

# *ROCKEY2*

## 開発者ガイド v2.0

---



Feitian Technologies Co., Ltd.  
Software Developer's Agreement

---

All Products of Feitian Technologies Co., Ltd. (Feitian) including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. Allowable Use – You may merge and link the Software with other programs for the sole purpose of protecting those programs in accordance with the usage described in the Developer's Guide. You may make archival copies of the Software.
2. Prohibited Use – The Software or ROCKEY2 hardware dongle or any other part of the Product may not be copied, reengineered, disassembled, decompiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software's source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Feitian provided enhancement or upgrade to the Product.
3. Warranty – Feitian warrants that the ROCKEY2 dongles and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.
4. Breach of Warranty – In the event of breach of this warranty, Feitian's sole obligation is to replace or repair, at the discretion of Feitian, any Product free of charge. Any replaced Product becomes the property of Feitian.

Warranty claims must be made in writing to Feitian during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Feitian. Any Products that you return to Feitian, or a Feitian authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. Limitation of Feitian's Liability – Feitian's entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall Feitian be liable for any damages caused by your failure to meet your obligations, nor for any loss of data, profit or savings, or any other consequential and incidental damages, even if Feitian has been advised of the possibility of damages, or for any claim by you based on any third-party claim.
6. Termination – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

Feitian Technologies (以下”Feitian”と略す) は、このドキュメントの内容の充実に全力を尽くしておりますが、このドキュメントの誤りや省略からの、いかなる種類の直接あるいは間接的な損失および損害について一切責任を負いません。また、本書は必ずしも ROCKEY ハードウェアあるいはソフトウェアに関する更新のすべてについて言及しているわけではありません。

#### <連絡先>

飛天ジャパン株式会社 (Feitian Japan Co., Ltd.)

Tel: 03-3668-6668

Fax: 03-3668-6667

Add: 東京都中央区日本橋小網町 9-3 CANAL TOWER 4 階

Zip Code: 〒103-0016

URL: <https://ftsafe.co.jp/>

Feitian Technologies Co., Ltd.

Tel: +86-10-62360800, 62360900

Fax: +86-10-82070027

Email: [feitian@public3.bta.net.cn](mailto:feitian@public3.bta.net.cn)

Add: 3rd Floor, Building No. 5, Jimen Hotel, Xueyuan Road, Haidian District, Beijing, PRC

Zip Code: 100088

URL: <http://www.FTsaf.com>

#### EC Attestation of Conformity



ROCKEY is in conformity with the protection requirements of CE Directives 89/336/EEC Amending Directive 92/31/EEC. ROCKEY satisfies the limits and verifying methods: EN55022/CISPR 22 Class B, EN55024: 1998.

#### FCC Standard

This device is in conformance with Part 15 of the FCC Rules and Regulation for Information Technology Equipment.

Operation of this product is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

#### USB



This equipment is USB based.

変更履歴：

改定年月	バージョン	内容
2007 年 11 月	1.0	初版
2008 年 8 月	1.1	RY2_Open と RY2_GenUID の説明内容追加
2009 年 6 月	1.2	.Net インベロープツールの関数選択方法を修正
2010 年 2 月	1.3	Setup.exe 関連内容を削除
2010 年 4 月	1.4	インベロープツールのバージョンアップに伴う利用方法の更新
2010 年 10 月	1.5	文章の校正（全文見直し）
2010 年 11 月	1.5.1	本社移転に伴う連絡先変更
2011 年 9 月	1.5.2	初期化ツールに HID 取得機能追加
2011 年 9 月	1.5.3	初期化ツールに HID 取得機能変更
2020 年 12 月	1.5.4	<ul style="list-style-type: none"> <li>・ [1.3] の [各種オペレーティングシステムのサポート] に Win10 を追加</li> <li>・ [第 6 章 ROCKEY2 インベロープツール] 全体見直し</li> <li>・ [附録 A CD-ROM の収録] 見直し</li> </ul>
2022 年 7 月	1.5.5	<ul style="list-style-type: none"> <li>・ [1.3] の [各種オペレーティングシステムのサポート] に Win11 を追加</li> <li>・ [6.2 インベロープする際によくあるご質問 (FAQ)] を追加</li> <li>・ 文章の校正（全文見直し）</li> </ul>
2022 年 7 月	1.5.6	<ul style="list-style-type: none"> <li>・ 7.1.3 RY2_Close 戻り値を訂正</li> </ul>
2025 年 2 月	2.0	<ul style="list-style-type: none"> <li>・ 全文見直し</li> <li>・ [6.2 インベロープツールのコマンドライン]、[6.3 注意事項]、[第 8 章 実用前に検討すべき事項]、[9.4 API に関する FAQ] を追加</li> </ul>

## はじめに

この度は、ROCKEY2 製品をご利用いただき、誠にありがとうございます。

ROCKEY2 は、ソフトウェアの不正コピーやライセンスの不正使用を防止するために設計された USB ドングルです。このドングルを用いることで、ソフトウェアはドングルが接続されている場合のみ使用可能となり、接続されていない場合には使用できない仕組みを実現します。

本ガイドでは、ROCKEY2 の基本的な概念、設定方法、各種ツール、および API の使用方法について解説します。以下に、本ガイドを読む前に役立つ重要なポイントを記載しましたので、ご確認ください。

### 1. ドングルについて

この文書における「ドングル」とは、ROCKEY2 を指します。

### 2. 本ガイドの対象

本ガイドは、ソフトウェア（EXE や DLL）の利用を制限したいと考えているソフトウェア開発者および販売者を対象としています。

### 3. 開発キット（SDK）と提供内容

ROCKEY2 を最大限に活用するために、開発キット「Software Developer's Kit（SDK）」を提供しています。

SDK には以下の内容が含まれています：

- ・ ROCKEY2 ドングル（1 本）：販売用のドングルと同一のものです。
- ・ SDK（各種ドキュメント、製品関連ツール、API ライブラリ、サンプル等を含む）  
※詳しくは『第 3 章 ROCKEY2 開発パッケージ』をご参照ください。
- ・ 製品関連資料

