストール先フォルダを変更することができます。例ではC:\PHP\ にインストールします。インストール先フォルダを設定して[Next]ボタンで進めてください。



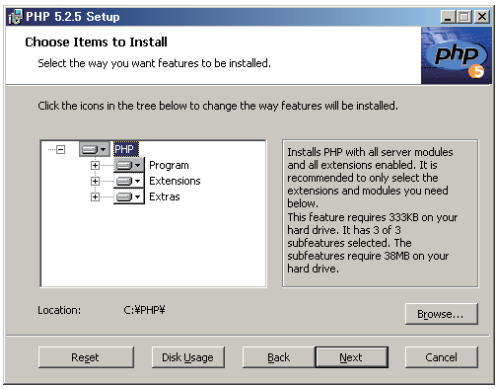
Web Serverの設定画面が表示されます。インストールしてあるApache HTTPサーバーに該当する項目をチェックして[Next]ボタンで進めてください。



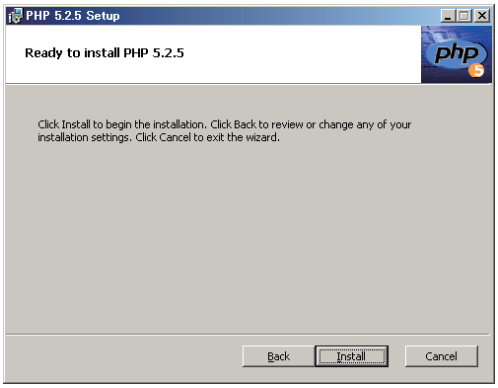
Apacheのコンフィグレーションフォルダの設定画面が表示されます。[Browse...]ボタンでApacheの"httpd.conf"ファイルがあるフォルダを設定してください。例では C:\Apache2\conf\ を指定しています。



インストールするアイテムの選択画面が表示されます。例では全てのアイテムをインストールしています。必要なアイテムを選択して[Next]ボタンで進めてください。



インストールの確認画面が表示されます。[Install]ボタンをクリックするとインストールが始まります。



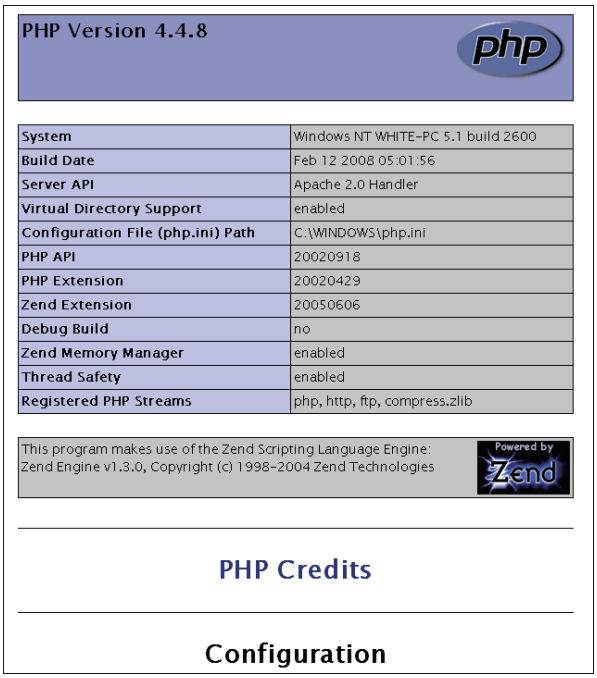
インストールの終了画面が表示されたら、PHPのインストールは完了です。 [Finish]ボタンで終了してください。

ApacheでPHPを使用するためにはこの後、PHPの "php.ini"ファイルと、Apacheの "httpd.conf" ファイルの設定を少し変更する必要があります。

変更する部分はPHPとApacheのバージョンによって異なりますので、お使いのPHP、Apacheの使用方法に従って設定してください。設定が完了したら、次の1行を記述した "test.php" というファイルを作成して Apacheディレクトリに置いてください。

【 test.php の内容】
<? phpinfo(); ?>

Webブラウザーで http://localhost/test.php にアクセスしてPHPが正常に機能していることを確認してください。右図のPHP情報画面が表示されればPHPは正常に機能しています。



コードメータアイデンティティのデモサイトの設定

最初に、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。これは、サーバー側とクライアント側の双方に必要です。

WindowsでのApacheとPHPのためのデモファイルは <CmIdentityPHP>フォルダに入っています。Apacheからアクセスが可能なフォルダに <CmIdentityPHP>フォルダの内容をコピーしてください。例では、C:\Apache2\htdoc\CmIdentity にコピーします。

C:\Apache2\htdoc\CmIdentity\　にコードメータアイデンティティの設定ファイル "Settings.php" があります。"Settings.php" にはオペレーティングシステムに依存する設定があります。これをWindowsプラットフォーム用に設定する必要があります。C:\Apache2\htdoc\CmIdentity\Settings.php をテキストエディタで開いてください。

```
<?php
// TODO: This file contains global variables and constants

// the path where all the files reside
$defaultpath     = "CmIdentity/";
// page to redirect if no target page is specified
$defaultpage     = $defaultpath."First.php";

// BoxKey authentication
$boxkeys = array("40f02de6d7131d9fb5a682d4848d19ef1941181d62bb45eaa
a・・・・・・00000000");

// ProductItem authentication
$firmcode            = 10;
$productcode         = 2009;
$encryptioncode      = 0;
$encryptioncodeoptions  = 0;
$publicfirmkey = "e48ff3222db2a0a965dcf66093b8352f889e07e826a0b244bb2
c・・・・・・00000000";

// Secret Data authentication (not yet implemented)
$secretdatatype      = 20;   // SD

$loglevel_client = 2; // (0=Errors, 1=Warnings, 2=Info, 3=Debug)

// File/Operating system specific settings
// (a) Mac OS X 10.5:
$java_exe       = "/usr/bin/java";
$server_jar     = "/Library/WebServer/Documents/CmIdentity/bin/WibuCmIdValidator.jar";

// (b) Windows
/*
$java_exe       = "C:\Program Files\Java\jre1.6.0_03\bin\java.exe";
$server_jar     = "C:\intetpub\wwwroot\CmIdentity\bin\WibuCmIdValidator.jar";
*/
```

```
//(c) Linux
/*
$java_exe       = "/usr/bin/java";
$server_jar     = "/srv/www/htdocs/CmIdentity/bin/WibuCmIdValidator.jar";
*/
?>
```

ファイル後半の　// File/Operating system specific settings　からがOSに依存する記述になります。
Windows以外をコメントアウトして、Javaの設定をお使いのPCの環境に合わせて設定しなおしてください。

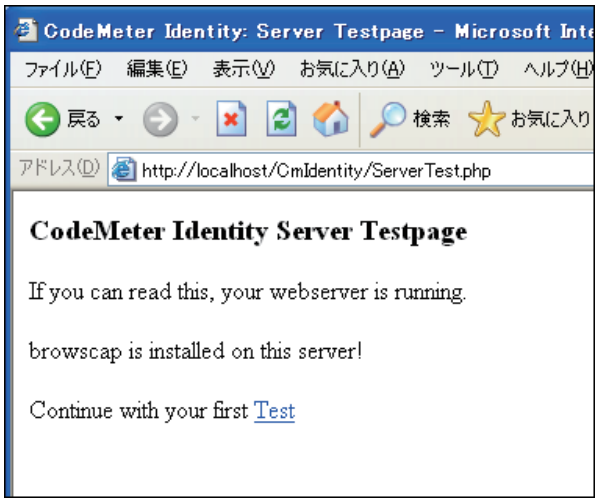
以下は設定例です。

```
// File/Operating system specific settings
// (a) Mac OS X 10.5:
// $java_exe       = "/usr/bin/java";
// $server_jar     = "/Library/WebServer/Documents/CmIdentity/bin/WibuCmIdValidator.jar";

// (b) Windows
/*
$java_exe       = "C:\Java\jre1.6.0_03\bin\java.exe";
$server_jar     = "C:\Apache2\htdocs\CmIdentity\bin\WibuCmIdValidator.jar";
*/

//(c) Linux
/*
// $java_exe       = "/usr/bin/java";
// $server_jar     = "/srv/www/htdocs/CmIdentity/bin/WibuCmIdValidator.jar";
*/
?>
```

動作テスト

browscap.ini のチェック

最初にWebブラウザーでhttp://localhost/ServerTest.phpにアクセスしてください。
右の画面が表示されます。

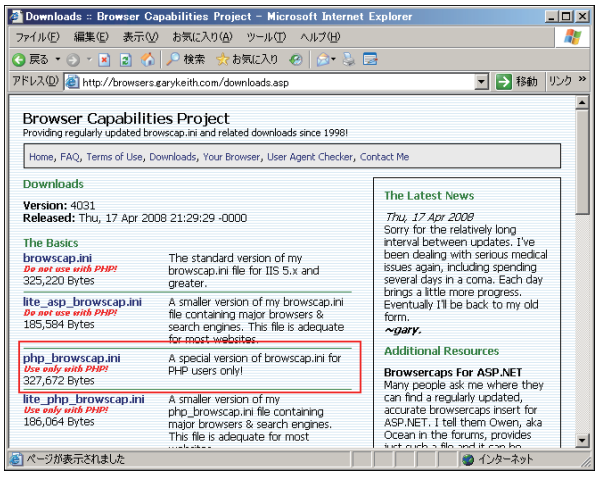


このページでbrowscap is not installed on this server ! Please install !というメッセージが表示される時は、PHP用のbrowscap.ini をインストールしてください。

php_browscap.ini のインストール

"php.ini" をテキストエディタで開いてbrowscap に "browscap.ini" の正しいパスを設定してください。
以下はその設定例です。デリミタは、スラッシュでなく \ということに注意してください。

```
[browscap]
browscap = C:\PHP\extras\browscap.ini
```



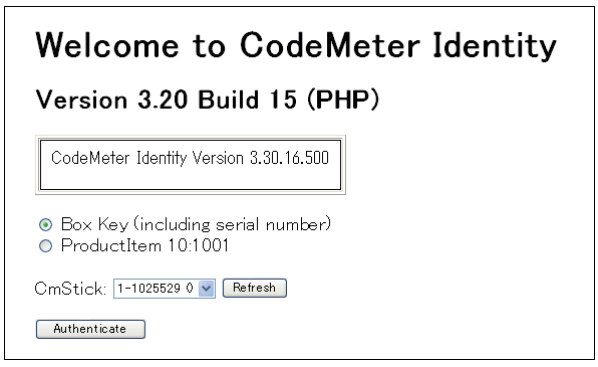
"browscap.ini"ファイルが無い場合は、http://browsers.garykeith.com/downloads.asp から "php_browscap.ini" をダウンロードして、 "php.ini"ファイルで指定されたフォルダにコピーしてください。

browscap の設定が完了したらもう一度　http://localhost/ServerTest.phpにアクセスして、browscapがインストールされていることを確認してください。

デモサイトの動作確認

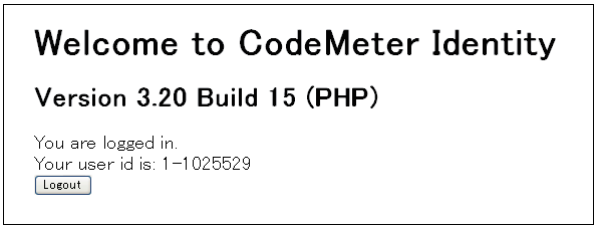
コードメータアイデンティティのデモサイトの動作テストを行います。最も簡単な方法はサーバー上で始めることです。

