

EASY-GUARD-KEY EG202

SOFTWARE PROTECTION TOOL
GUARD-KEY GENERATOR



SAVA Inc.

目次

製品内容について	2
1. 概要	3
2. フォルダファイルについて	3
3. 使い方	4
4. ガードキーの作成(Tools-USB フォルダ)	4
5. ライセンスファイルの作成(Tools-USB フォルダ)	5
6. ハードウェアがない場合のライセンスファイルの作成(Tools-USB フォルダ)	5
7. GenInfo.exe について(Tools-USB フォルダ)	5
8. ガードキーの複数使用、相性、セキュリティ	6
9. アクティベーション	6
10. 関数の使用方法	8
11. 6432Bit フォルダの中身	9
12. 国外での使用	9
13. ご注意	9
14. 修理・保証規定	9

製品内容について

マスターキーのほか、動作用サンプルが付属します。

EG202-32 又は EG202-6432、EG202A-6432 と書かれた USB メモリがマスターキーです。

プログラム: EG202 のプログラム、サンプル、ライブラリ、マニュアルはここに入っています。マスターキー上でプログラムを稼働させることもできますが、PC のハードディスクにコピーして使うことも出来ます。

書き換え禁止: マスターキーはフォーマットしたり、書き換えたりしないでください。マスターキーがない限り、EG202 の書き込みツールが動きません。

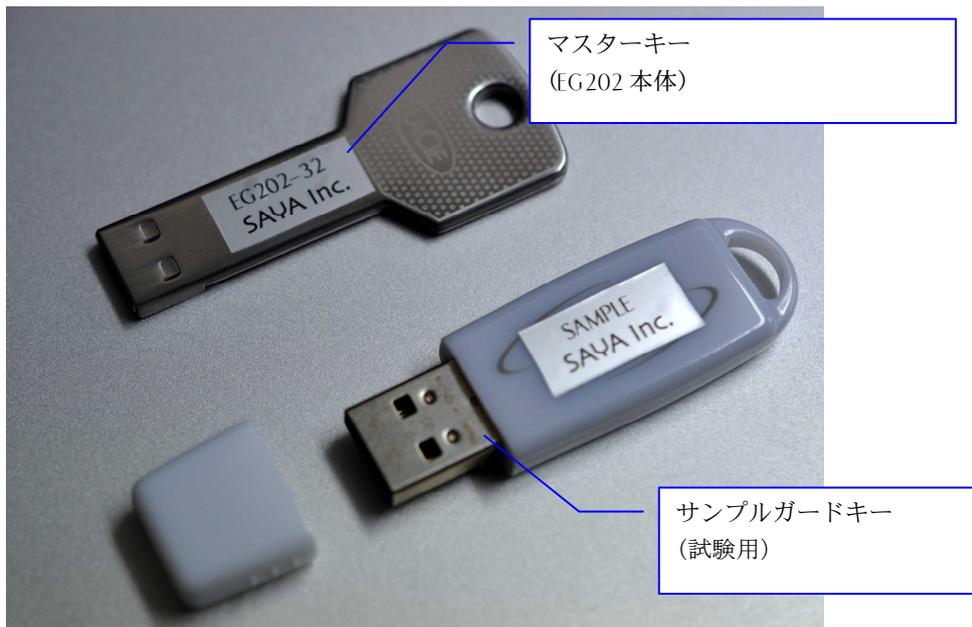
ベンダコード: マスターキーは、企業毎に異なるベンダコードが割り当てられます。これは、異なる企業のソフトウェアは、仮に同じユーザコード、ライセンスコードであっても動かないようにするためです。もし企業部署を超えて、同じベンダコードの EG202 の再発注が必要な場合には必ず、その旨を弊社にお伝えください。

SAMPLE と書かれた USB メモリがサンプルで製作したガードキーです。

この USB サンプルは以下の内容にてガードキー化しています。

ユーザコード 20105555 (Sample3 で、このユーザコードを認証しています)

ライセンスコード 10~14,80 を有効 (Sample1 でライセンスコード 80 を認証しています)



重要: EG202 はベンダ管理をしています

ガードキーはユーザごとに、独自のベンダコードが割り当てられています。ベンダコードがないと、EG202 を持つ別のユーザが、同じユーザコード、ライセンスコードを入手するなどすれば、保護したいアプリケーションを稼働させることが出来てしまうためです。