ROCKEY2 を正式購入する前に、[弊社のウェブサイト](#)から無償の評価キットをお申し込みいただけます。

### 4. 初めて ROCKEY2 を利用する場合

ROCKEY2 を初めて利用する場合は、以下のステップを実施することをお勧めします。

#### Step1: 基本概念の理解

ROCKEY2 の基本的な概念を理解します。

※詳しくは『第 4 章 ROCKEY2 の概念』をご参照ください。

#### Step 2: ドングルの初期化

- ① SeedCode の指定と UID の生成（必須）

ドングルを初期化する際に、SeedCode を指定して UID を生成します。

- ② データゾーンへのデータ書き込み（任意）

必要に応じて、データゾーンにカスタムデータを書き込みます。

※詳しくは『第 5 章 ROCKEY2 初期化ツール』をご参照ください。

### **Step 3: プロテクト方式の選択**

ROCKEY2 を使用してソフトウェアをプロテクトする際、以下の 2 つの方式から選択します。

#### ■ エンベロープツール方式

プロテクトツールを利用して、ソフトウェアを簡単に保護します。

※詳しくは『第 6 章 ROCKEY2 エンベロープツール』をご参照ください。

#### ■ API 方式

SDK で提供される関数（API）を利用し、カスタマイズ可能なプロテクトを実現します。

※詳しくは『第 7 章 ROCKEY2 の関数（API）』および SDK¥Windows¥Samples をご参照ください。

### **Step 4: 実際運用に向けた検討事項**

ROCKEY2 を運用する前に、以下の関連事項について検討します。

※詳しくは『第 8 章 実用前に検討すべき事項』をご参照ください。

## **5. お問い合わせ先**

製品に関するご質問がございましたら、弊社の[製品のお問い合わせページ](#)よりお問い合わせください。

# 目次

<b>第 1 章 概要 .....</b>	<b>2</b>
1.1 ROCKEY2 の機能 .....	2
1.2 ROCKEY2 によるソフトウェアのプロテクト方式.....	2
1.3 製品特徴 .....	3
1.4 使用上のご注意.....	3
<b>第 2 章 ROCKEY2 ハードウェアの特徴.....</b>	<b>4</b>
2.1 ROCKEY2 の内部構造.....	4
2.2 ROCKEY2 ハードウェア・インターフェース .....	4
2.2.1 USB インタフェース.....	4
2.2.2 専用ドライバが不要 .....	4
2.2.3 LED .....	5
<b>第 3 章 ROCKEY2 開発パッケージ .....</b>	<b>5</b>
3.1 CD-ROM の内容 .....	5
3.2 最新版の入手方法 .....	5
<b>第 4 章 ROCKEY2 の概念 .....</b>	<b>6</b>
4.1 ユーザーID (UID) と Seed コード.....	6
4.2 ハードウェア ID (HID) .....	6
4.3 ユーザーデータゾーン.....	7
<b>第 5 章 ROCKEY2 初期化ツール.....</b>	<b>8</b>
5.1 ユーザーID (UID) の生成 .....	8
5.2 ハードウェア ID (HID) の取得 .....	9
5.3 ユーザーデータゾーンへの書き込み .....	10
5.4 ユーザーデータゾーンの読み出し.....	11
<b>第 6 章 ROCKEY2 インベロープツール .....</b>	<b>13</b>
6.1 インベロープツールの利用方法 .....	13
6.1.1 事前準備 .....	13
6.1.2 インベロープ方法.....	14
6.1.3 動作確認 .....	19
6.1.4 配布 .....	20
6.2 インベロープツールのコマンドライン .....	21
6.2.1 RockeyCmdShell の種類 .....	21
6.2.2 RockeyCmdShell の起動及び基本のパラメーター .....	21
6.2.3 ROCKEY2 製品に関するパラメータ.....	22
6.2.4 PE アプリケーションに関するパラメータ.....	23
6.2.5 Microsoft .NET アプリケーションに関するパラメータ .....	23
6.2.6 ROCKEY2 のコマンド実行例.....	24
6.2.6.1 ROCKEY2 で PE32 を保護する場合.....	24
6.2.6.2 ROCKEY2 で PE64 を保護する場合.....	24
6.2.6.3 ROCKEY2 で.NET32 を保護する場合 .....	25
6.2.6.4 ROCKEY2 で.NET64 を保護する場合 .....	25
6.3 注意事項 .....	25
<b>第 7 章 ROCKEY2 の関数 (API) .....</b>	<b>26</b>

7.1	Rockey2 API インターフェース .....	26
7.1.1	RY2_Find : 接続された ROCKEY2 を検索する .....	26
7.1.2	RY2_Open : 指定された ROCKEY2 をオープンする .....	26
7.1.3	RY2_Close : 指定された dongle を閉じる .....	27
7.1.4	RY2_GenUID : ユーザー ID を生成する .....	27
7.1.5	RY2_Read : dongle のデータを読み出し .....	27
7.1.6	RY2_Write : dongle にデータを書き込み .....	27
7.1.7	エラーコード .....	28
7.2	基本的なサンプル .....	28
7.2.1	Step0--オリジナルプログラム .....	29
7.2.2	Step1--dongle の初期化 (ユーザー ID の生成) .....	29
7.2.3	Step2-- 指定された UID で dongle を開く .....	30
7.2.4	Step3-- dongle のユーザーデータゾーンにデータを書き込む .....	31
7.2.5	Step4-- dongle のユーザーデータゾーンの内容を読み出す .....	32
7.3	総合応用サンプル .....	33
7.3.1	Step5-- 一つの ROCKEY2 の複数データゾーンへの書き込み/読み出し .....	33
<b>第 8 章 実用前に検討すべき事項 .....</b>		<b>35</b>
8.1	ソフトウェアと ROCKEY2 の紐付けを計画 .....	35
8.2	自社社内専用初期化ツールの作成 .....	35
8.3	プロテクト方式の選択及び注意点 .....	36
<b>第 9 章 FAQ .....</b>		<b>37</b>
9.1	トラブルシューティングの確認事項 .....	37
9.2	基本の運用に関する FAQ .....	37
9.2.1	プロテクトをかけるプログラム数に制限はあるでしょうか? .....	37
9.2.2	プロテクトされたソフトウェアは複数の dongle に対応できますか? .....	37
9.2.3	プロテクトをかけた EXE や DLL 自体の不正コピーは制限できますか? .....	37
9.3	エンベロープツールに関する FAQ .....	38
9.3.1	[Result: failed, error code:100, error message Find Dongle Failed] エラー .....	38
9.3.2	[Result: failed, error code:100, error message Open Dongle Failed] エラー .....	38
9.3.3	[Result: failed, error code:57, error message ACCESS TARGET FILE FAIL] エラー .....	38
9.3.4	[Result: failed, error code:48, error message PE SAVE FAIL] エラー .....	39
9.3.5	エンベロープされたアプリを実行すると [runtime.dll が見つからない] エラー .....	39
9.3.6	ソフトウェアがエンベロープされた後、起動できません .....	39
9.3.7	Linux のエンベロープツールについて .....	40
9.4	API に関する FAQ .....	40
9.4.1	ROCKEY2 の API をどのように呼び出しますか? .....	40
9.4.2	.NET アプリケーションではどのライブラリを利用できますか? .....	40



## 第1章 概要

---

### 1.1 ROCKEY2 の機能

ROCKEY2 は、ソフトウェアの著作権保護とライセンス管理目的とした USB ドングル（ハードウェアキー）です。ROCKEY2 の専用エンベロープツールや API を利用してソフトウェアをプロテクトすると、ソフトウェアはドングルを持つ顧客のみ使用可能となり、開発者はソフトウェア資産を不正コピーや不正使用から守り、ライセンス管理の強化を実現できます。

### 1.2 ROCKEY2 によるソフトウェアのプロテクト方式

ROCKEY2 を利用してソフトウェアのプロテクトするのは、ROCKEY2 のプロテクト方式は 【エンベロープツール方式】 と 【API 方式】 の 2 つを提供します。

#### ■ 【エンベロープツール方式】

ROCKEY2 のエンベロープツール（Envelope.exe）を使用して、ソフトウェアを保護する方法です。既存のソフトウェアやライブラリ（\*.exe/\*.dll/\*.ocx）を修正せずに迅速かつ簡単にプロテクトできます。

