

# EASY-GUARD-KEY EG202

SOFTWARE PROTECTION TOOL  
GUARD-KEY GENERATOR



## SAVA Inc.

### 目次

<b>1.概要</b>	<b>2</b>
<b>2.フォルダファイルについて</b>	<b>2</b>
<b>3.使い方</b>	<b>2</b>
<b>4.ガードキーの作成(Tools-USB フォルダ)</b>	<b>3</b>
<b>5.ライセンスファイルの作成(Tools-USB フォルダ)</b>	<b>3</b>
<b>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)</b>	<b>3</b>
<b>7.GenInfo.exe について(Tools-USB フォルダ)</b>	<b>4</b>
<b>8.ガードキーの複数使用、相性、セキュリティ</b>	<b>4</b>
<b>9.関数の使用方法</b>	<b>4</b>
<b>10.オプション</b>	<b>5</b>
<b>11.ご注意</b>	<b>5</b>

## 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で、コストも高いなど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 3 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 USB メモリをガードキーとして使うので、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 3 個
- ⑧ WindowsXP(32Bit)/WindowsVista(32/64Bit)/Windows7(32/64Bit)対応

## 2.フォルダファイルについて

EG202 は CDROM ではなく 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 自体のガードキーをかかえているので、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できます。

Sample 1	: 実装方法を示したサンプルソース(VisualC++)です。 : このサンプルはライセンス管理方法、ガードキーの有無の検出方法について説明します。
Sample 2	: 実装方法を示したサンプルソース(VisualC++)です。 : このサンプルは複数ガードキーの検出方法について説明します。
Sample 3	: 実装方法を示したサンプルソース(VisualC#.net)です。
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です。
SDK	: インポートライブラリとヘッダファイル(関数の使用方法の解説を含む)です
DLL-USB	: 配布用 DLL です。Tools-USB で生成されたガードキーはこちらで認証します。
その他のファイル	: 絶対に消さないで下さい。

## 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があるか否かを確認し、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 の EasyGuardEX.h をインクルードして、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】** EG202 は、ライセンス管理機能を内蔵します。ライセンス管理は、ガードキー化した、USB メモリに置かれた、ライセンスファイルで行われます。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が記録され、新たに追加された API、bLicenseCheck から、ライセンスコードを読み出すことができます。この関数は引数で指定したライセンスコードが有効か、無効かを true/false で返します。お客様のソフトウェアでは、ライセンスコードの読み出しが成功したか否かで、ソフトウェアの機能を有効・無効を行います。(ライセンスファイルでソフトウェアの機能を制限します) ライセンスファイルの生成は Tools-USB¥GenKeyEX.exe によって行われます。ライセンス管理が不要であれば bLicenseCheck の実装、ライセンス管理は不要です。なお 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、読み出しが可能です。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。

### 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)

- 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 を起動します。まずガードキー化したい“ドライブ名 A～Z(半角英数字)”を入れてください。
- 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出来ます。
- ④”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。
- ⑤”ガードキードライブの確認”ボタンでコンペアできます。
- ⑥ディスクドライブをガードキー化する場合、SAYA 支給の、本ソフトウェア用の USB ガードキーが必要です。



#### 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)、“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あり、わかりづらいため、ライセンスコードをネーミングしたものです。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”にこの数値に相当するライセンスが有効か、無効かを確認できます。
- 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、“ライセンス管理読み込み”ボタンで、機能名、ライセンスコード、チェックボックスの状態を、保存・読み出してきます。
- 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 lisence.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テータックテキストに、○(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ライセンスをアップグレードする場合など、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 では、USB ガードキーなしで、ライセンスファイルを生成することができますので、メールなどで送信して、ライセンスの更新が可能です。

- 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

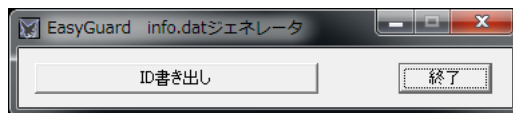
Info.dat は、GenInfo.exe を使い、配布済みのガードキーから生成させることも可能です。(GenInfo.exe は配布可能です)

- 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）に書き込んでください。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 ガードキーを指した状態で、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.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**

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

#### 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が格納されます。  
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。  
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 : 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、ライセンスの有効、無効を確認します。

### VC++、C++、C

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

#### 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。

**戻り値** VC++、C++、C : 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。

## vCheckGuardKeyArray

本関数は C:¥~Z:¥を検索し、ガードキーの有無と、ユーザコードを取得します。複数のガードキーを使った場合、最も若いドライブしか使うことが出来ません。

### VC++、C++、C

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

#### 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)

**戻り値** VC++、C++、C : なし。

## 10. オプション

### ライセンスファイルの拡張

ライセンスファイル 255 項目まで拡張させることが可能です。

### アクティベーション

最初にガードキーを装着、アクティベーションすると、その PC はガードキーなしでも、ライセンス管理が可能です。

アクティブにできる PC は 1 台だけで、他の PC をアクティブにするには、現在アクティブな PC をノンアクティブにする必要があります。

これらの管理を一括して行えます。

## 11. 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 ファイルを操作するとガード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。  
このような場合、vCheckGuardKeyArray を使うことでユーザコードを取得できます。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 ユーザコードが同じガードキーで bCheckGuardKeyEX を実行すると、戻り値も、lpdwUserCode も同じ状態になります。
- ⑤ EG202 は USB フラッシュメモリ以外はガードキー化できません。EG201 には多くのメディアをサポートするバージョンが同梱されています。