PCにCM-Stickを装着して、Webブラウザで http://<server name>/cmidentity/ にアクセスしてください。(例. http://localhost/cmidentity/)
右のページが表示されます。



2つのモードが、あります：
認証方法の 1 つは、CM-Stickの内部キー（シリアルを含む）を使用します。
もう 1 つの方法は、ファームコードとプロダクトコードを使用します。
どちらの方法も試すことができます。

[Authenticate]ボタンをクリックしてCM-Stickを介した認証が正しく行われると、右のようなログインメッセージが表示されます。
Box Key によって認証を行った場合、Your user id is: にはCM-Stickのシリアル番号が表示されます。
ProductItem によって認証を行った場合、Your user id is: には ファームコードとプロダクトコードが表示されます。



Chapter 10

CmBoxPgm の使い方

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。

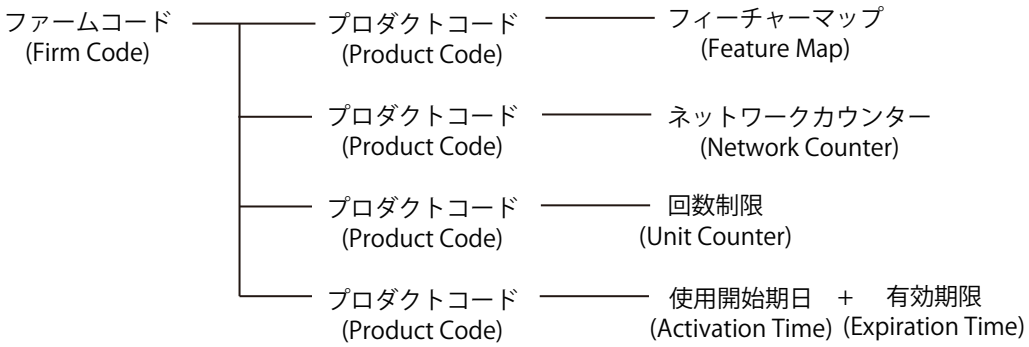
CmBoxPgmを使って、コードメータキーにコードを登録するには、下記の条件が必要です。
① コードメータ開発キットがインストールされていること。
② 貴社のライセンスファイルCmFirm.wbcがインストールされていること。
③ 貴社のコードメータFSB (CM-FSB)がPCに装着されていること。

作業は、Windows 2000/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ー、回数制限（ユニットカウンタ）や有効期限などのセキュリティオプション項目を追加します。

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ませんので、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。

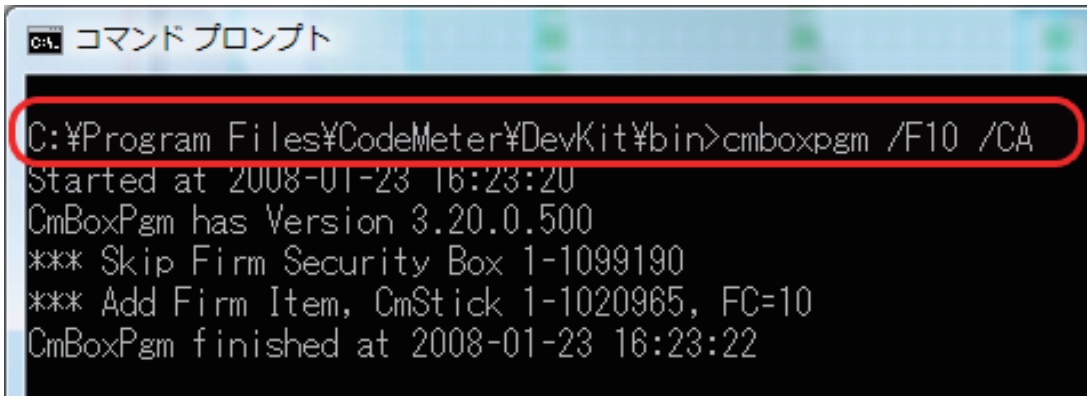
¥Program Files¥CodeMeter¥DevKit¥bin

［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 ）



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。

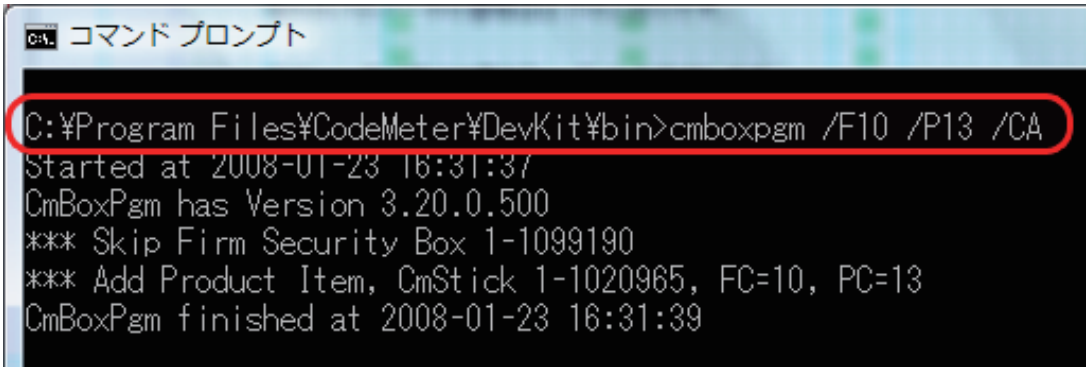
¥Program Files¥CodeMeter¥DevKit¥bin

［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 (ProductCode) = 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）



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ー(Access Counter)を設定。xは0 - 0xffffまでの符号なし整数(unsigned integer)を使用。デフォルト値は0xffff。

/FPT
Firm Precise Timeを設定。

/FTx
ファームアイテムテキスト(Firm Item Text)を設定。xにはテキスト文字を指定。テキスト文字はダブルクオーテーションで囲む。

/FUCx
ファームアップデートカウンターを設定。xは、符号なし整数(unsigned integer)で、デフォルト値は0。

/CA
新規のファームアイテムを作成

/CU
既存のファームアイテムを更新

/CD

ファームアイテムを削除

（例）

CmBoxPgm /QN1,4 /F206 /FPTR0 /FT¥"The Firm Item Text¥" /CA

インデックス範囲1から4の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（フィーチャーマップ）を設定。xは、0 - 0xffffffffの範囲の符号なし整数（unsigned integer)。

/PHD

Hidden Dataを設定。

/PTx

プロダクトアイテムテキストを設定。

/PNWC

Network License Counter（ネットワーク・ライセンス・カウンター）を設定

/PPD

Protected Dataを設定。

/PSD

Secret Dataを設定。

/PT ¥"<text> ¥"

プロダクトアイテムテキストを設定。

/PUC

Unit Counter（ユニットカウンター）を設定

/PD

User Dataを設定。

/CA

新規のプロダクトアイテムを作成

/CU

既存のプロダクトアイテムを更新

/CD

プロダクトアイテムを削除

/L

プロダクトアイテムをリスト表示する。

（例）

CmBoxPgm /QS1-1234 /F206 /P2001 /PETR30 /PUCA1492 /PFM0x8000 /CA

シリアル番号1-1234の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

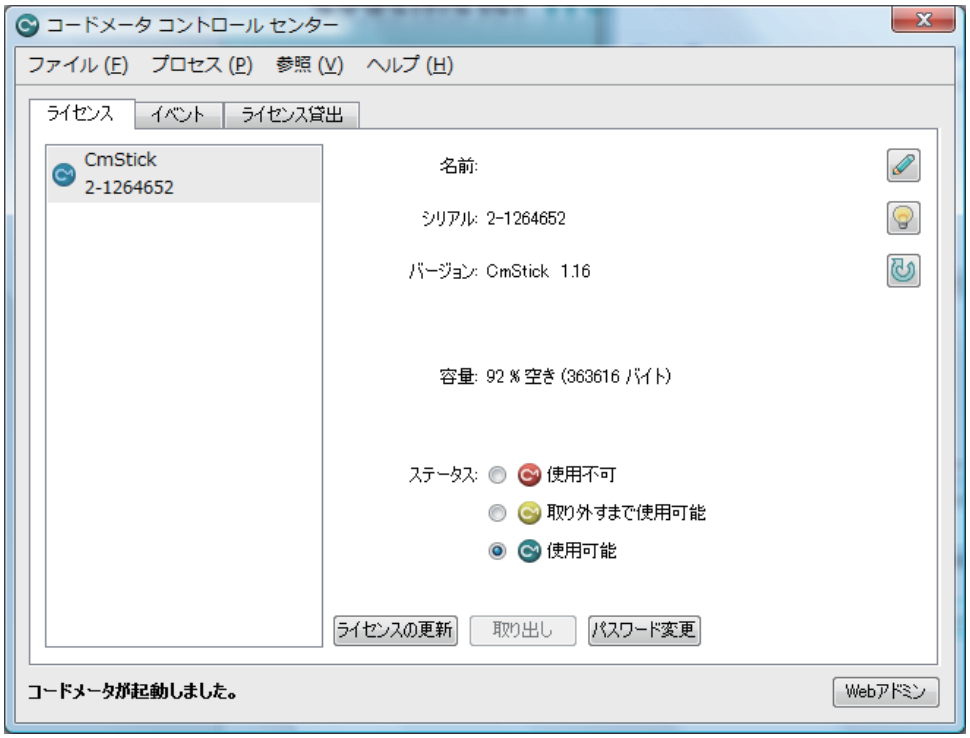
CmBoxPgm /?productitem

Product Itemに関するヘルプ情報をすべて表示

Chapter 11

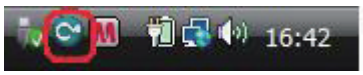
コードメータ コントロールセンターの使い方

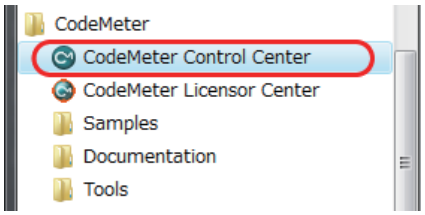
11-1. コードメータ コントロールセンターの説明



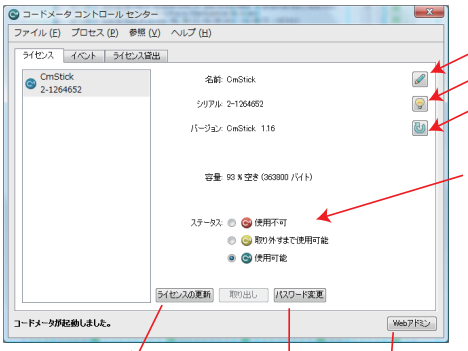
コードメータコントロールセンターの起動方法

コードメータコントロールセンターを起動するには、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。





CM-Stick のステータス情報等を表示



CM-Stick の名前を設定・変更する
選択されている CM-Stick の LED を点滅する
選択されている CM-Stick のファームウェアをバージョンアップする

ライセンスの更新を行う

CM-Stick 自身のパスワード
(PIN コード）の変更を行う

Web アドミン
を起動する

ステータス
CM-Stick のステータス状況が表示されます。
ボタンが選択されているステータスが現在のステータスになります。
ボタンを選択して CM-Stick のセキュリティ設定を変更することができます。
（変更するには CM-Stick のパスワードが必要です）
（ステータスの説明）

◯ **使用不可**
CM-Stick が使用不可の状態。

◯ **CM スティックを抜くまで使用可能**
CM-Stick を装着している間は使用可能。一度 PC から取り外すと、再度 PC に装着した際、パスワードを要求され、正しいパスワードを入力しない限り使用可能状態にならない。