このため EG202 は暗号化されたベンダコードを組み込んでおり、出荷先毎にユニークなベンダコードを割り当てています。

もし EG202 を再購入される場合には、必ず以前と同じベンダコードを発行するよう弊社に要請してください。

また商社経由での再購入の場合、以前購入した企業名を提示してください。この場合、エンドユーザ直送か、電話確認の上で、同じ企業であることが証明されない限り、同じベンダコードでの再販はできません。

1.概要

EG202 は、ソフトウェアプロテクションの新しい提案です。従来のガードキーでは、ソフトウェアプロテクションの目的としては過剰すぎるほど、関数が膨大かつ難解でした。更に専用のハードウェアが必要、コストが高い、管理が面倒などの理由から、高額なソフトウェア以外での利用は困難でした。これに対しEG202では、わずか5つの関数と1つのツールでソフトウェアプロテクションを実現できます。EG202は、ライセンスファイル形式を採用し、最大16項目のライセンスの管理、複数のガードキーの取り扱いが可能です。EG202は市販のUSBメモリをガードキーとして使うため、大幅なコストダウンが可能です。

- ① 市販のUSBメモリをガードキーにさせることができる
- ② USBメモリのファイルコピー、デスクコピー、クローニングでは破れない強固なセキュリティ
- ③ 複数のガードキーの管理（この場合④のマルチライセンス管理は最も若いドライブのみが有効となる）
- ④ 1つのガードキーで最大16項目のライセンス管理
- ⑥ 一度ガードキーで登録したコンピュータは、ガードキーなしでも稼働できるアクティベーション（オプション）
- ⑦ 関数はたった5個
- ⑧ WindowsXP(32Bit)/WindowsVista(32/64Bit)/Windows7(32/64Bit)/Windows8(32/64Bit)/Windows8.1(32/64Bit)/Windows10(32/64Bit)対応

ラインナップ(機種名)

EG202HS32 32Bitモデル
 EG202HS6432 64/32Bitモデル(DLLは32Bit用、64Bit用で分かれています)
 EG202HA6432 64/32Bitモデル+アクティベーション機能(DLLは32Bit用、64Bit用で分かれています)

2.フォルダファイルについて

EG202はUSBメモリでソフトウェアを提供します。このUSBメモリはEG202自体のガードキー(マスターキー)を兼ねており、EG202の作業時にはこのUSBメモリをPCに実装していなければなりません。USBメモリには以下のファイル及びフォルダが格納されています。いずれもハードディスクの任意のフォルダにコピーして使うことが出来ます。

各フォルダの内容(機種により内容が異なります)

[全機種共通]

32Bit¥Sample1	: サンプルソース 32bit 版(VisualC++)です。ガードキーの検出方法を説明します。
32Bit¥Sample2	: サンプルソース 32bit 版(VisualC++)です。複数ガードキーの検出方法を説明します。
32Bit¥Sample3	: 実装方法を示したサンプルソース 32bit 版(VisualC#.net)です。
32Bit¥SampleVB6	: 実装方法を示したサンプルソース 32bit 版(VisualBASIC)です。
32Bit¥Tools-USB	: USBフラッシュメモリをガードキーにする為のツール 32bit 版です。
32Bit¥SDK	: VisualC++用のインポートライブラリ 32bit 版(EasyGuardEX.lib) : ヘッダファイル(EasyGuardEX.h) : VisualBASIC用の関数定義ファイル EasyGuardEX.bas
32Bit¥DLL-USB	: 配布用DLL 32bit 版です。(64bit版と同じDLL名です)
EasyGuard-EG202.pdf	: マニュアル
その他のファイル	: 絶対に消さないで下さい。

[EG202HA6432,EG202HS6432のみ]

64Bit¥Sample1	: サンプルソース 64bit 版(VisualC++)です。ガードキーの検出方法を説明します。
64Bit¥Sample2	: サンプルソース 64bit 版(VisualC++)です。複数ガードキーの検出方法を説明します。
64Bit¥Tools-USB	: USBフラッシュメモリをガードキーにする為のツール 64bit 版です。
64Bit¥SDK	: VisualC++用のインポートライブラリ 64bit 版(EasyGuardEX.lib) : ヘッダファイル(EasyGuardEX.h) : VisualBASIC用の関数定義ファイル EasyGuardEX.bas
64Bit¥DLL-USB	: 配布用DLL 64bit 版です。(32bit版と同じDLL名です)