ソースコードを持っていないソフトウェアや ROCKEY2 の API に慣れていない開発者にとって効果的な方式です。

（詳細は、『第 6 章 ROCKEY2 エンベロープツール』を参照してください）

#### ■ 【API 方式】

ソフトウェアのソースコードに ROCKEY2 の通信インターフェース（API）を埋め込んで、ソフトウェアを保護する方式です。多くの API を使用することによって、柔軟かつ強力なプロテクトを実現できます。

Excel や Word 等 VBA に組み込むと、ファイルを保護することも可能です。

本マニュアルに VC++ のサンプルプログラムを掲載しています、他の開発言語のサンプルは SDK の Samples ディレクトリを参照してください。

（詳細は、『第 7 章 ROCKEY2 の関数（API）』を参照してください）

尚、ROCKEY2 のエンベロープツールと API の両方を結合したセキュリティシステムは、より高度なプロテクションを実現できます。

## 1.3 製品特徴

### ✓ シンプル&使いやすい

ROCKEY シリーズの中で最もシンプルな機能を備えており、簡単にソフトウェアを保護できます。サーバ運用が不要で、コストを抑えて導入が可能です。

### ✓ 高い互換性

ROCKEY2 は、USB 接続機器としての基本的な互換性に加え、同一タイプのdongleを複数接続するカスケード接続にも対応しています。

### ✓ 高速処理能力

ROCKEY2 は最適化されたアルゴリズムを内蔵しており、ソフトウェアのパフォーマンスに影響を与えることなく、強力な保護機能を提供します。

### ✓ 高い信頼性

ROCKEY2 は、全世界でユニークなハードウェア ID を持つことを保証し、不正コピーからソフトウェア資産を守ります。このユニークな ID は、dongleの CPU に焼き付けられており、変更や複製が不可能です。

### ✓ 広範な OS サポート

ROCKEY2 でプロテクトされたソフトウェアは、Windows (Windows 7/8/8.1/10/11、Server 2008(R2)/2012(R2)/2016/2019/2022)、Linux、MacOS の各 OS で実行可能です。

### ✓ 豊富なプログラミング言語インターフェース

ROCKEY2 は、VC++、C#、VB、VB.NET、Java、Python、Delphi、VBA など、広範なプログラミング言語に対応しています。開発者はサンプルを参照して、既存のプロジェクトや環境に容易に ROCKEY2 を統合でき、開発プロセスを迅速化することが可能です。

## 1.4 使用上のご注意

- ・ 本製品は USB 給電を利用します、パソコンがスリープモードに USB 給電が切れた場合、正しく動作できない場合があります。
- ・ 本製品を抜き差しするときは、必ず筐体部分を持ち、無理な力を加えないでください。故障・破壊などの原因となることがあります。
- ・ 本製品をパソコンなどに斜めに挿したり、引っ張ったりすると、本製品やパソコンなどの端子を壊す恐れがあります。充分にご注意ください。
- ・ 分解したり、改造したりしないでください。改造などにより、本製品やパソコンなどに不具合が生じても、当社は責任を負いかねます。予めご了承ください。

## 第2章 ROCKEY2 ハードウェアの特徴

### 2.1 ROCKEY2 の内部構造

ROCKEY2 のコア部分には、USB インターフェース機能を備えた専用 CPU が搭載されています。この CPU には EPROM メモリが内蔵されており、データはこのメモリに保存されます。そのため、予期せぬ電源オフが発生してもデータが失われることはありません。

ROCKEY2 のメモリは、通常の EPROM タイプのdongleよりも大容量の保存領域を提供しています。ユーザーデータゾーンは 5 つに分かれており、合計で 2,560 バイトの容量を持ちます。開発者は、このユーザーデータゾーンを活用して、ソフトウェアに関連する重要な情報（例：シリアルナンバーなど）を安全に格納することができます。

### 2.2 ROCKEY2 ハードウェア・インターフェース

#### 2.2.1 USB インタフェース

ROCKEY2 は USB（Type-A）インターフェース方式を採用しており、プラグ・アンド・プレイに対応しています。随時コンピューターに挿し込んだり抜いたりできます。ただし、注意が必要なのは、ソフトウェアが作動中にdongleを抜いてしまうと、システムが停止してしまう可能性があるため、十分にご注意ください。

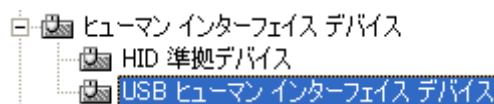
USB（Type-C）の端末に利用するには、データ転送機能を持つ変換アダプタを別途利用してください。

ROCKEY2 は USB 1.1/2.0/3.x をサポートしています。USB dongleは USB ハブを介して同時に最大 16 個まで使用することも可能です。

#### 2.2.2 専用ドライバが不要

Windows や Linux などの OS が標準で搭載している USB ヒューマンインターフェースのドライバで動作します。専用のデバイスドライバは必要ありません。

例えば、WindowsOS の端末に接続すると、デバイスマネージャに"USB ヒューマンインターフェイスデバイス"が表示されます。



### 2.2.3 LED

ROCKEY2 USB の LED ランプによって、dongle の状態を確認できます。

- ・ 点灯：正常
- ・ 一時点滅：データ通信中
- ・ 連続点滅：正しく認識されていない、またはハードウェアが故障している可能性があります。
- ・ 消灯：通電していない、またはハードウェアが故障している可能性があります。

## 第3章 ROCKEY2 開発パッケージ

---

ROCKEY2 の開発パッケージは SDK に同梱された CD-ROM に格納されています。

### 3.1 CD-ROM の内容

<ROCKEY2 SDK>

<Linux>	ROCKEY2 Linux 環境のドライバー、API 及びサンプル
<MacOS>	ROCKEY2 MacOS 環境の API 及びサンプル
<Windows>	ROCKEY2 Windows 環境の API 及びサンプル
└<Api32>	ROCKEY2 Win32API
└<Api64>	ROCKEY2 Win64API
└<Docs>	ROCKEY2 ドキュメント（英語版）
└<Include>	ROCKEY2 関数ヘッダファイル
└<Samples>	ROCKEY2 各開発言語のサンプル
└<Utilities>	ROCKEY2 ツール
└└<Envelope>	ROCKEY2 エベローブツール
└└<Init>	ROCKEY2 初期化ツール

<日本語ドキュメント>

ROCKEY2 クイックスタートガイド	ROCKEY2 クイックスタートガイド
ROCKEY2_マニュアル	ROCKEY2 日本語マニュアル

- ※ 詳細は、CD-ROM に収録されたリリースノートをご確認ください。
- ※ CD-ROM の内容は圧縮されていないため、開発者は必要なファイルをローカルにコピーしてご利用ください。
- ※ 英語版マニュアルは若干古いため、日本語マニュアルをご参照ください。

### 3.2 最新版の入手方法

最新版は下記弊社のウェブサイトよりダウンロードできます：

<https://ftsafes.co.jp/support/download/rockey2/>

## 第4章 ROCKEY2 の概念

本章では、ROCKEY2 のソフトウェアプロテクションにおける基本概念と機能について説明します。

### 4.1 ユーザーID (UID) と Seed コード

ユーザーID (UID) は、ROCKEY2 のパスワードとして機能し、dongleの検索やデータの読み書きに使用されます。

ROCKEY2 が出荷される際、ユーザーID はデフォルトで「0」に設定されています。使用する前に、ROCKEY2 の初期化ツールや API を使用して、Seed コードを指定し、新しいユーザーID を生成する必要があります。