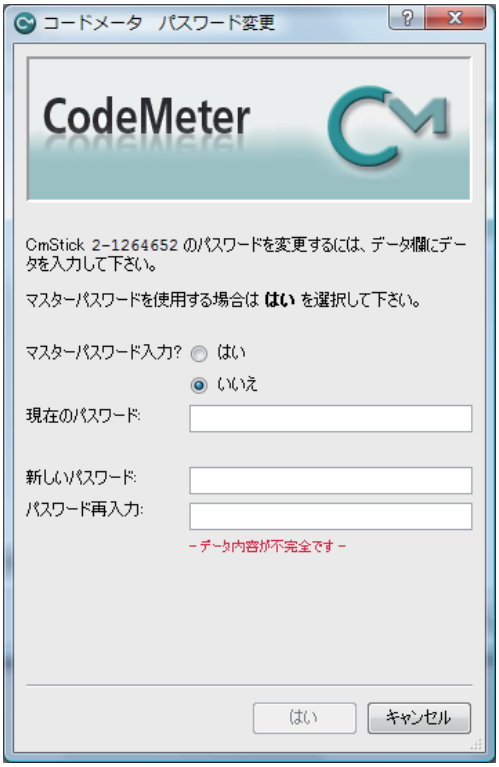
◯ **使用可能**
CM-Stick は使用可能の状態。PC から取り外し、再度 PC に装着しても、パスワード要求されずに使用可能の状態になる。

CM-Stick 自身のパスワード (PIN コード）を変更する

「パスワード変更」ボタンをクリックすると、パスワード変更画面が表示されます。

マスターパスワードは使用しませんので、"パスワード入力?"は"いいえ"を選択します。

現在のパスワードおよび新しいパスワードを入力し「はい」ボタンをクリックします。

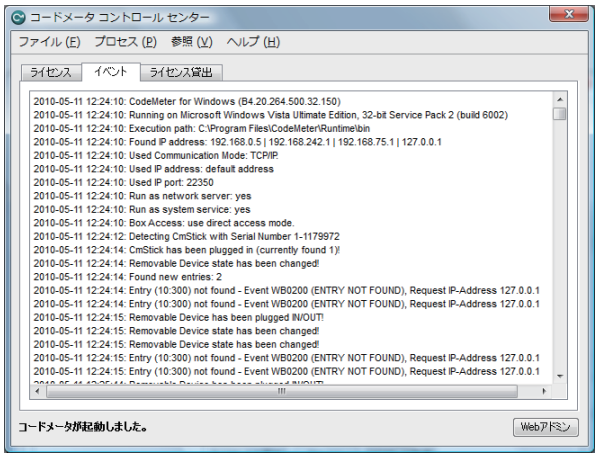


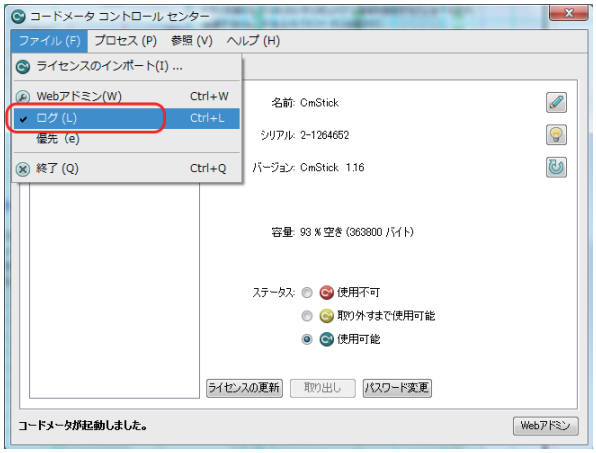
イベントログ

CM-Stickのイベントログを収集します。

イベントログファイルは、
¥Program Files¥CodeMeter¥Logsフォルダに作成されます。(拡張子"log")

[NOTE]
イベントログを収集するには、コードメータコントロールセンターの「ファイル」メニューから「ログ」をチェックする必要があります。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。



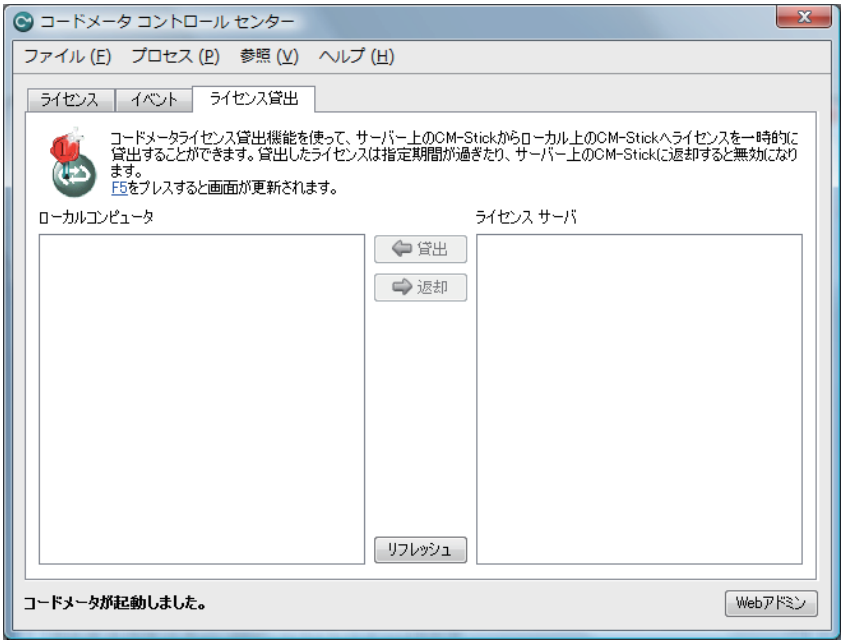


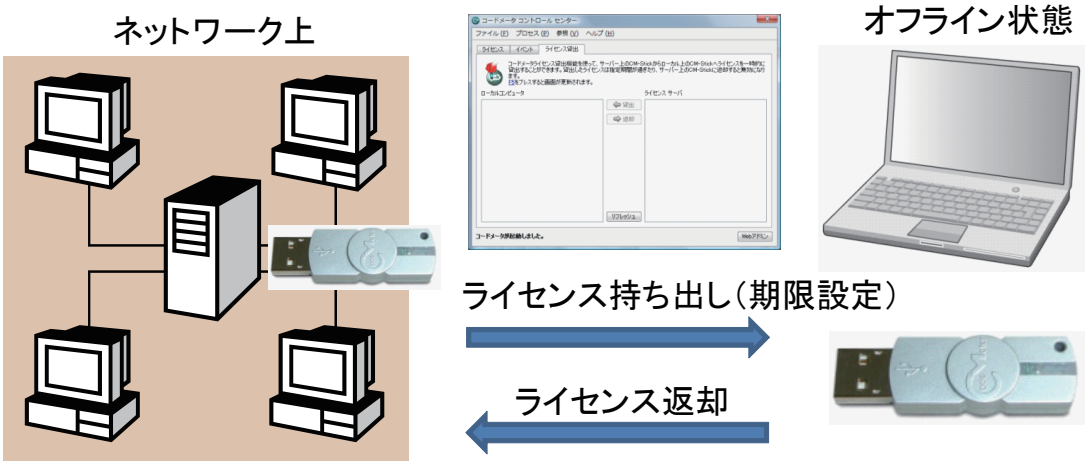
ライセンス貸出

コードメータライセンス貸出機能を使って、サーバー上のCM-Stickからローカル上のCM-Stick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PC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ローカルのCM-Stick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、ネットワーク上のCM-Stick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、ローカル上のCM-Stickは使用できなくなります。(全体のライセンス数は変化しない)

また、ライセンスを貸出する際に、有効期限を設定して貸出すため、有効期間が過ぎるとそのCM-Stick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)





11-2. 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。

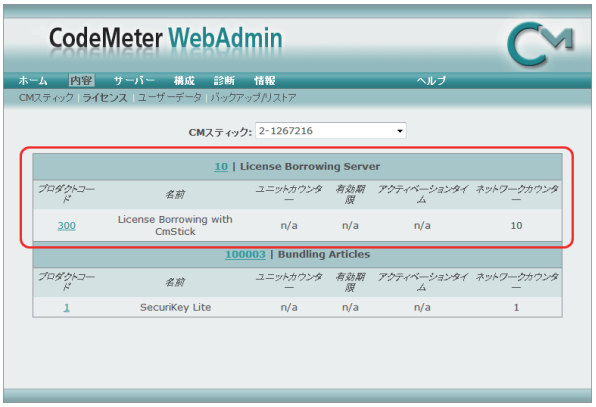
作業は、コマンドライン上でCmBoxPgm.exeを使って行います。また、あらかじめ、サーバー用CM-Stickのシリアル番号を確認しておいてください。（パラメータ入力時に必要）

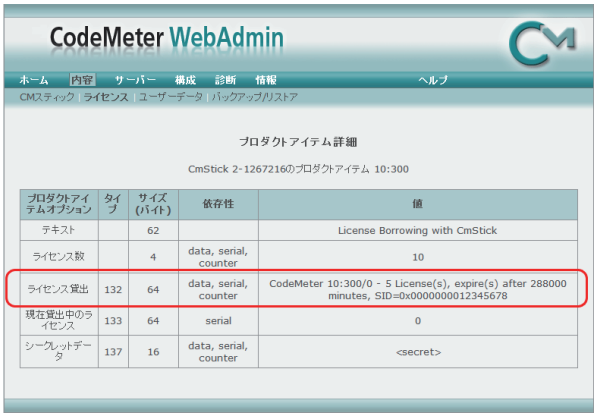
CmBoxPgm.exe /qs2-1267216 /f10 /ft:"License Borrowing Server" /cau /p300 /pt:"License Borrowing with CmStick" /plq10 /bls:cm,10,300,0,5,28800,0x12345678 /ca

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。





[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]

[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。

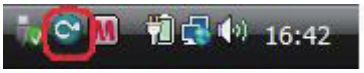
3．コードメータコントロールセンター上でライセンス貸出・返却作業を行う

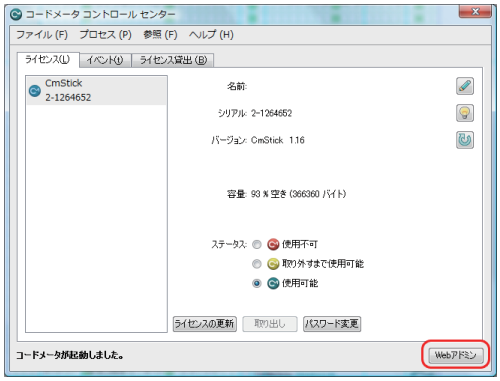
①サーバー用 CM-Stick をコードメータサーバーに装着する

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。





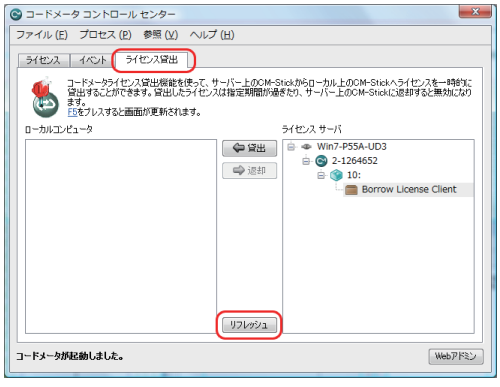
「構成」メニューをクリックし、「ネットワーク設定」画面を開き、「ネットワークサーバーの実行」にチェックを入れ、画面下の「設定」ボタンをクリックします。