[EG202HA6432のみ]

64Bit¥DLL-USB	: 配布用DLL 64bit 版です。(32bit版と同じDLL名です)
---------------	--------------------------------------

32bit版と64bit版について

上のフォルダで32bit以下に展開されているファイルは、32bit用です。

上のフォルダで64bit以下に展開されているファイルは、64bit用です。

EG202HS32には64bit以下のフォルダがありません。

EG202HS64、EG202HA6432には32bit、64bit双方のフォルダがあり、32bit、64bit両方に対応します。

USB上のファイルについて

USB上から実行することも出来ますが、誤って消去するなどの事故を防ぐため、使用するパソコンのハードディスクにコピーしてお使いいただくことを推奨します。コピー先は、どこにあっても、複数あってもよく、EG202-USBメモリさえあれば起動します。

3.使い方

EG202 を使って USB メモリドライブをガードキーに変換する手順は以下の通りです。

- ① **【認証コード組み込み】**お客様が保護したいソフトウェアの起動ルーチンに、関数 "bCheckGuardKeyEX" を実装してください。この関数はガードキーの有無を確認し true か false を返します。false の場合、ソフトウェアを起動させないようにします。VisualC++ で使う場合、SDK フォルダの EasyGuardEX.h をインクルード、EasyGuardEX.lib をプロジェクトに登録します。タイマーやコマンドボタンなどに関数 "bCheckGuardKeyEX" を実装しておく、常時ガードキーを監視できるので、より効果的です。ガードキーが外された場合、警告を出すなどしてソフトウェアを止めるなどの処理を実装します。なお複数のガードキーをハンドリングするためには関数 vCheckGuardKeyArray を使います。
- ② **【ライセンスファイルのハンドリング】**ソフトウェアによっては、機能を複数搭載し、エディション(購入価格)などで、機能の使用範囲を制限したい場合があります。このような場合、EG202 のライセンス機能を活用すれば安全なライセンス管理が可能です。ライセンス管理は、ガードキー(USB メモリ)上のライセンスファイルで行われ、専用ツール(Tools-USB フォルダの GenKeyEX.exe)で管理します。ライセンスファイルには暗号化された、16 個のライセンスコード、そのライセンスコードの有効/無効を記録されます。この状態は関数 bLicenseCheck で呼び出すことが出来ます。お客様のソフトウェアでは、ライセンスコードの状態、ソフトウェア機能を有効・無効判定を行います。ライセンス管理が不要であれば関数 bLicenseCheck の実装をしなくてもよいのです。尚 bLicenseCheck は最も若いガードキードライブのライセンス管理を制御するので、複数のガードキーのライセンス管理は出来ません。
- ③ **【ガードキーの作成】**ガードキー化したい USB メモリを接続し、Tools-USB¥GenKeyEX.exe でガードキー化してください。GenKeyEX.exe は弊社支給の EG202 付属の USB フラッシュメモリが専用のガードキーになっていますので、必ず EG202 付属の USB フラッシュメモリを接続した上で作業してください。ターゲットの USB メモリと EG202 付属の USB フラッシュメモリのドライブ番号を間違わないよう気をつけてください。ソフトウェアの使い方は 4 章に記載します。
- ④ **【ライセンスファイルの作成】**Tools-USB¥GenKeyEX.exe で機能名(文字列)、ライセンス番号(数字)を定義し、チェックボックスでライセンスの有効・無効を設定します。この内容は保存、読出しが可能です。ライセンスファイルの生成ボタンをクリックすると、ライセンスファイルはガードキーに生成されます。
- ⑤ **【配布】**お客様が保護したいアプリケーションは、②でガードキー化したガードキー(USB メモリ)なしには起動しません。この USB メモリと、DLL-USB 内にある配布用 DLL、そして①で作成したアプリケーションをエンドユーザに提供します。ガードキー自体には 32bit/64bit の区別はありませんので 32bit/64bit いずれのアプリケーションからでも使用することができます。

配布に必要なもの

1. USB フラッシュメモリ
2. DLL-USB の配布用 DLL バイナリ
3. 作成されたアプリケーション

4.ガードキーの作成(Tools-USB フォルダ)