Seed コードは、UID を生成するための「鍵」となる文字列です。以下の要件を満たす文字列を自由に設定できます：

- ・ 半角英文字（大文字・小文字の区別あり）、数字、記号で構成。
- ・ 最大 64 バイトの文字列。

ROCKEY2 には独自のハッシュアルゴリズムが組み込まれており、このアルゴリズムを使用して指定された Seed コードから UID を生成します。同じ Seed コードを使えば、必ず同じ UID が生成されます。ただし、UID から Seed コードを逆算することはできません。そのため、他のdongleの UID が分かっても、Seed コードがないと同じ UID を生成することはできません。

- ※ Seed コードは、UID を生成するための非常に重要なデータです。必ず記録し、安全に保管してください。Seed コードを紛失すると、同じ UID を再生成することはできません。そのため、Seed コードの管理には十分ご注意ください。
- ※ 生成された UID が負の値（例: 「-1259880565」）になる場合があります。負の UID を使用する際は、マイナス記号「-」も含めて入力してください。

### 4.2 ハードウェア ID (HID)

ハードウェア ID (HID) は、ROCKEY2 dongleにおける固有の識別番号で、工場出荷時に設定されます。各 ROCKEY2 の HID は一意であり、初期化ツールや API 関数 (RY2\_Open) を使用して取得できます。

- ※ HID を正しく取得するには、dongleにユーザーID (UID) を設定する必要があります。
- ※ 工場出荷時の ROCKEY2 では、ユーザーID (UID) がデフォルト値「0」に設定されています。この状態では HID も「0」として取得されます。正しい HID を取得するため、事前に UID を設定してください。

ROCKEY2 を利用してソフトウェアを保護する場合、特定の HID を指定することで、特定のdongleのみ動作するように設定できます。

### 4.3 ユーザーデータゾーン

ROCKEY2 には、内部に 5 つのデータゾーンがあり、各データゾーンの容量は 512 バイトで、合計 2,560 バイトです。

初期化ツールや API 関数を使用することで、各データゾーンにデータの読み書きが可能です。たとえば、ソフトウェアの実行に必要な重要な情報を ROCKEY2 のデータゾーンに書き込むことで、ソフトウェアの正当な利用を検証する仕組みを構築できます。

※ ユーザーデータゾーンを使用する場合は、[API 方式] を利用してください。[エンベロープツール方式] では、ユーザーデータゾーンの機能は利用できません。

## 第5章 ROCKEY2 初期化ツール

ROCKEY2 初期化ツール（¥Windows¥Utilities¥Init¥Init ROCKEY2.exe）を使用することで、以下の機能を実現できます（図 5-1 参照）：

- ✓ [Generate UID]： dongle のユーザー ID（UID）を生成します。
- ✓ [HID]： dongle のハードウェア ID を取得します。
- ✓ [Write]： ユーザーデータゾーンにデータを書き込みます。
- ✓ [Read]： ユーザーデータゾーンにデータを読み出します。

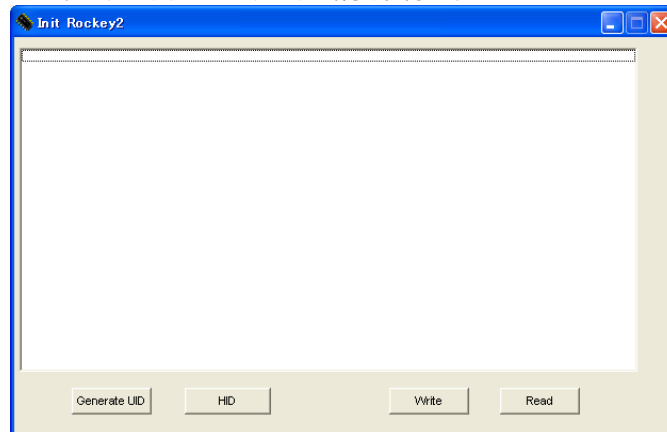


図 5-1

以降では、各機能について詳細に説明します。

### 5.1 ユーザー ID（UID）の生成

ROCKEY2 のユーザー ID（UID）は、初期化ツールを使用して生成できます。以下の手順に従って操作してください。

<手順>

- 1、ROCKEY2 を PC に接続します。
- 2、初期化ツールの「Generate UID」（UID 生成）をクリックします。  
図 5-2 のように、接続されている ROCKEY2 の数が「Find x ROCKEY2」として表示されます（x は ROCKEY2 の個数）。  
また、Seed コードの入力画面が表示されます。

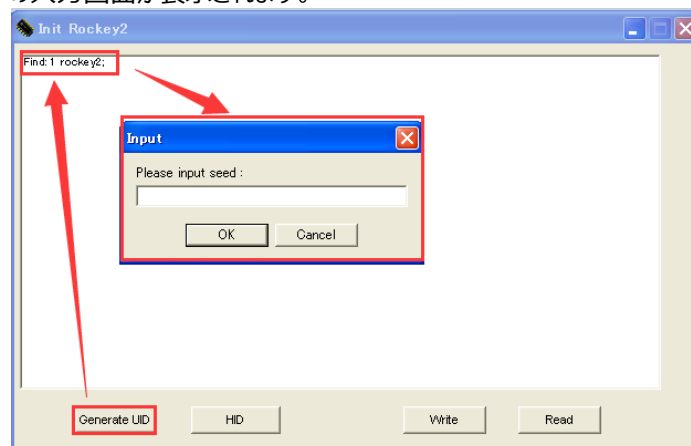


図 5-2

- 3、Seed コードを入力してユーザーID（UID）を生成します。
  - ※ Seed コードに、64 バイト以下の英数字や記号を使用してください。
  - ※ 同じ Seed コードを設定すると、必ず同じユーザーID が生成されます。

例：Seed コードに「12345」を入力し、「OK」をクリックすると、UID「715400947」が生成されます（図 5-4 参照）。生成された UID を有効にするためには、ROCKEY2 を一度コンピュータから取り外し、再度接続してください。

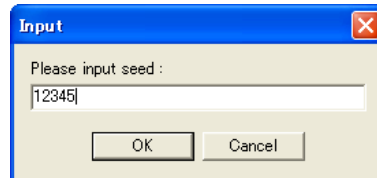


図 5-3

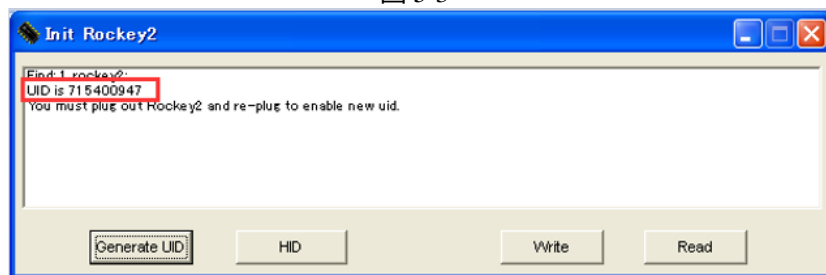
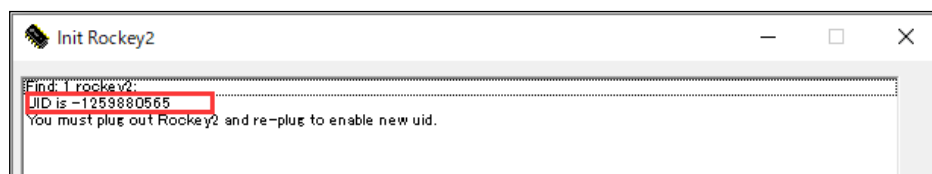