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。

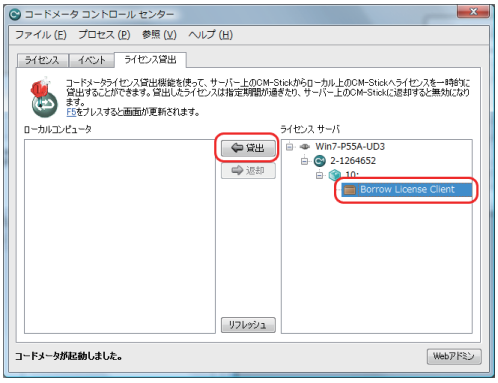


[NOTE]
「リフレッシュ」ボタンの代わりに"F5"をプレスすることも可能です。また、ネットワーク状況により、"ライセンスサーバー"ウィンドウにライセンスが表示されるまで時間がかかる場合があります。

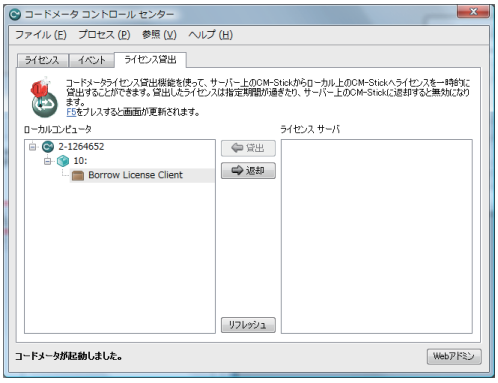
[NOTE]
ライセンス貸出を行うには、ローカルPCはネットワークに接続している必要があります。

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。

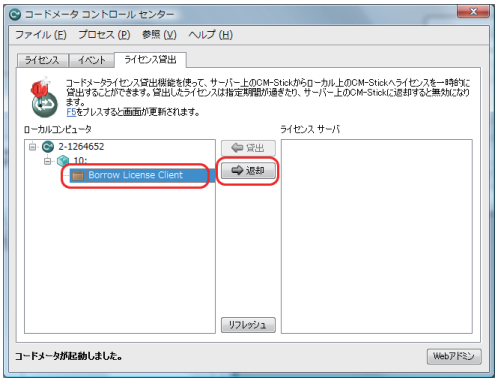


貸出に成功すると、"ローカルコンピュータ"ウィンドウにライセンスが表示されます。これでライセンス貸出は完了です。



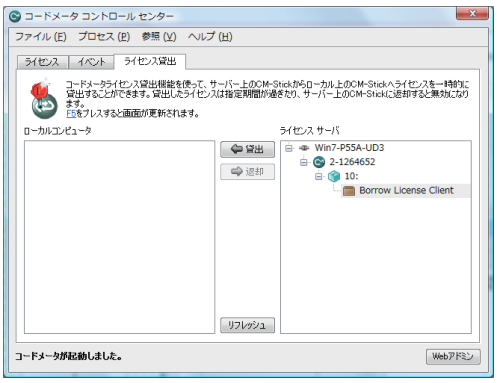
⑥ ライセンスを「返却」する場合

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。



11-3. 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]

"/bls"のパラメータについては、前章「11-2. ライセンス貸出・返却の方法」をご参照ください。

Chapter 12

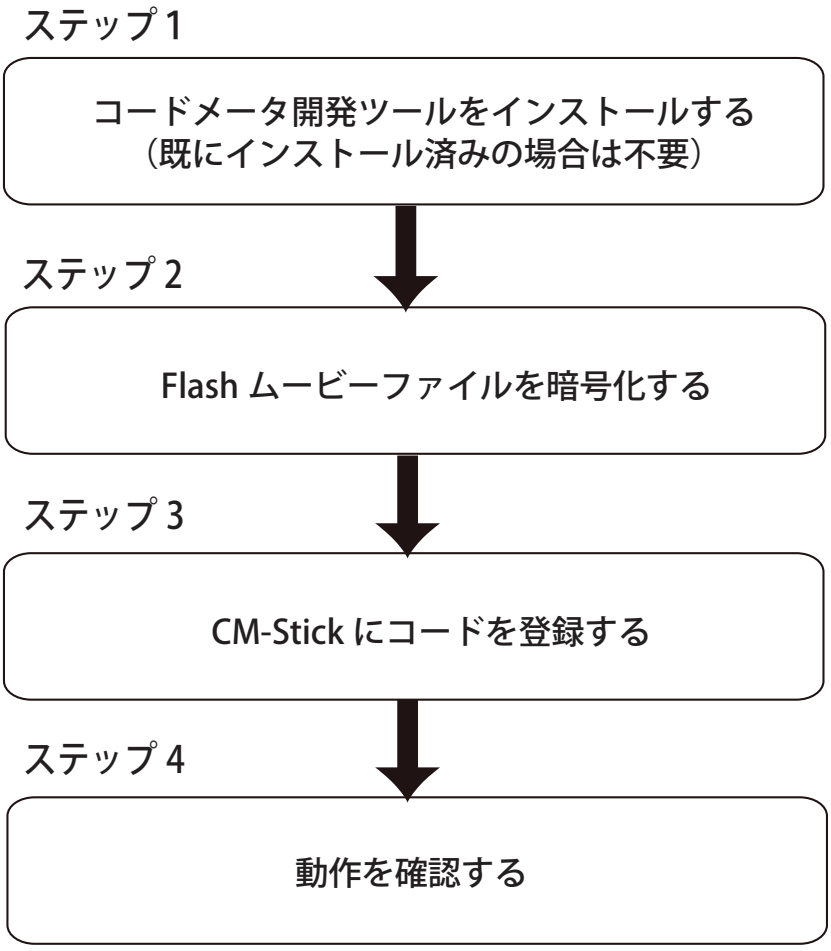
Adobe Flash ムービーファイルにプロテクトをかける

12-1. Adobe Flash ムービーファイルにプロテクトをかける

コードメータには、Adobe Flashムービーファイル(SWFまたはFLVファイル)をプロテクトする機能が搭載されています。コンテンツの販売、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効果的です。Adobe Flashムービーファイルを暗号化するには、自動暗号化ツールAxProtectorを使用します。

コードメータCDの中にあるサンプルFlashムービーファイルにプロテクトをかけてみます。サンプルFlashムービーファイルは、コードメータCDのTools/Flash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。



12-2. 作業に必要なもの

Adobe Flashムービー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 自動暗号化ツールAxProtector
② コードメータFSB (CM-FSB)

作業はWindows 2000/XP/Vista/7 (32bit/64bit) 上で行います。

12-3. コードメータ開発ツールをインストールする

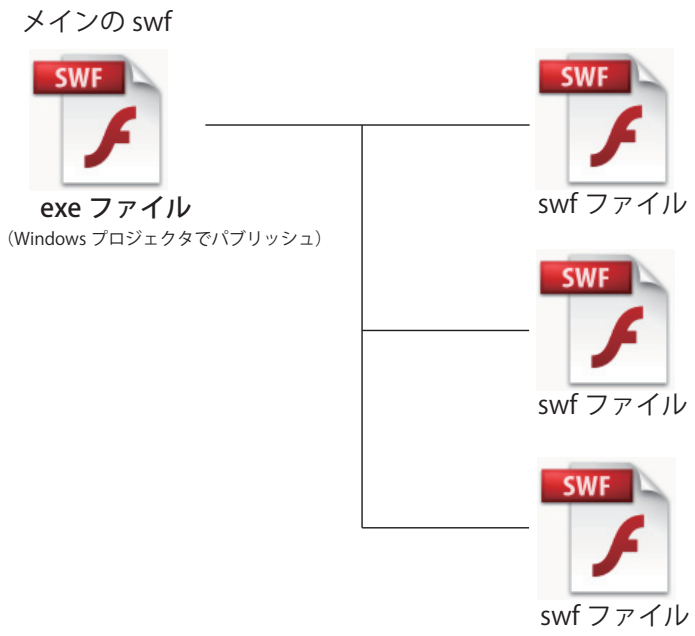
まずはじめに、コードメータ開発ツールをインストールします。すでに、コードメータ開発ツールがインストールされている場合は、次の「12-4. Flashムービーファイルを暗号化する」に進んでください。まだ、コードメータ開発キットのインストールが済んでいない場合は、前章「Chapter2 コードメータ開発ツールをインストールする」を参照してください。

12-4. Flash ムービーファイルを暗号化する

コードメータ開発キットのインストールが終了したら、次にFlashムービーファイルの暗号化を行います。暗号化が可能なファイルは、EXEファイル（Windows プロジェクタ）、swfファイルとflvファイルになります。暗号化作業には、コードメータFSB(CM-FSB)が必要です。

1. 暗号化に必要なファイル構成

コードメータで暗号化処理を行う場合、Flashコンテンツの大元になる最初のムービーは、htmlとswfの組み合わせでなく、Windowsプロジェクタ-exeファイルにする必要があります。Flashドキュメントファイル(flaファイル）をパブリッシュする際、Windowsプロジェクタ(exe）にてパブリッシュしてください。それ以外は、swfファイルまたはflvファイルのままで問題ありません。



2. 最初のムービーファイル (start.exe) を暗号化する

最初のムービーファイル名を仮に"start.exe"とします。この"start.exe"を暗号化するには、次の２通りの方法があります。

① 自動暗号化ツールAxProtectorのGUI上で暗号化する
② 自動暗号化ツールAxProtectorのコマンドライン環境で暗号化する

① 自動暗号化ツール AxProtector の GUI 上で暗号化する

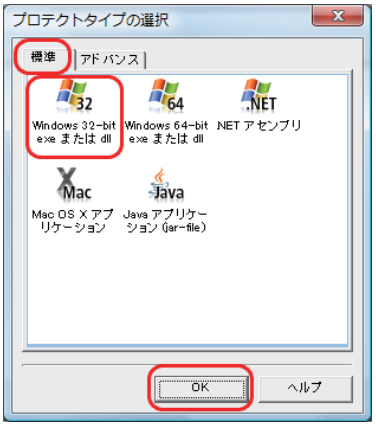
① CM-FSB を USB ポートに装着します

まず、貴社のコードメータFSB(CM-FSB)をUSBポートに装着します。評価版の場合は評価用CM-FSBをポートに装着します。

② AxProtector を起動する

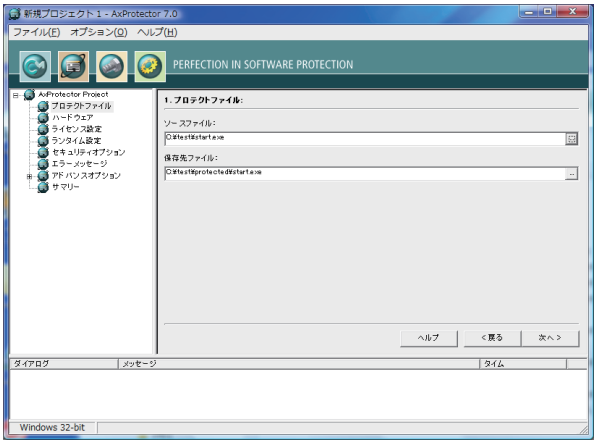
【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】から「AxProtector」を起動し 「Windows 32-bit exe または dll」を選択し、OKボタンをクリックします。

"AxProtectorGui.exe"は、インストール先の¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥binフォルダにありますので、直接ダブルクリックで起動することも可能です。