ディスクドライブをガードキー化する場合、EG202-USB ガードキー(マスターキー)が必要です。

- ①EG202 の ToolS-USB フォルダは任意の場所にコピー可能です。但し、フォルダ内のファイル構造などは崩さないでください。
 - ②Tools-USB フォルダの GenKeyEX.exe を起動し、ガードキー化したい“ドライブ名 A~Z(半角英数字)”を入力します。
 - ③続いて“ユーザーコード 32Bit(半角英数字)”に 0~4294967295(半角英数字)の値を入れてください。この数値は関数 "bCheckGuardKeyEX" で取得できます。この値は通常アプリケーション毎にユニークな値を割り振り、アプリケーションの起動時に、“bCheckGuardKeyEX”でこの値が正しいかどうかチェックします。これにより同じ EasyGuard で生成された 2 つのアプリケーションにおいて、アプリケーション A のガードキーで、アプリ B が動いてしまう問題を回避することができます。
 - ④ディスクボリュームラベル名を入力してください。
 - ⑤”初期化実行”ボタンをクリックしてください。これでガードキー化を実行します。
 - ⑥”ガードキードライブの確認”ボタンでコンペアできます。
- ※②③④の値は保存され、次回起動時に継承されます。**

ドライブ番号 A~Z(半角英数字) ②

ユーザーコード 32Bit(半角英数字) 20106555 ③

初期化実行 ⑤

ガードキードライブの確認 ⑥

ボリュームラベル TEST_DRIVE ④

機能名	ライセンスコード	認証	機能名	ライセンスコード	認証
<input checked="" type="checkbox"/> デモ機能1	10	<input type="checkbox"/>		0	<input type="checkbox"/>
<input checked="" type="checkbox"/> デモ機能2	11	<input type="checkbox"/>		0	<input type="checkbox"/>
<input checked="" type="checkbox"/> デモ機能3	12	<input type="checkbox"/>		0	<input type="checkbox"/>
<input checked="" type="checkbox"/> デモ機能4	13	<input type="checkbox"/>		0	<input type="checkbox"/>
<input checked="" type="checkbox"/> デモ機能5	80	<input type="checkbox"/>		0	<input type="checkbox"/>
<input checked="" type="checkbox"/> デモ機能6	14	<input type="checkbox"/>		0	<input type="checkbox"/>
<input type="checkbox"/>	0	<input type="checkbox"/>		0	<input type="checkbox"/>
<input type="checkbox"/>	0	<input type="checkbox"/>		0	<input type="checkbox"/>

UserID1 0 UserID2 0

上の2つのUSBIDを使いライセンスファイル生成 ID書き出し ライセンスファイル認証確認 ライセンス管理保存

5.ライセンスファイルの作成(Tools-USB フォルダ)

- ①“機能名”(任意の文字列)と、それに対応する任意の“ライセンスコード”(半角 10 進数の値)を入力してください。機能名はライセンスコードが数値でわかり難いためにライセンスコードに割付けした名称です。実際のライセンス管理で使用されるのは、ライセンスコードのほうで、関数“bLicenseCheck”で、この値のライセンスの有効/無効を判定します。
- ②チェックボックスで ON/OFF させることで、機能コードに相当するライセンスを有効化、無効化します。
- ③“ライセンス管理保存”ボタンと“ライセンス管理読み込み”ボタンで、これら①②の状態を、保存・読み出しできます。
- ④各種設定が終了したら、“ライセンスファイル生成ボタン”をクリックしてください。ガードキー化した USB メモリにライセンスファイル license.dat を生成します。このファイルはテキストファイルではなく、暗号化されたバイナリファイルです。
- ⑤結果は、“認証ボタン”で確認できます。USB フラッシュメモリドライブのライセンスファイルを確認し、“認証”と記されている下に並んでいるスタティックテキストに、○(ライセンス有効)×(ライセンス無効)が表示されます。



6.ハードウェアがない場合のライセンスファイルの作成(Tools-USB フォルダ)

販売済みのソフトウェアをアップグレードする場合、毎回 USB ガードキーを回収してライセンス設定をやり直すのは、現実的ではありません。EG202 は USB ガードキーなしで、ライセンスファイルを生成できますので、メール等でライセンス更新が可能です。