図 5-4

- ※ UID を複数回生成することができます。但し、UID を新たに生成すると、ROCKEY2 のユーザーデータゾーンに保存されたすべてのデータが削除されます。
  - ※ 生成された UID が負の値（例：「-1259880565」）になる場合があります。負の UID を使用する際は、マイナス記号「-」も含めて入力してください。
- 例えば、Seed に abcde123 を指定すると、生成された UID は「-1259880565」です。



## 5.2 ハードウェア ID（HID）の取得

ROCKEY2 のハードウェア ID（HID）は、ユーザーID（UID）が生成された後に取得できます。以下の手順で操作してください。

<手順>

- 1、ROCKEY2 を PC に接続します。
- 2、初期化ツールの [HID] をクリックします。ROCKEY2 のハードウェア ID が表示されます。



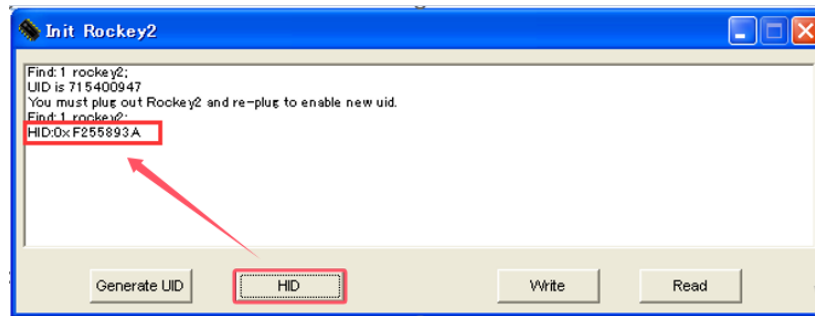


図 5-5

- ※ 工場出荷時、ROCKEY2 の UID は「0」に設定されています。この状態では HID が正しく取得できず、HID が「0」と表示されます。正確な HID を取得するには、事前に UID を設定してください。
- ※ 初期化ツール以外にも、ROCKEY2 の API 関数（RY2\_Open）を使用して HID を取得できます。RY2\_Open (int mode, DWORD uid, DWORD\* hid)  
この関数を使用して ROCKEY2 を正しくオープンすると、HID が第 3 パラメータ（hid）に設定されます。詳細は、[\[RY2\\_Open：指定された ROCKEY2 をオープンする\]](#) を参照してください。

### 5.3 ユーザーデータゾーンへの書き込み

ROCKEY2 のユーザー ID（UID）が生成された後、5 つのデータゾーン（0～4）にデータを書き込むことができます。以下に、データゾーン 0 への書き込み手順を説明します。

- ※ **ユーザーデータゾーンは ROCKEY2 の API 方式のみ利用可能です。** エンベロープツール方式では利用できません。エンベロープツール方式を利用する場合、ユーザーデータゾーンの設定は不要です。