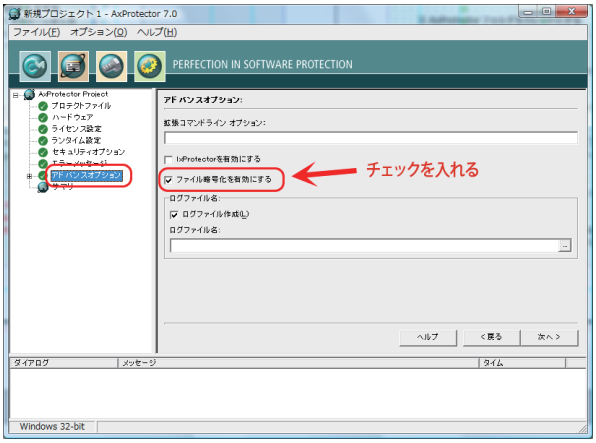
③ファイル名やオプション項目を入力する

AxProtector起動後は、通常通り、各画面で必要な項目を入力します。AxProtectorについての使用方法は、「Chapter 5 自動暗号化ツール AxProtectorについて」をご参照ください。



④「ファイル暗号化を有効にする」にチェックを入れる

AxProtectorの「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」にチェックを入れます。このオプション項目にチェックを入れないと、swfやflvの暗号化ファイルが開けませんので、必ず忘れずにチェックを入れてください。

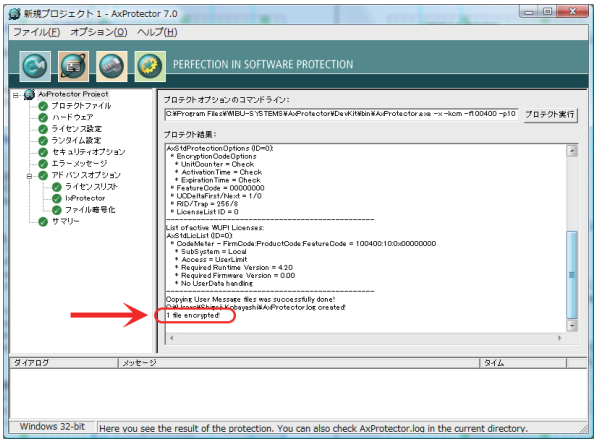


[注意!!]
必ず、「ファイル暗号化を有効にする」にチェックを入れてください。チェックを入れないと、暗号化されたswfやflvファイルが開きませんのでご注意ください。

⑤暗号化を実行する

「サマリー」画面で暗号化を実行し、"1 file encrypted!"が表示されていることを確認します。

これで、暗号化されたstart.exeが作成されました。



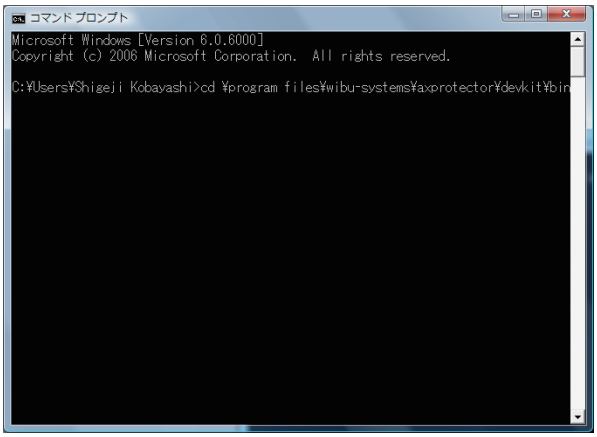
② 自動暗号化ツール AxProtector のコマンドライン環境で暗号化する

① CM-FSB を USB ポートに装着します。

まず、貴社のコードメータFSB(CM-FSB)をUSBポートに装着します。評価版の場合は評価用CM-FSBをポートに装着します。

②コマンドプロンプトを開く

コマンドプロンプトを開きます。[すべてのプログラム]->[アクセサリ]->[コマンドプロンプト]を選択します。



③ AxProtector フォルダをカレントにする

CD(Change Directory)コマンドを使い、AxProtectorフォルダをカレントにします。

¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin

［例］コマンドラインから
>CD ¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin ↓

④ start.exe を暗号化する

コマンドラインから、start.exeを暗号化します。ファームコード(Firm Code)は10、プロダクトコード(Product Code)は13で暗号化し、暗号化後のファイル名を"pstart.exe"にします。"start.exe"はtestフォルダにあるものとします。

>AxProtector /kcm /f10 /p13 /cf0 /aaes /sl /nn /cad /o:¥test¥pstart.exe ¥test¥start.exe

* パラメータ/o: には、暗号化生成されるファイル名を指定します。
* パラメータ/cadは、「ファイル暗号化を有効にする」オプションパラメータです。

パラメータ入力後、リターンキーを押すと暗号化が開始されます。暗号化処理が成功すると、下記のメッセージが表示されます。testフォルダに"pstart.exe"が作成されたことを確認してください。

*** Creating RID keys: 100.0%**
1 file encrypted!

3. swf ファイル（flv ファイル）を暗号化する

swfファイルやflvファイルを暗号化する場合も、前述のstart.exeを暗号化する方法と同じく２通りの方法があります。

① 自動暗号化ツールAxProtectorのGUI上で暗号化する
② 自動暗号化ツールAxProtectorのコマンドライン環境で暗号化する

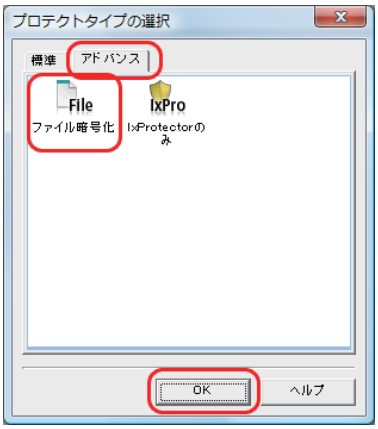
① 自動暗号化ツール AxProtector の GUI 上で暗号化する

1 CM-FSB を USB ポートに装着します

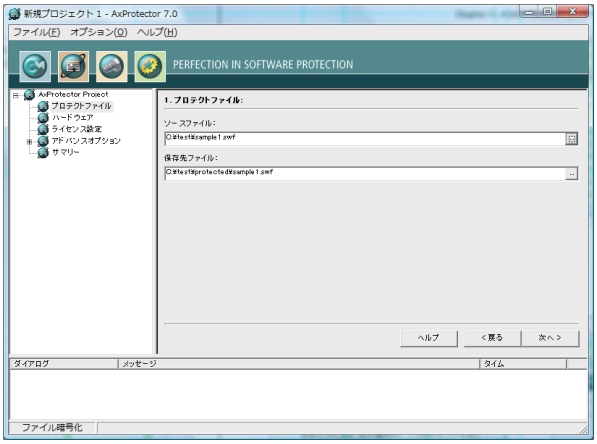
まず、貴社のコードメータFSB(CM-FSB)をUSBポートに装着します。評価版の場合は評価用CM-FSBをポートに装着します。

2 AxProtector を起動する

【スタート】→【すべてのプログラム(P)】→【AxProtector】→【AxProtector】から「AxProtector」を起動し、「プロトタイプの選択」画面の「アドバンス」上タブから「ファイル暗号化」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フォルダにありますので、直接ダブルクリックで起動することも可能です。

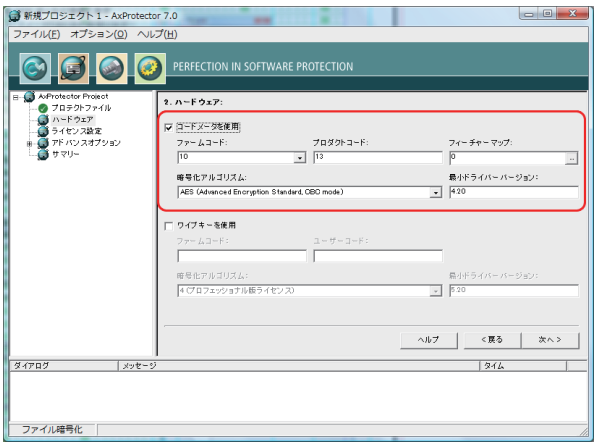


3 ファイル名を入力する

AxProtector起動後は、通常通り、各画面で必要な項目を入力します。まず最初に、「プロテクトファイル」画面でソースファイル名と暗号生成されるファイル（保存先ファイル）名を入力します。AxProtectorについての使用方法は、「Chapter 5 自動暗号化ツール AxProtectorについて」をご参照ください。

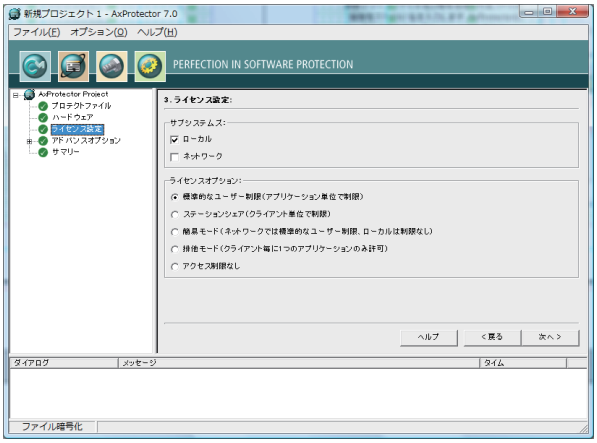
4 ファームコード・プロダクトコードを入力する

暗号化に使用するファームコード、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、最小ドライバーバージョンを入力します。ここでは、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=0、暗号化アルゴリズム=AES（デフォルト）、最小ドライバーバージョン=4.20を設定します。



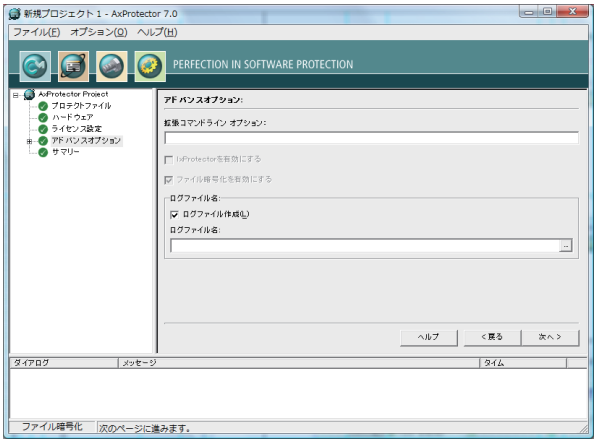
⑤ ライセンス設定

サブシステムズには"ローカル"、ライセンスオプションには"標準的なユーザー制限（アプリケーション単位で制限）"を指定します。ファイル暗号化の場合、サブシステムズ="ネットワーク"は対応しません。