- ①予め配布前のガードキーを USB に接続し、ID 書き出しボタンをクリックします。これで USB ガードキーに info.dat が生成されます。Info.dat はテキストファイルで、中身は以下のようになっています。(例)
User information, UserID1=f008d02a, UserID2=ae518578
Info.dat は GenInfo.exe を使い、配布済みのガードキーから生成させることも可能です。(GenInfo.exe は配布可能です)
- ②エディットボックス UserID1 と UserID2 に、Info.dat の同じキーの数値を 10 進数にして半角数値で入力します。
上の例では f008d02a と ae518578 なので、4027109418 と、2924578168 となります。
- ③上の 2 つの USBID を使いライセンスファイルを生成“ボタンをクリックします。Tools-USB フォルダに、license.dat が生成されますので、これをターゲットの USB ガードキー(例ユーザの USB)に上書きしてください。これでライセンスが更新されます。



7.GenInfo.exe について(Tools-USB フォルダ)

Tools-USB¥GenKeyEX.exeはライセンス管理用ツールなのでエンドユーザに配布することが出来ません。しか配布済み USB ガードキーから、Info.dat を回収したい場合があると思います。そのために配布可能な GenInfo.exe を用意しました。使い方は簡単で、USB ガードキを指した状態で、GenInfo.exe を起動し、ID 書き出しボタンをクリックするだけです。これで USB ガードキーに info.dat が生成されます。



8. ガードキーの複数使用、相性、セキュリティ

[複数使用] ライセンスファイルによる管理は、複数の USB ガードキーを挿した場合、最も若いドライブのみ有効になります。

[相性問題と原因] USBメモリによっては、正常に動作しない場合があります。これは規格を完全に準拠していなくても、ドライブとしては認識できたためです。このような一部の仕様を簡略化した製品が販売されている場合があります。従って、弊社は**全てのUSBメモリのガードキー化は保証しません**。必ず動作確認を行ってください。このようなデバイスであっても、EG-201 であれば Tools-ALL でガードキー化することができます。但しセキュリティレベルは低下し、ライセンス管理や複数キーの認識機能もありません。EG-201 については弊社までお問い合わせください。

[動作確認] コンピュータ A でガードキー化したものを、別のコンピュータ B で認識できるか確認します。確認は Tools-USB の Sample.exe を使い、初期化しているにも関わらず、ライセンスエラーとなれば相性問題が発生していると判断できます。

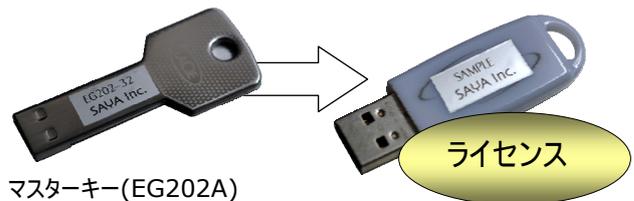
[セキュリティ] ガードキーは強力なセキュリティがかけられており、license.dat を除く、ルートにあるガードキー関連ファイルを別の USB メモリにコピーしても動作しません。またリカバリツールで丸ごとクローンを作成しても動作しません。

9. アクティベーション

EG202A-6432 のみ提供される機能です。次のような方法で、スマートなキーレスソフトウェアガード機能を実現します。分かりやすくするため、ガードキーの認証要素を“ライセンス”とすると、以下のような動作になります。

【ガードキーの作成～配布】

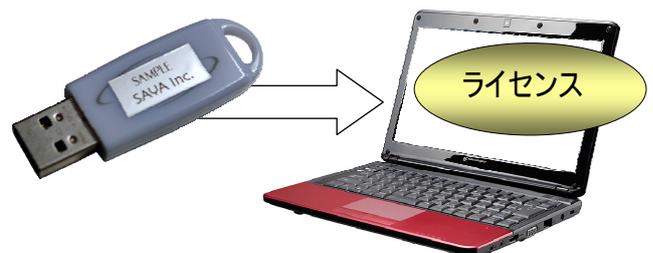
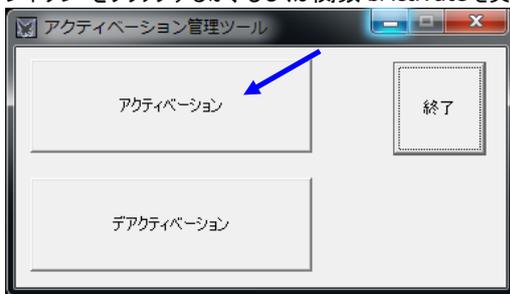
通常通りガードキーを作成、配布します。この時、Tools-USB¥active_pc.exe を一緒に配布するか、bActivate 関数および bDeActivate 関数をソフトウェアに組み込んでおきます。