<データゾーン 0 への書き込み手順>

- 1、初期化ツールの「Write」（書き込み）ボタンをクリックします。
- 2、UID を入力します。

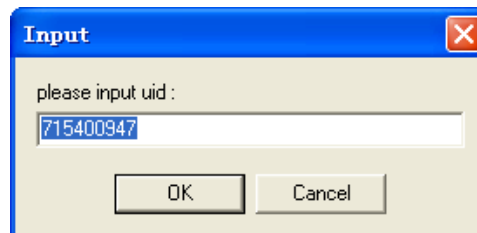


図 5-6 生成された ROCKEY2 の UID を入力

- 3、データゾーン番号を指定します。

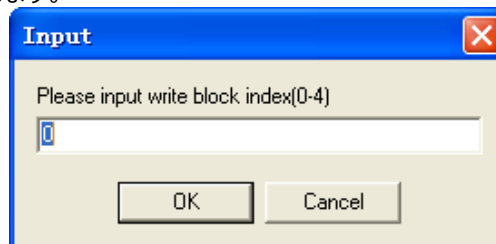


図 5-7 データゾーン番号（0～4）を入力

4、書き込みデータを入力します。

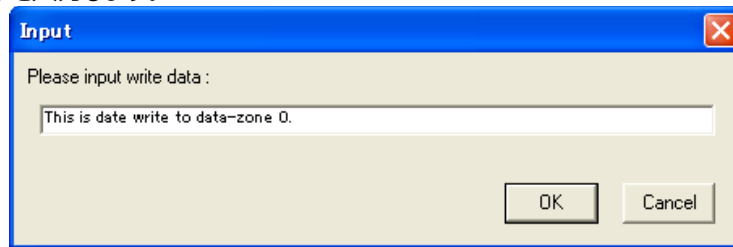


図 5-8 書き込みデータを入力

5、書き込み結果を確認します。

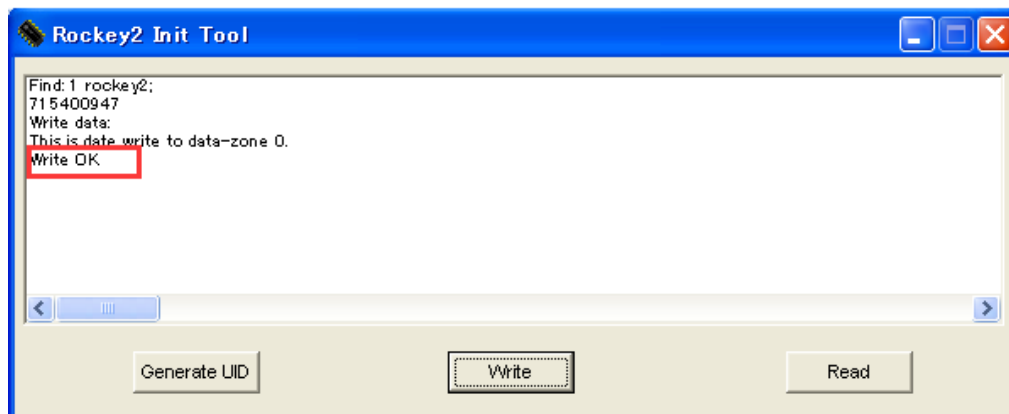


図 5-9 書き込み結果を確認

## 5.4 ユーザーデータゾーンの読み出し

ROCKEY2 のデータゾーン（0～4）に格納されたデータを読み出すことができます。以下に、データゾーン 0 の読み出し手順を説明します。

※ ユーザーデータゾーンは ROCKEY2 の API 方式のみ利用可能です。エンベロープツール方式では利用できません。

<データゾーン 0 の読み出し手順>

- 1、初期化ツールの“Read”（読み出し）ボタンをクリックします。
- 2、UID を入力します。

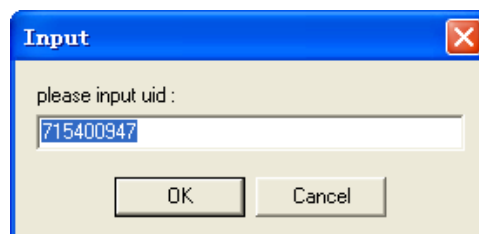


図 5-10 生成された ROCKEY2 の UID を入力

- 3、データゾーン番号を指定します。

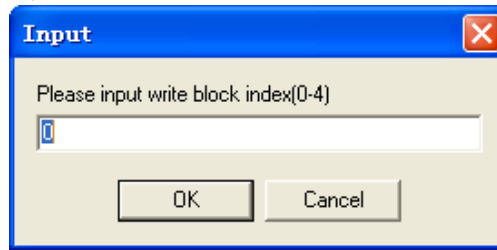


図 5-11 データゾーン番号を入力

- 4、データゾーンに格納されたデータが表示されます。

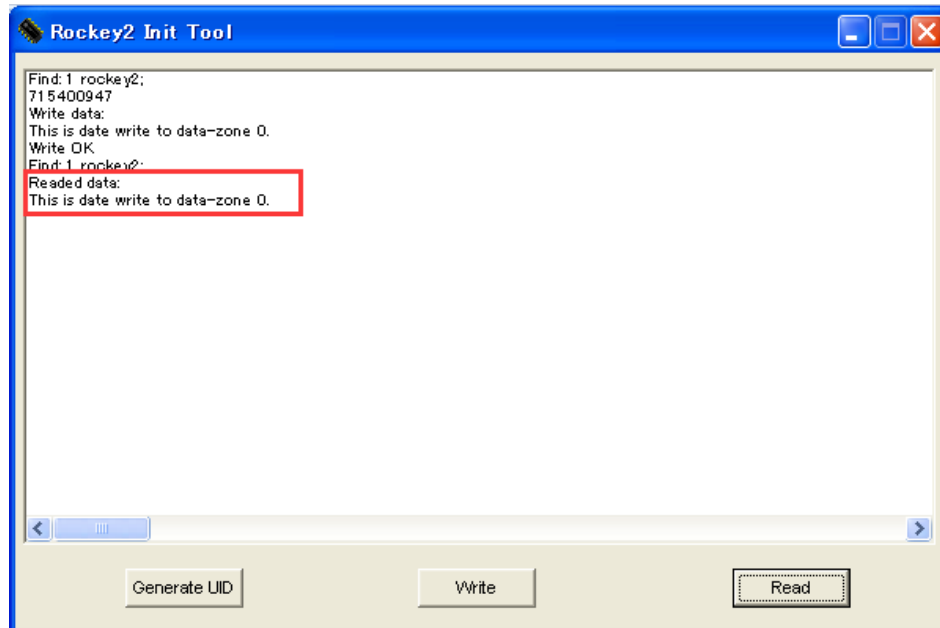


図 5-12 ROCKEY2 読み出し結果を確認

以上は、ROCKEY2 初期化ツールの利用方法についての説明です。

初期化ツールの各機能は API 方式で実現できます。今後の運用や効率化のために、API を活用して専用の社内初期化ツールを作成することをお勧めします。

## 第6章 ROCKEY2 エンベロープツール

ROCKEY2 のエンベロープツールは、ソフトウェアやデータを保護するための最も簡単で便利な方法です。プロテクトしたいソフトウェアをエンベロープツールに追加して、簡単な設定を実施するだけで、ソフトウェアのプロテクトが実現できます。

ソースコードがないソフトウェアをプロテクトしたい場合や、ROCKEY2 API の組み込み作業に時間をかけられない開発者にとって、エンベロープツールは有効な方法です。

エンベロープツールは、既存の 32 ビットと 64 ビットの Portable Executable (PE) ファイルおよび Microsoft .NET プログラム (2.x~4.x の Framework) の (\*.exe/\*.dll/\*.ocx) に対してプロテクトが可能です。

エンベロープツールの起動画面は下図 6-1 の通りです。

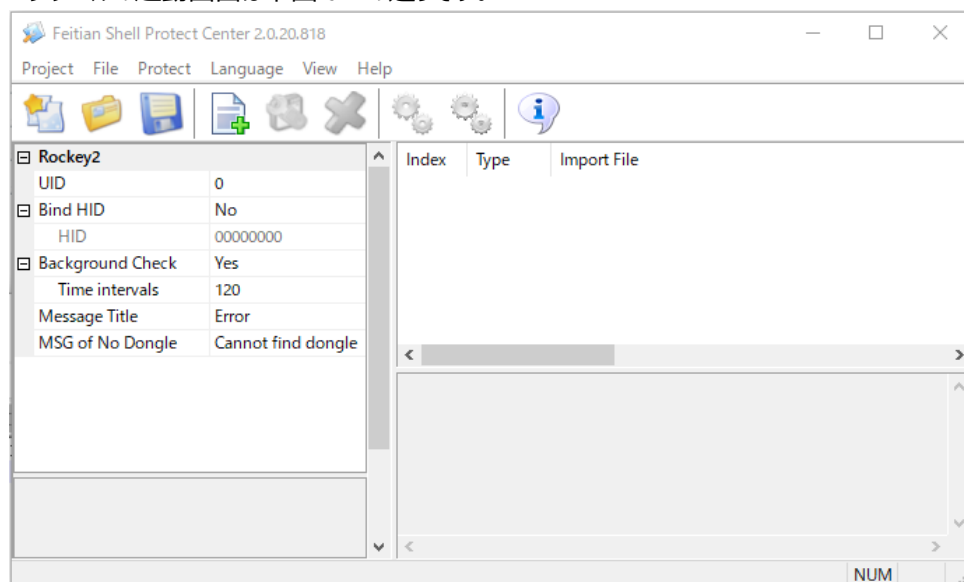


図 6-1 エンベロープツール画面

### 6.1 エンベロープツールの利用方法

#### 6.1.1 事前準備

- 1、UID が設定された ROCKEY2  
エンベロープツールを使用する前に、ROCKEY2 の UID を生成し、生成された UID を記録する必要があります。  
※詳細は『5.1 ユーザーID (UID) の生成』を参照してください。
- 2、エンベロープツールのフォルダをローカルにコピー  
SDK の CD を利用する場合、エンベロープツールのフォルダ (ROCKEY2 SDK Vx.xx¥Windows¥Utilities¥Envelope) をローカル環境にコピーしてから使用してください。

### 3、保護したいソフトウェア

エンベロープツールによって保護されたファイルは、元のソフトウェアと同じフォルダに生成されます。そのため、保護したいソフトウェア（EXE や DLL）を、読み書き可能なフォルダに移動してください。

## 6.1.2 エンベロープ方法

- ① ROCKEY2 を挿した状態でエンベロープツール（ROCKEY2 SDK Vx.xx\Windows\Utilities\Envelope\Envelope.exe）を実行します。
- ② エンベロープツールの左上にソフトウェアが利用できる ROCKEY2 の条件及びエラーに関連する項目を設定します。