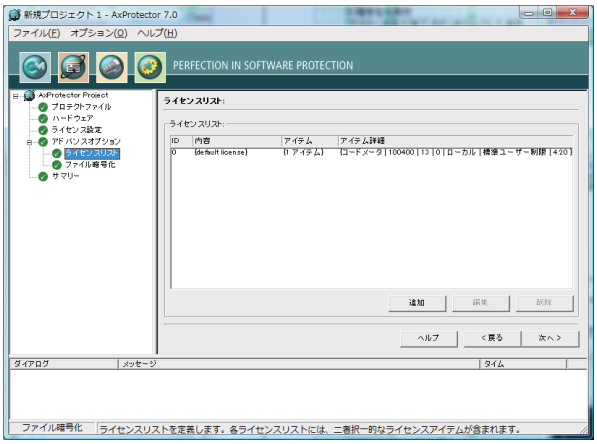
⑥ アドバンスオプション

「アドバンスオプション」画面では特に設定する必要がありませんので、そのまま「次へ」をクリックします。



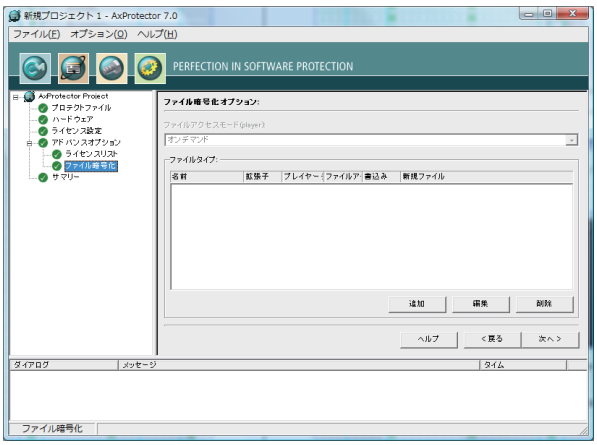
⑦ ライセンスリスト

「ライセンスリスト」画面では特に設定する必要がありませんので、そのまま「次へ」をクリックします。



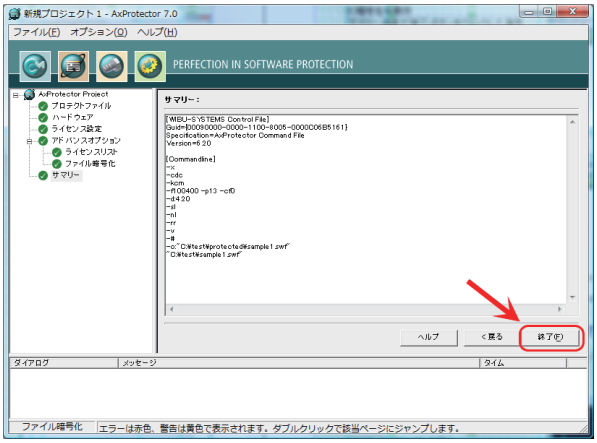
⑧ ファイル暗号化オプション

Flashのswfまたはflvの暗号化の場合、この「ファイル暗号化オプション」は使用しませんので、そのまま「次へ」をクリックします。

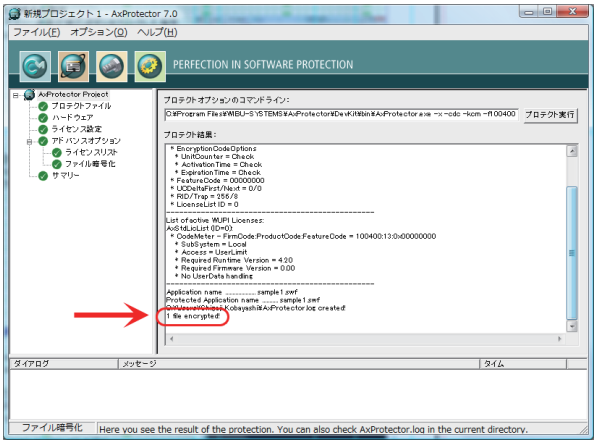


⑨ 暗号化を実行

「サマリー」画面で「終了」ボタンをクリックして、暗号化を実行します。



正常に暗号化されると、"1 file encrypted!"が表示されます。これで、swf（またはflv）のファイルが暗号化されました。



② 自動暗号化ツール AxProtector のコマンドライン環境で暗号化する

swfファイルやflvファイルをコマンドライン環境で暗号化する場合、パラメータ"/cd"を使用します。（exeの場合は"/cad"を使用）

① CM-FSB を USB ポートに装着します。

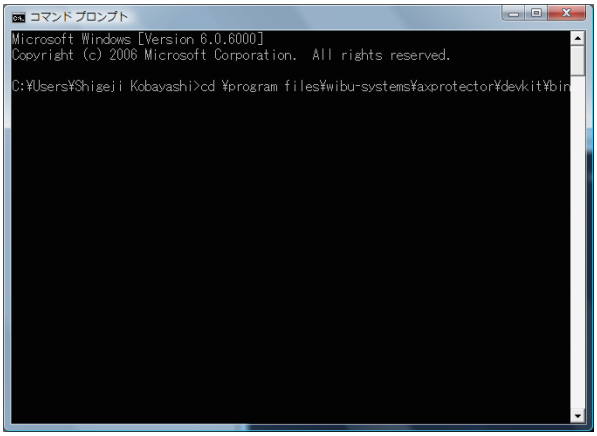
まず、貴社のコードメータFSB（CM-FSB）をUSBポートに装着します。（評価版の場合は評価用CM-FSB）

②コマンドプロンプトを開く

コマンドプロンプトを開きます。[すべてのプログラム]->[アクセサリ]->[コマンドプロンプト]を選択します。

③ AxProtector フォルダをカレントにする

CD(Change Directory)コマンドを使い、AxProtectorフォルダをカレントにします。



¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin

［例］コマンドラインから

>CD ¥Program Files¥WIBU-SYSTEMS¥AxProtector¥DevKit¥bin ↓

④ swf ファイル（flv ファイル）を暗号化する

コマンドラインから、xxx.swfファイルを暗号化します。ファームコード(Firm Code)は10、プロダクトコード(Product Code)は13を設定し、元のファイルを暗号化ファイルで上書きします。元の"xxx.swf"ファイルはtestフォルダにあるものとします。暗号化ファイルを上書きする場合、元のファイルは無くなりますので、必ずオリジナルファイルを別のフォルダにバックアップしてから作業を進めてください。暗号化ファイルを別フォルダに作成することも可能ですが、フォルダ名が変わることでリンク構成が変わることを留意ください。

>AxProtector /kcm /f10 /p13 /cf0 /aaes /sl /nn /cd /o:¥test¥xxx.swf ¥test¥xxx.swf

パラメータ入力後、リターンキーを押すと暗号化が開始されます。暗号化処理が成功すると、下記のメッセージが表示されます。

*** Creating RID keys: 100.0%**
1 file encrypted!

他のswfファイル(flvファイル）も同様に暗号化します。

[NOTE]
パラメータ/o: には、暗号化生成されるファイル名を指定します。
swfまたはflvファイルを暗号化する場合は、パラメータ"/cad"の代わりに"/cd"を使用します。

12-5. 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。実際の登録方法は、前章「Chapter 3 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」を参照してください。

12-6. 動作を確認する

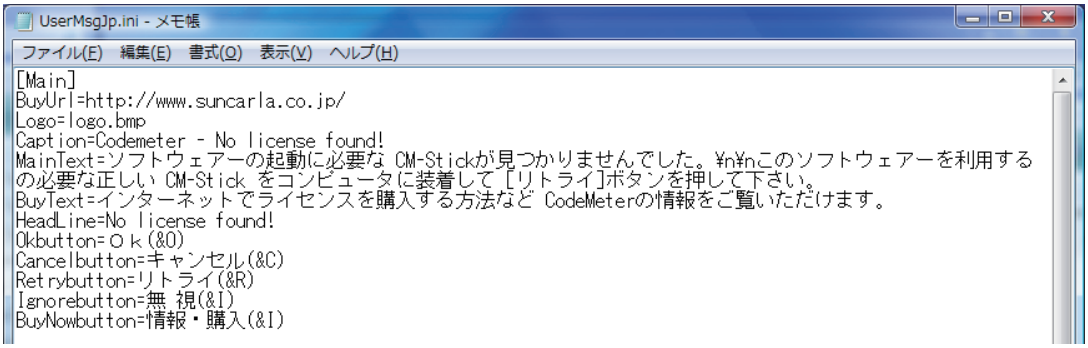
作成したCM-StickをPCに装着し、先ほど暗号化したFlashムービーファイルをstart.exeから開きます。CM-Stickがあれば開き、ないと開かないことが確認できます。起動は、必ずstart.exeから始めます。他のswfファイルは暗号化されているため、単独で開いても開きません。start.exeが、暗号化されているswfファイルを復号化する役目をします。

12-7. エラーメッセージをカスタマイズする

CM-Stickが見つからない時のエラーメッセージをカスタマイズすることができます。
コードメータCDのTools¥AxProtector¥UserMsgフォルダの中にある"CmUserMsgJp.dll"と"UserMsgJp.ini"を利用します。

① "UserMsgJp.ini" ファイルをカスタマイズする

"UserMsgJp.ini"ファイルをメモ帳などのテキストエディタで開き、各項目を貴社ニーズに応じて修正します。



【UserMsgJp.iniファイルの説明】
デフォルトで設定されているファイルの内容例です。

- ○ 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。
- ○ 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 ： 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[リトライ]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[詳しくは](HPへのリンク)ボタンの名前を設定します。

UserMsgJp.iniファイルを編集することで、エラーメッセージ文だけでなく、メッセージウィンドウのタイトルや、ボタンの名前、URLのリンク先、添付画像などについても変更することができます。

② Flash ムービーファイルを暗号化する

コマンドライン上でFlashムービーファイルを暗号化します。カスタマイズしたメッセージは、ファイルを暗号化することで反映されます。AxProtectorで暗号化する際、"/u:"パラメータで"CmUserMsgJp.dll"を指定します。"CmUserMsgJp.dll"と"UserMsgJp.ini"がtestフォルダにあるとします。

EXEプログラムの場合：

>AxProtector /kcm /f10 /p13 /cf0 /aaes /sl /nn /cad /u:¥test¥CmUserMsgJp.dll /o:¥test¥pstart.exe ¥test¥start.exe

SWFまたはFLVファイルの場合：

>AxProtector /kcm /f10 /p13 /cf0 /aaes /sl /nn /cd /u:¥test¥CmUserMsgJp.dll /o:¥test¥xxx.swf ¥test¥xxx.swf

③ メッセージを確認する

暗号化したファイルと同じフォルダに"CmUserMsgJp.dll"と"UserMsgJp.ini"をコピーします。また、BMPの画像を使用した場合は、そのBMPファイルも同じフォルダにコピーします。

CM-Stickを装着しない状態でFlashムービーを起動し、カスタマイズしたメッセージが表示されるか確認してください。念のため、CM-Stickを装着し、Flashムービーが正常動作することも確認します。

ユーザーに配布する場合は、"CmUserMsgJp.dll"、"UserMsgJp.ini"、使用したBMPファイルをFlashコンテンツと一緒に配布します。

12-8. ユーザーに配布する場合

暗号化したFlashムービーコンテンツをエンドユーザーに配布する場合は、

①暗号化したFlashファイル(CmUserMsg.dll, UserMsgJp.ini, BMPを含みます)
②CM-Stick
③コードメータランタイムキット
　CodeMeterRuntime32.exe (32ビットOSの場合)
　CodeMeterRuntime64.exe (64ビットOSの場合)

が必要です。

初めてFlashムービーコンテンツを開く場合は、まずコードメータラインタイムキットをPCにインストールします。CodeMeterRuntime32.exeまたはCodeMeterRuntime64.exeをダブルクリックし、メッセージに従って進めます。次に、CM-StickをPCに装着すると、デバイスインストールが自動的に行われ、数秒後に使用準備ができますので、暗号化されたFlashムービーをスタートさせます。

コードメータランタイムキットは、一度PCにインストールすると次回からは不要です。CM-Stickを装着し、Flashムービーを起動するだけです。

動作環境

OS: Windows2000/2003/XP/Vista/7 (32bit/64bit)
Adobe Flash Player 7/8/9/10

Chapter 13

CodeMeter Core API について

13-1. 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。

13-2. サンプルプログラムについて

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

¥Program Files¥CodeMeter¥DevKit¥Samples

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, HTML, 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ュフェストファイルを含めてください。
- マニュフェストで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。