マスターキー(EG202A)

【アクティベーション】

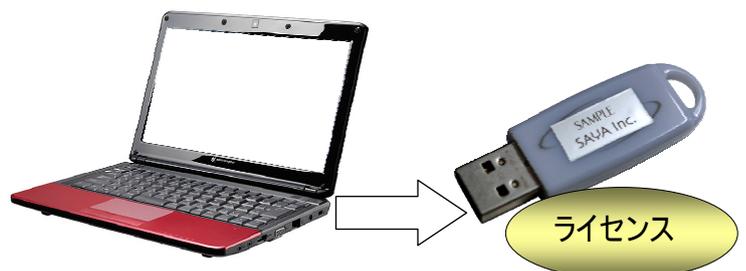
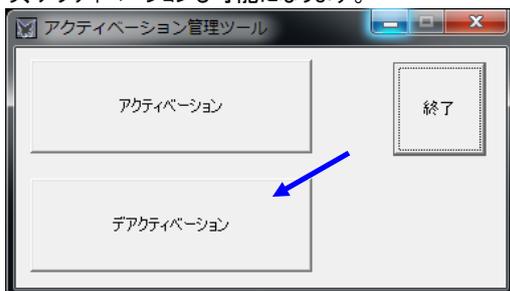
エンドユーザ様のコンピュータにおいて、ガードキーが認識している状態で、Tools-USB¥active_pc.exe を起動し“アクティベーションボタン”をクリックするか、もしくは関数 bActivate を実行します。これをアクティベーション化と呼びます。



アクティベーション化したコンピュータは、ガードキーを外してもガードキーがあるかのように振舞います。この間、ガードキーは無効になり、アクティベーション化されたコンピュータ以外では、ガードキーがないかのように振舞います。またこのガードキーではアクティベーションもできなくなります。このガードキーは次のデアクティベーションで必要になるので、絶対にフォーマットしたり、ファイルを消去しないでください。

【デアクティベーション】

再び通常のガードキー動作に戻すには、ガードキーをコンピュータに挿した状態で、Tools-USB¥active_pc.exe を起動して“デアクティベーションボタン”を実行します。もしくは関数 bDeActivate を実行します。これをデアクティベーション化と呼び、ガードキーは有効になり、アクティベーションも可能になります。



アクティベーションに関する補足事項

- (1) アクティベーション化できる機能は、関数 bCheckGuardKeyEX、bLicenseCheck のみで、vCheckGuardKeyArray はアクティベーションの影響を受けません。
- (2) 既にアクティベーション化されたコンピュータはデアクティベートせずにアクティベーションすることはできません。
- (3) ガードキーでアクティベーションできるコンピュータは 1 台だけですが、デアクティベートすることで、アクティベーションするコンピュータを移動することができます。
- (4) アクティベーション化されたコンピュータに、別のガードキーが装着されると、ガードキーを優先します。
- (5) アクティベーションしたコンピュータの、ハードウェア構成を変更する場合（ディスクや CPU やネットワークカードを交換する）、必ずデアクティベートして、ライセンスをガードキーに戻してください。ハードウェア構成が変更されると、認証できなくなったり、デアクティベートできなくなります。
- (6) 64bitOS で、32bit 用の EG202 モジュール(DLL)は動作します。しかし 64bit 版でアクティベートした場合、32bit 版アクティベーションアプリケーションは動作しませんし、その逆も駄目です。
- (7) 64bitOS で、32bit 用の EG202 モジュール(DLL)は動作します。しかし 64bit 版でアクティベートした場合、32bit 版デアクティベーションは動作しませんし、その逆も駄目です。
- (8) 64bitOS での 32bit モジュールの動作確認は行っておりますが、相性問題などが生じる恐れがあるので推奨しません。