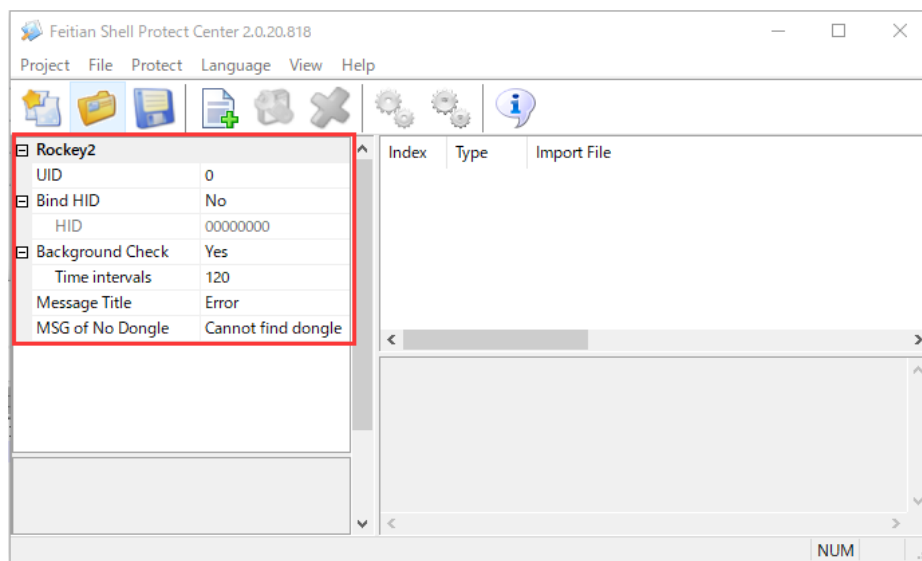


図 6-2 ROCKEY2 設定項目

**UID**：ソフトウェアに使用する ROCKEY2 の UID を入力します。

※ ROCKEY2 の UID が負の値の場合は、マイナス記号「-」も含めて入力してください。

**Bind HID**：ソフトウェアに利用する ROCKEY2 の HID を設定します。

- ・ 「Yes」を選択すると、特定な HID の ROCKEY2 のみソフトウェアを実行できます。指定する HID を入力してください。

※ ROCKEY2 の HID の取得方法については、[5.2 ハードウェア ID (HID) の取得] を参照してください。取得された HID は"0x"で始まる 16 進数ですが、HID 入力時に"0x"を含める必要はありません。

例：取得された HID：0xF255893A ⇒ Bind HID の入力欄：F255893A

※ 各 ROCKEY2 の HID は異なるため、ソフトウェアと ROCKEY2 の間に 1 対 1 の関係を構築できます。ただし、ROCKEY2 を紛失すると、そのソフトウェアは使用できなくなります。その場合、新しい ROCKEY2 を使用してソフトウェアを再度プロテクトし、ROCKEY2 とそれに対応するソフトウェアを購入者に再提供する必要があります。

- ・ 「No」を選択すると、ROCKEY2 の HID に関係なくソフトウェアを実行できます。  
 ※ この場合、ソフトウェアと ROCKEY2 の間には 1 対 N の関係が成立し、同じ UID を持つ ROCKEY2 であれば、どれでもソフトウェアを使用できます。ROCKEY2 を紛失した場合でも、購入者には同じ UID の ROCKEY2 を再提供するだけで済み、ソフトウェアを再度プロテクトする必要はありません。


**Background Check**：プロテクトされたソフトウェアが実行中に、定期的に ROCKEY2 の接続を確認するかどうかを設定します。

- ・ 「Yes」を選択すると、ソフトウェアが実行中でも一定の時間間隔で ROCKEY2 の接続状態を確認します。その際、**Time intervals** に時間間隔（単位：秒）を入力してください。実行後に ROCKEY2 を抜いた場合、エラーが発生します。
- ・ 「No」を選択すると、ソフトウェアが起動する際に ROCKEY2 を確認し、その後は接続状態をチェックしなくなります。これにより、プロテクトされたソフトウェアは実行時に ROCKEY2 を確認し、実行後に ROCKEY2 を抜いてもソフトウェアは引き続き実行されます。

**Message Title**：ソフトウェアに利用する dongle が見つからない場合、表示するエラーメッセージのウィンドウ・タイトルを設定します。日本語を設定することができます。

**MSG of No Dongle**：ソフトウェアに利用する dongle が見つからない場合、表示するエラーメッセージの内容を設定します。日本語を設定することができます。

- ③ 保護したいソフトウェアを、エンベロープツール画面の右上にあるファイル欄にドラッグするか、メニューの

「File」→「Add File」を選択するか、またはツールバーの  アイコンをクリックすることで追加できます。

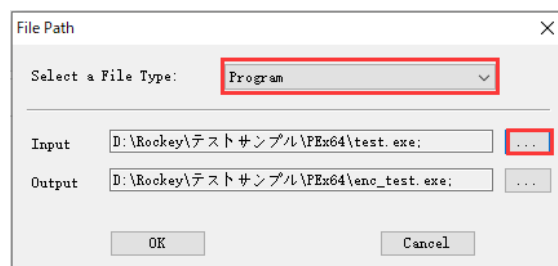


図 6-3 保護するソフトウェアの追加画面

**Select a File Type**：保護したい内容を選択します。プログラムまたはデータを選択可能です。

**Input**：「...」ボタンをクリックして、エンベロープ対象のファイル（例：test.exe）を選択します。

**Output**：エンベロープ後、出力されたプログラム名(例：enc\_test.exe)が自動的に表示されます。

「OK」ボタンをクリックすると、選択されたプログラムファイルが右側に追加されます。

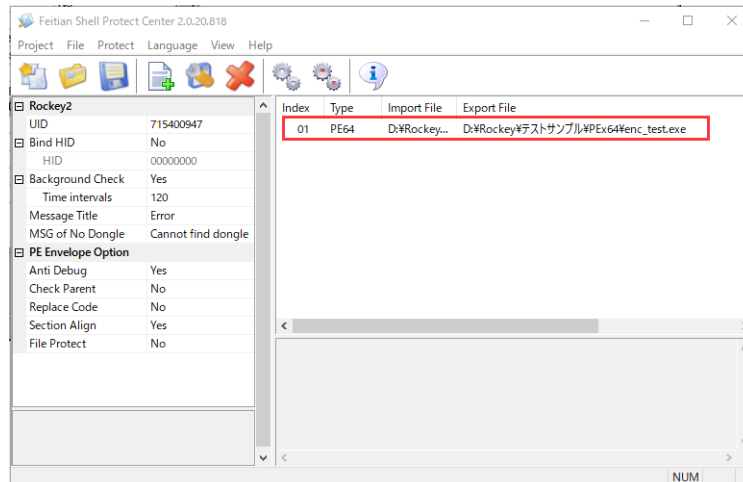


図 6-4 追加されたソフトウェア

Type (種類)	Type の意味
PE32	32bit の PE ソフトウェア
PE64	64bit の PE ソフトウェア
.Net32	32bit の .NET ソフトウェア
.Net64	64bit の .NET ソフトウェア

- ※ 【PE ソフトウェア】 は Portable Executable(PE)ソフトウェアの略称です。
- ※ 【.NET ソフトウェア】 は Microsoft .NET Framework を使用して開発されたソフトウェアの略称です。

- ④ 画面右側のソフトウェアを選択して、エンベロープオプションが画面左下に表示されます。ソフトウェアの種類 (Type) によって表示されるオプションが異なります。オプションを設定してください。