13-3. CodeMeter Core API 一覧

Accessing
- CmAccess
- CmAccess2
- CmRelease

Authentication
- CmCalculateDigest
- CmCalculateSignature
- CmGetPublicKey
- CmValidateSignature

Enabling
- CmEnablingWriteApplicationKey
- CmEnablingGetApplicationContext
- CmEnablingGetChallenge
- CmEnablingSendResponse
- CmEnablingWithdrawAccessRights

Encryption
- CmCrypt
- CmCryptEcies
- CmCryptSim
- CmCalculatePioCoreKey
- CmGetSecureData
- CmDecryptPioData
- CmGetPioDataKey

Error Management
- CmGetLastErrorCode
- CmGetLastErrorText
- CmSetLastErrorCode

Management
- CmBorrow
- CmCheckEvents
- CmGetBoxContents
- CmGetBoxes
- CmGetInfo
- CmGetLicenseInfo
- CmGetServers
- CmGetVersion
- CmRevalidateBox

Programming
- CmActLicenseControl
- CmCreateProductItemOption
- CmCreateSequence
- CmProgram
- CmReserveFirmItem
- CmValidateEntry

Remote Programming
- CmExecuteRemoteUpdate
- CmGetRemoteContext

CmGetRemoteContext2
CmGetRemoteContextBuffer
CmListRemoteUpdate
CmListRemoteUpdate2
CmListRemoteUpdateBuffer
CmSetRemoteUpdate
CmSetRemoteUpdate2
CmSetRemoteUpdateBuffer

Time Update

CmSetCertifiedTimeUpdate

13-4. Linux で使用する場合

CodeMeterバージョン 4.20 は以下の条件が必要です。

Linux kernel 2.4.x (or kernel 2.6.x)
i386-architecture (ia32 or x86_64)
glibc 2.3 or higher
stdc++ 5.0 (gcc3) / stdc++ 6.0 (gcc4)
X11 Server (XFree86 or X.org)
TCP/IP network support
hotplug or udev
and Java JRE 1.4.2 or newer for online shopping with JTrigger

Installation

コードメータパッケージをインストールするには、Linuxディストリビューションに付属のパッケージマネジャーを使用するか、以下のシェルコマンドを実行してください。

CodeMeter Runtime Packageのインストール:

RPM: rpm -ivh package_name.i368.rpm
DEB: dpkg -i package_name.deb

CodeMeter Runtime Packageのアンインストール:

RPM: rpm -ev CodeMeter
DEB: dpkg -r codemeter
DEB: dpkg --purge codemeter

Configuration

コンフィグレーションには、Server.iniファイルをお使いください。(/etc/wibu/CodeMeter)
コンフィグレーションはCodeMeterLinが起動していない時のみ可能です。

コードメータをオンラインショップなどのWebブラウザーで使用する場合、WebブラウザーがJava 1.4.2以上およびJavaScriptをサポートする必要があります。Webブラウザーから、URL: file://usr/share/doc/CodeMeter/AppletExample.htmlを開き、Java-appletが動作することを確認できます。次に、WibuCmTrigger.jarをjava-ext-directoryにインストールします。通常はインストールパッケージのインストーラスクリプトが自動的に行います。もし行われない場合は、WibuCmTrigger.jarをディレクトリ ${JAVAHOME}/lib/extにコピーしてください。

Hotplug

コードメータには、コードメータStickを認識するためのUSB-hotplugging インフラストラクチャが必要です。 ホットプラギングが新しいLinuxディストリビューションで機能していない場合は、新しいudevメカニズムが使用されているかどうかを確認してください。udevが使用されている場合は、/usr/share/CodeMeter/52-codemeter.rulesを/etc/udev/rules.d/にコピーすることにより問題が解決します。

CodeMeterLinとCmStickとの通信を行うためには、USBマスストレージデバイスとSCSIジェネリックデバイスをサポートするLinuxカーネル(kernel 2.4.x またはkernel 2.6.x)が必要です。セルフコンパイルLinuxカーネルを使用している場合は、USBストレージとSGデバイスをモジュールとしてサポートしているか確認してください。

kernel-config例:

CONFIG_BLK_DEV_SD=m
CONFIG_CHR_DEV_SG=m
CONFIG_SCSI_MULTI_LUN=y

kernelが正常にコンフィギュアされている場合、次のコマンドを実行することで、現在接続されているすべてのCmStickがリスト表示されます。

codemeter-info -L

もし表示されない場合は、Linuxカーネルまたはカーネルモジュールが正しくロードできていない可能性が考えられます。

問題が発生した場合は、下記内容を明記の上、弊社サポートまでご連絡ください。

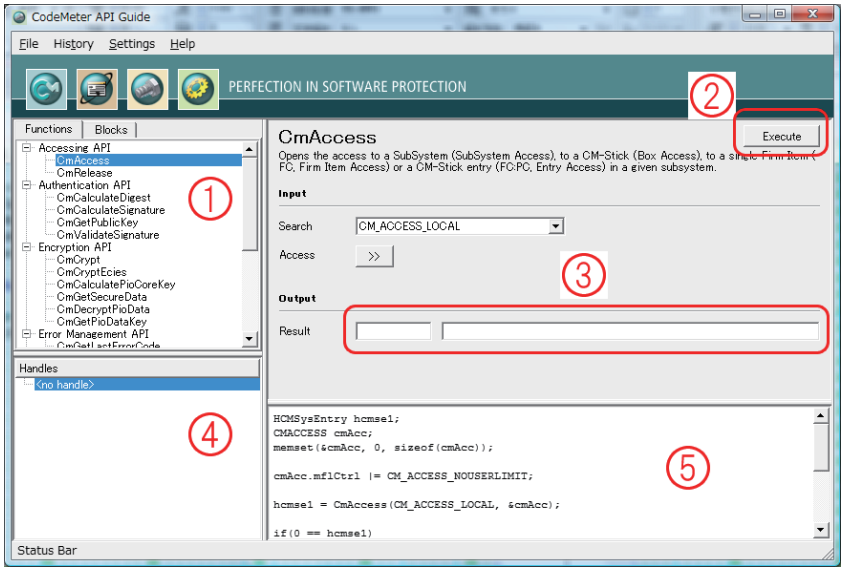
CodeMeterのバージョン
システム内容:
ご使用のOSとカーネルバージョン
インストールした C/C++ ライブラリのバージョン
USBサブシステム USB 1.1 or USB 2.0n?
USB ハブを使用しているか?
どのホットプラギングメカニズムを使用しているか? hotplug or udev?（パッケージマネジャで確認）
その他関連する内容および情報など

13-5. 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実行させながら、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CM-Stick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

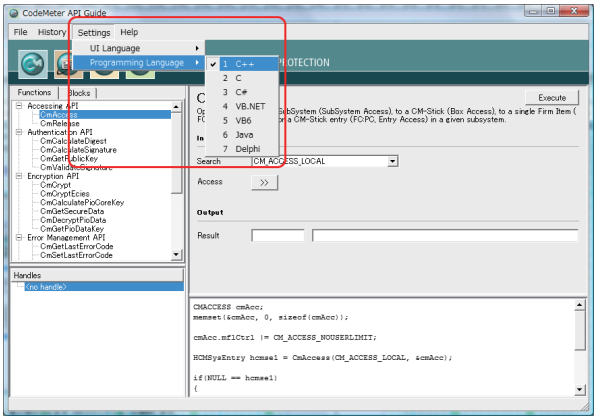
【スタート】→【プログラム(P)】→【CodeMeter】→【Tools】→【CodeMeterApiGuide】をクリックして起動します。



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

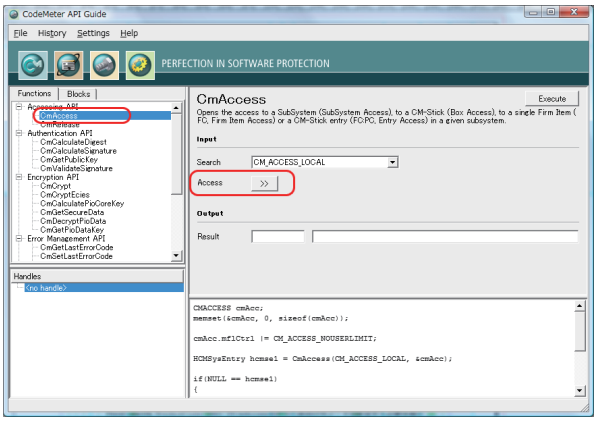
13-6. CodeMeter API ガイドの使用例

実際に ファームコード=10、プロダクトコード=13のCM-Stick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。

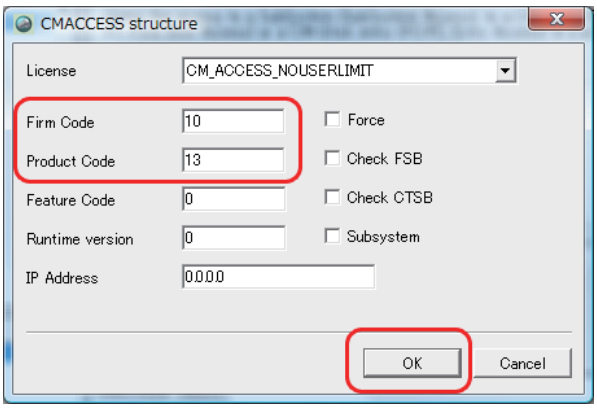


特定のファームコードとプロダクトコードが設定されたCM-Stickがあるかどうかをチェックするだけであれば、CmAccess() → CmGetLasetErrorCoe() → CmRelease() を行うだけで実現できます。CodeMeter API ガイドで実際に実行してみましょう。

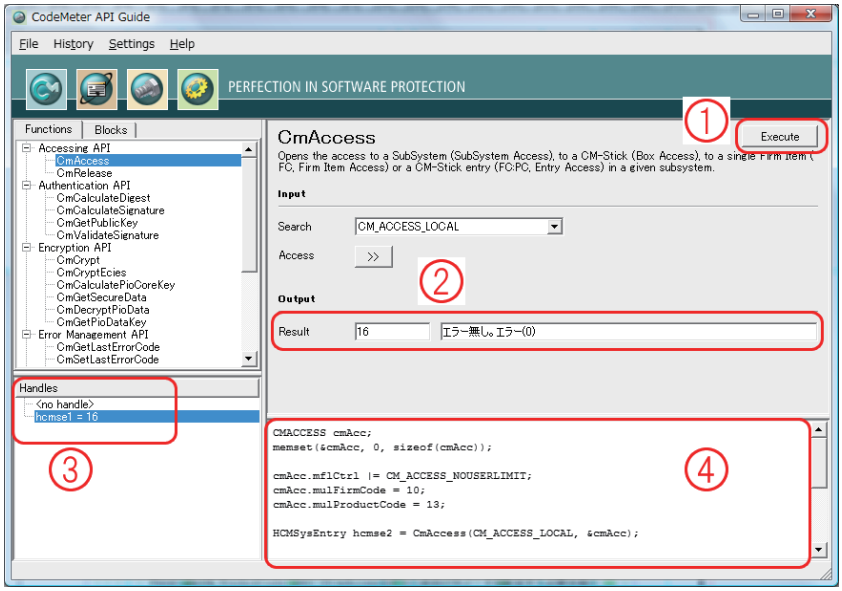
①FirmCode＝10、ProductCode=13 が登録されたCM-StickをPCに装着し、CmAccess を選択します。Access項目の［>>］をクリックし、CMACCESS ストラクチャ画面を開きます。