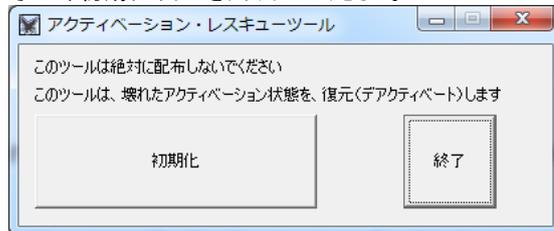
レスキューシステム

アクティベーションシステムはエラー訂正システムで保護されており、堅牢です。しかしアクティベーション化されたコンピュータが破損したり、ディスクに大きな部分破損（クラスタエラー）が生じると、デアクティベートできなくなります。この状態から復旧を可能にするツールが Tools-USB¥rescue.exe です。このプログラムはアクティベートしてライセンスを失ったガードキーに、再びライセンスを与えます。このツールは、ライセンスを、コンピュータに移動して、アクティベートできないはずのガードキーに再びライセンスを与えます。このため以下の点にご注意ください。

- 配布はしないでください。（マスターキーがないと動作しません）
- USB は現在安いので、複数配布するなどして、レスキューシステムの使用頻度を少なくしてください。
- レスキュー依頼が多いお客様には、アクティベーションを禁止したり、レスキューを有料化するなどの処置を行ってください。
- アクティベートされたコンピュータも初期化するので、コンピュータが不調の場合の復旧にも役立ちます。（エンドユーザー様への出向サポートが必要です）

使い方

回収した USB ガードキー、及びマスターキーをコンピュータに装着し Tools-USB¥rescue.exe を起動します。そして、初期化ボタンをクリックしてください。



以下のメッセージが出たらレスキュー成功です。



10. 関数の使用方法

bCheckGuardKeyEX

本関数はガードキードライブがあるか否か A～Z ドライブを調べます。複数のガードキーがある場合には、最も若いドライブを対象とし、それ以降は検証しません。

VC++、C++、C、C#

```
BOOL bCheckGuardKeyEX ( char *cDrive, LPDWORD lpdwUserCode, BOOL bPassFDD );
```

VB

```
Declare Function bCheckGuardKeyEX Lib "EasyGuardEX.dll" ( ByVal cDrive As String, _
    ByVal lpdwUserCode As Long, ByVal bPassFDD As Long) As Long
```

引数の詳細

cDrive	発見されたガードキードライブのドライブ番号の文字列が格納されます。(例"H:")
lpdwUserCode	埋め込まれているユーザコードへの符号なし 32Bit ポインタ。 これは第 4 章③に相当する数値です。
bPassFDD	旧バージョン EG-201 との互換性のために用意されています。本引数は意味がありません。

戻り値 VC++、C++、C、VB

: ガードキーの検索に成功すると TRUE、失敗すると FALSE が返ります。

bLicenseCheck

本関数はライセンスファイルを使い、ライセンスコードの有効、無効を確認します。
この関数を実行する前に、必ず bCheckGuardKeyEX を呼び出してください。

VC++、C++、C、C#

```
BOOL bLicenseCheck ( LPDWORD dwFunctionNum );
```

VB

```
Declare Function bLicenseCheck Lib "EasyGuardEX.dll" ( ByVal dwFunctionNum As Long) As Long
```

引数の詳細

dwFunctionNum	このライセンスコードが有効か、無効化を問い合わせます。 これは第 5 章①に相当する数値です。
---------------	--

戻り値 VC++、C++、C、VB

引数 dwFunctionNum に相当するライセンスが有効なら TRUE、無効なら FALSE が返ります。
これは第 5 章②に相当する数値です。

vCheckGuardKeyArray

本関数は C:¥～Z:¥を検索し、全てのガードキーの有無と、ユーザコードを取得します。

VC++、C++、C、C#

```
void vCheckGuardKeyArray ( LPDWORD lpdwUserCode, LPBOOL lpbValid );
```

VB

```
Declare Function vCheckGuardKeyArray Lib "EasyGuardEX.dll" _
    (ByRef lpdwUserCode As Long, ByRef lpbValid As Long)
```

引数の詳細

lpdwUserCode	ドライブ C:¥～Z:¥までのユーザコードが格納されます。DWORD 型配列の先頭が C:¥、2 番目がそして最後の D:¥、24 番目が Z:¥になります。(この引数は 24 個の DWORD 型配列となります)
lpbValid	ドライブ C:¥～Z:¥までのドライブがガードキーであるがどうかを、TRUE(ガードキーである)、FALSE(ガードキーではない)として、この配列に格納します。BOOL 型配列の先頭が C:¥、2 番目が D:¥、そして最後の 24 番目が Z:¥になります。(この引数は 24 個の BOOL 型配列となります)

bActivate

本関数は、アクティベートを実行します。EG202A-6432 のみ使用できます。

VC++、C++、C、C#

```
BOOL bActivate ( );
```

VB

```
Declare Function bActivate Lib "EasyGuardEX.dll" ( ) As Long
```

戻り値 VC++、C++、C、VB

成功すると TRUE、失敗すると FALSE が返ります。

bDeActivate

本関数は、ディアクティベートを実行します。EG202A-6432 のみ使用できます。

VC++、C++、C、C#

```
BOOL bDeActivate ( );
```

VB

```
Declare Function bDeActivate Lib "EasyGuardEX.dll" ( ) As Long
```

戻り値 VC++、C++、C、VB

成功すると TRUE、失敗すると FALSE が返ります。

11.6432Bit フォルダの中身

EG202 は 32Bit 版と 64Bit 版で同じ DLL 名を基本にしており、エクセルのマクロや AnyCPU コンパイルなど、実行コード長が未知のアプリに適応するには、そのアプリが何 32/64Bit いずれかのコードで動作しているかを判断 (Windows の API で簡単に判定できる仕組みがあります) した上で、32Bit または 64Bit の DLL を読み出さなければなりません。このような場合、アプリと同じフォルダに DLL がないと駄目なケースが幾つかあります。そこで EG202HA6432 では 32Bit と 64Bit で異なる名称の DLL にすることで、このようなアプリにも対応しています。

<32Bit 用の DLL、インポートライブラリ、VB 定義ファイル>

EasyGuard32.dll
EasyGuardEX32.dll
EasyGuardEX32.lib
EasyGuardEX64.bas

<64Bit 用の DLL、インポートライブラリ、VB 定義ファイル>

EasyGuardEX64.dll
EasyGuard64.dll
EasyGuardEX64.lib
EasyGuardEX64.bas

12. 国外でのご利用

フォルダ us-ver 以下のモジュールをご利用ください。

海外での利用では、日本語版では正常に認証しない可能性があります。

またパラメタシートは弊社までお問い合わせください。

13. ご注意

- ① ガードキーをフォーマットしたり、ガードキーの dfxcfx.ars、defrtt.axc、license.dat、*juts ファイル、マスターキーの*.bin ファイルを操作するとガードキー、マスターキーとして機能しなくなります。
- ② 複数のガードキーを装着した場合には、関数 bCheckGuardKeyEX、bLicenseCheck は、最も若いドライブを検索します。vCheckGuardKeyArray は全てのガードキーを検出 (ユーザコードを含む) できます。但し複数のガードキーのライセンスファイルは、ハンドリングできません。まとめると以下の通りです。

一つのガードキー	複数 (16 項目) のライセンス管理が可能。
複数のガードキー	最も若いドライブだけで、複数 (16 項目) のライセンス管理が可能。 それ以外のドライブではライセンス管理はできない。
- ③ Tools¥GenKeyEX.exe によるガードキーの生成は 1 台の PC に複数のガードキーを組み込んで作業を行う事が可能です。
- ④ EG202 は、EG201 と異なり USB フラッシュメモリ以外はガードキー化できません。誤ってハードディスクなどをガードキー化した場合には、該当ドライブの dfxcfx.ars、defrtt.axc、license.dat、*juts ファイルを削除してください。
- ⑤ USB メモリは一定の頻度で故障します。ユーザのガードキーが故障することを考慮して運用してください。

14. 修理保証規定

1. 不正コピーを防止する目的から、マスターキーの再発行はできません。マスターキーの取り扱いには十分ご注意ください。サンプルのガードキーはサービス品につき再発行できません。
2. マスターキー破損の場合には、再購入となります。
3. マスターキーをフォーマットした、あるいは書き換えた場合には、保証期間内は、購入価格の 70% で修理いたします。この場合、マスターキー、保証書を弊社までお送りください。弊社までの送料はお客様のご負担となります。
4. 保証期間内であっても、以下の場合には全額にさせていただきます。
 - ◎保証書のご提示がない場合。
 - ◎フォーマット・書き換えによる修理 (↑) を 2 回以上依頼した場合
 - ◎マスターキーが PC から認識しない場合や、マスターキーが物理的に破損している場合
 - ◎弊社から出荷していない同等品の USB メモリをご提示いただいた場合
5. EG202 は万全のセキュリティを施していますが、この分野は日進月歩であるため、完全で永久的なセキュリティを保証するものではありません。セキュリティが破られることに伴う損失は補償しません。