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。

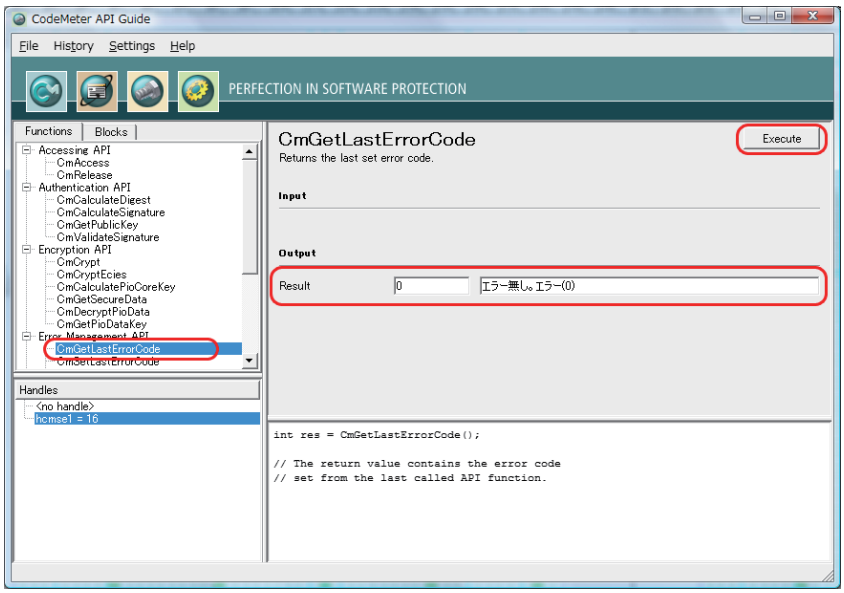


③ CodeMeter APIガイド画面に戻り、「Execute」ボタンをクリックすると、CMACCESSストラクチャで設定した内容でCmAccessファンクションが実行され、その結果がResult欄に表示されます。(②) CmAccessファンクションが返したハンドルは、「Handles」欄に表示されます。(③) 実行されたソースコードは(④)に表示されます。

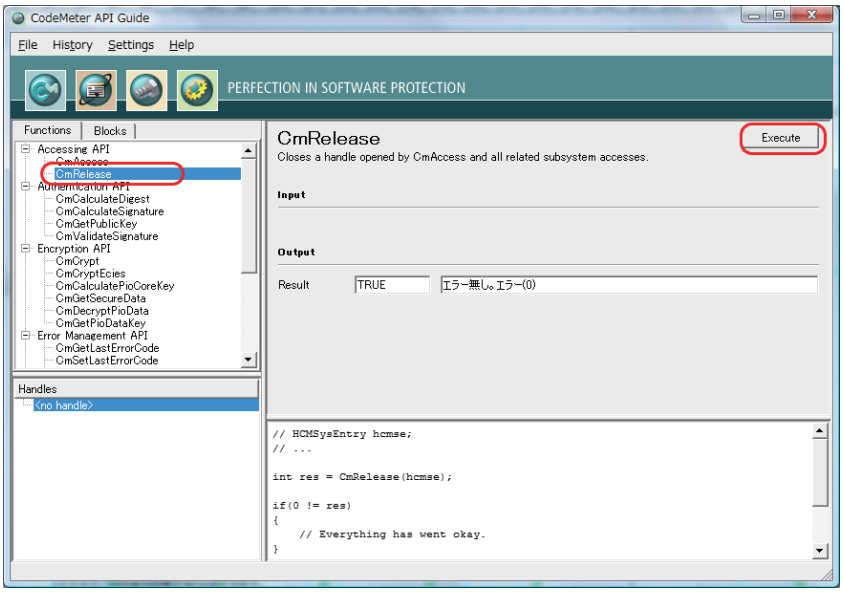


実行に成功すると、リターン値に0以外の整数値が返り、実行エラーの場合は、リターン値に0が返ります。(②) リターン値が0であるとエラーが発生したことはわかりますが、どのようなエラーが発生したかまではわかりません。どのようなエラーが発生したかを知るために、CmAccessファンクションに続けてCmGetLastErrorCodeファンクションを実行します。このCmGetLastErrorCodeファンクションは、直前に実行したファンクションのエラーコードを取得するファンクションです。

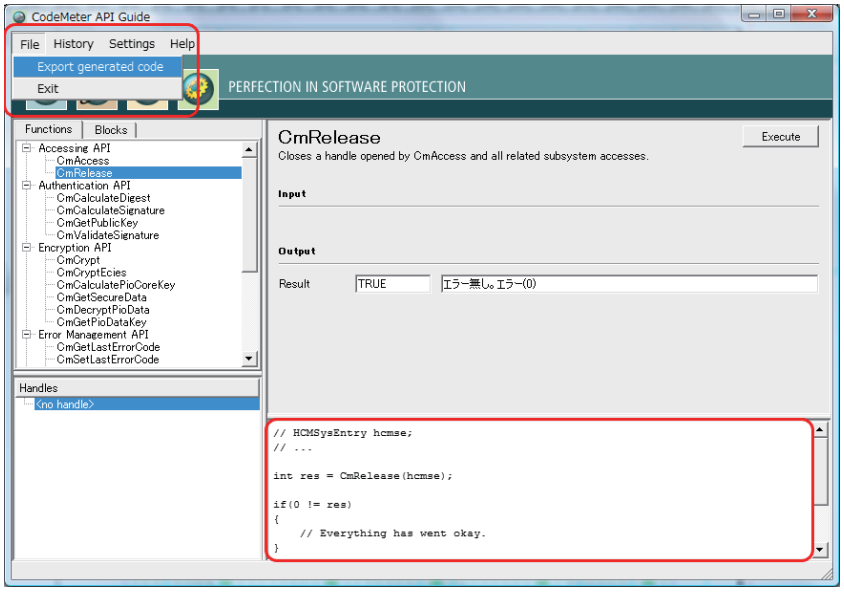
④ Functions一覧から、CmGetLastErrorCodeを選択し、「Execute」ボタンをクリックします。エラーが無い場合、リターン値に0が返ります。エラーの場合は、そのエラーコードが返ります。エラーコードを調べることでエラーの内容を把握することができます。



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



13-7. Access Mode について

コードメータCM-Stickにアクセスするモードには次の４種類があります。CmAccessファンクションを使って、CM-Stickへのアクセスモードを決めます。

Subsystem Access
Box Access
Firm Item Access
Entry Access

Subsystem Access:
このSubsystem Accessは１つのサブシステムアクセスであり、CodeMeter.exeがこのサブシステム上で起動していればCM-Stickがなくてもハンドルを返します。このアクセスを行うには、Firm Code = 0に設定し、CMACCESSストラクチャのmflCtrlパラメータにはCM_ACCESS_SUBSYSTEMを設定します。
サブシステム上でCodeMeter.exeが起動しているかどうかを確認するときに使用します。

Box Access
このアクセスは１つのCM-Stickに対するアクセスです。Box Accessは、次のようになります。
* Firm Code = 0で Product Code = 0の場合、最初に見つけたCM-Stickにアクセスします。
* Firm Code = 0で Product Code = 0、さらにCMACCESSストラクチャのmcmBoxInfoでシリアル番号を指定した場合、そのシリアル番号を持つCM-Stickだけにアクセスします。

Firm Item Access
このアクセスは、プロダクトアイテムやプロダクトアイテムオプションを持たないファームアイテムに対するアクセスです。このアクセスは、CM-Stickに新規のプロダクトアイテムを書くときに必要になります。Firm Item Accessには次の設定が必要です。
* Product Code = 0の場合、指定したFirm Item（ファームアイテム）を持つ最初に見つけたCM-Stickにアクセスします。
* Product Code = 0で、CMACCESSストラクチャのmcmBoxInfoでシリアル番号を指定した場合、指定したFirm Item（ファームアイテム）を持ち、なおかつ、そのシリアル番号を持つCM-Stickだけにアクセスします。

Entry Access
このアクセスは、CM-Stickエントリに対するアクセスです。すべてのエントリが対象になります。Firm Item = 0のエントリの場合はローカルアクセスだけになります。Entry Accessは次の設定が必要です。
* Product Codeが0でない場合、指定したエントリを持つ最初に見つけたCM-Stickのエントリにアクセスします。
* Product Codeが0でなく、CMACCESSストラクチャのmcmBoxInfoでシリアル番号を指定した場合、そのシリアル番号を持つCM-Stickのエントリだけにアクセスします。

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

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 2000/XP/Vista/7, Windows Server 2000/2003)
①32bit用コードメータランタイムキット "CodeMeterRuntime32.exe"（22MB)

Windows OS(64bit)の場合
(Windows XP/Vista/7 Server 2003/2008)
①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe" (24MB)

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(64bit), Windows Server 2003/2008 (64bit)
①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe" (24MB)

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

.NETアプリケーション(32bit/64bit版)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、コードメータ.NETアセンブリ"WibuCmNET.msi"をインストールする必要があります。

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 CodeMeter .NET assembly (Win32/64)"WibuCmNET.msi"

Windows OS(64bit)の場合
①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe"
② CodeMeter .NET assembly (Win32/64) "WibuCmNET.msi"

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 2000/XP/Vista/7 (32bit/64bit), Windows Server 2000/2003/2008 (32bit/64bit)
Adobe Acrobat 6/7/8/9 または Adobe Reader 6/7/8/9

14-6. 暗号化された Flash ファイルを配布する

AxProtectorで暗号化されたFlash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だけが必要です。

Windows OS(32bit)の場合
（Windows 2000/XP/Vista/7, Windows Server 2000/2003)
①32bit用コードメータランタイムキット "CodeMeterRuntime32.exe"（22MB)

Windows OS(64bit)の場合
(Windows XP/Vista/7 Server 2003/2008)
①64bit用コードメータランタイムキット "CodeMeterRuntime64.exe" (24MB)

動作環境
OS: Windows2000/XP/Vista/7 (32bit/64bit), Windows Server 2000/2003/2008 (32bit/64bit)
Adobe Flash Player 7/8/9/10

14-7. 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
① Mac用コードメータランタイムキット"CmRuntimeUser_4.20.264.500.dmg" (19MB)

14-8. Linux アプリケーションを配布する

RPM package for SuSe, Red Hat etc

32bit:
CodeMeter-4.20.264-500.i386.rpm
AxProtector-7.0.356-1.i386.rpm (AxProtectorで暗号化したアプリケーション用)

64bit:
CodeMeter64-4.20.264-500.x86_64.rpm
AxProtector64-7.0.356-1.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-4.20.264.500-i386.deb
codemeter-lite_4.20.264.500_i386.deb (Pure driver installation for systems without GUI)
axprotector-7.0.356_i386.deb (AxProtectorで暗号化したアプリケーション用)

64bit:
codemeter64_4.20.264.500_amd64.deb
axprotector64_7.0.356_amd64.deb (AxProtectorで暗号化したアプリケーション用)

14-9. Sun Solaris アプリケーションを配布する

SPARC / SPARCV9

CodeMeter Runtime
codemeter_4.20-sol-SPARC.tar.tar

CodeMeter Runtime 64-Bit Extension
codemeter_4.20-sol-SPARCV9.tar.tar

i386 / amd64

CodeMeter Runtime
codemeter_4.20-sol-i386.tar.tar

CodeMeter Runtime 64-Bit Extension
codemeter_4.20-sol-x64.tar.tar

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