### ■【PE Envelope Option】:

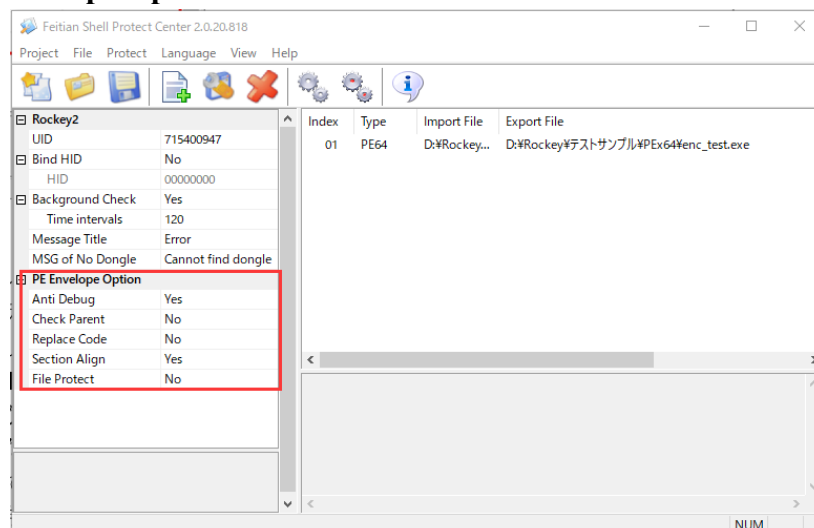


図 6-5 PE Envelope Option

**Anti Debug** : マルウェアのデバッグを阻止する機能です。

通常、攻撃者がデバッガを使用してソフトウェアのフロー、API 呼び出し、引数、データの流れなどを分析します。**Anti Debug** を利用すると、デバッガに Breakpoint を設定してもブレークが効かなくなり、ステップの実行を阻止することができます。

また、保護されたソフトウェアを実行する際に、他のソフトウェアがソフトウェアのメモリを利用しているかどうかを確認し、利用されている場合は保護されたソフトウェアを強制終了します。  
通常 **Anti Debug** を有効「Yes」に設定することをお勧めします。

**Check Parent**：他のプログラムやデバッガから呼び出せないように設定します。攻撃者は通常、逆コンパイラやデバッガツールを使用してソフトウェアを呼び出して内容を分析します。  
ソフトウェアが他のソフトウェアから呼び出されない場合は、**Check Parent** を有効「Yes」に設定してください。  
逆に、保護されたソフトウェアが他のソフトウェアから呼び出される可能性がある場合は、**Check Parent** を無効「No」に設定してください。

**Replace Code**：このオプションは、主に保護されたソフトウェアのソースコードをエンコーディングし、不要なコードや無意味なジャンプを挿入することで解読を難しくします。

このオプションを有効にすることをお勧めしますが、ソフトウェアによっては実行速度に影響がある場合があります。

**Section Align**：インストーラーを含むソフトウェアを保護します。  
例えば、InstallShield を使用しているソフトウェアなどが該当します。

インストーラーが含まれている場合は、この項目を有効に設定してください。含まれていない場合は、この項目を無効に設定してください。

**File Protect**：呼び出し先のファイルを保護する必要があるかどうかを設定します。  
エンベロープツールはプログラム及びデータに対して保護できます。例えば、エンベロープツールで PDF ファイルに対して保護すると、保護された PDF ファイルは通常の Adobe Reader では開けません。  
エンベロープツール（File Protect：無効）で保護された Adobe Reader でも開けません。  
エンベロープツール（File Protect：有効）で保護された Adobe Reader のみ開けます。

## ■【.NET Envelope Option】：

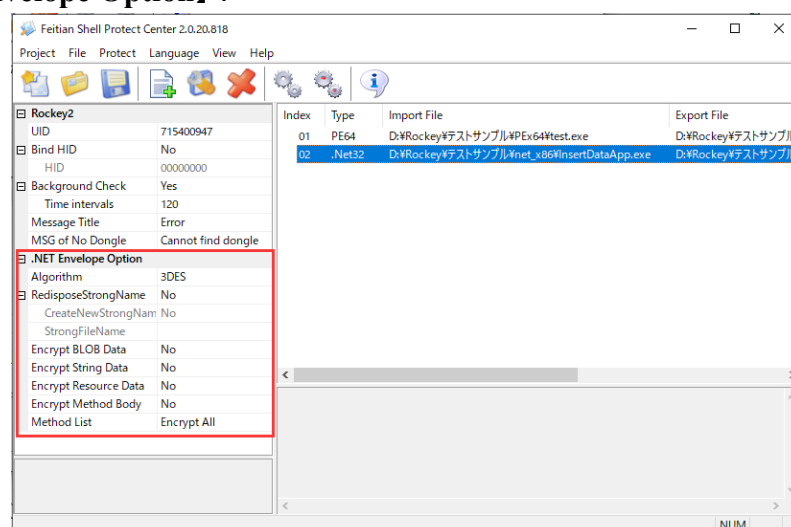


図 6-6 .NET Envelope Option



**Algorithm** : 暗号化用アルゴリズム (3DES/DES) を選択できます。

**RedisposeStrongName** : .NET ソフトウェアの厳密名 (StrongName) を設定するかどうかを設定します。

ソフトウェアに厳密名を設定したい場合は、この項目を有効にしてください。

新しい厳密名を設定したい場合は、**CreateNewStrongName** を「Yes」に設定します。

既存の厳密名を使用したい場合は、**CreateNewStrongName** を「No」に設定し、**StrongFileName** に厳密名を設定してください。

**Encrypt BLOB Data** : BLOB データを暗号化するかどうかを設定します。

BLOB データは.NET ソフトウェアが他のソフトウェアと通信するときに使用されるバイナリデータです。

「Encrypt BLOB Data」を有効にすると通信内容が暗号化され、無効にすると通信内容が平文のままになり、情報漏洩の可能性が高まります。ソフトウェアが他のプロセスと通信する場合は、「Encrypt BLOB Data」の利用をお勧めします。

**Encrypt String Data** : 文字列データを暗号化するかどうかを設定します。

このオプションを有効にすると、ファイルに記載された関数名やメッセージなどの文字列が暗号化され、解読が難しくなります。文字列に重要なデータが含まれている場合は、「Encrypt String Data」の利用をお勧めします。

**Encrypt Resource Data** : リソースデータを暗号化するかどうかを設定します。

このオプションを有効にすると、ソフトウェアに含まれたリソースデータ (画像、音、ビデオなど) が保護されます。ソフトウェアに大量なリソースデータが含まれる場合は、「Encrypt Resource Data」の利用をお勧めします。

**Encrypt Method Body** : **Method List** に選択された関数の内容を暗号化するかどうかを設定します。具体的にはソースコードの置換などの方法を使用して難読化されます。

**Method List** : 保護する関数の一覧が表示されます。デフォルトでは全ての関数が選択されています。

Microsoft .NET Framework を使用して開発されたソフトウェアに対しては、一部の機能が対応していない場合があります。エンベロープされたソフトウェアが実行できない場合は、保護する関数を減らして再度試してみてください。

- ⑤ メニュー「Protect」→「Protect Current Files」や「Protect All Files」を選択するか、またはツールバーの



(Protect Current File) ボタンをクリックして、エンベロープをかけます。エンベロープの結果は画面の

下に表示されます :

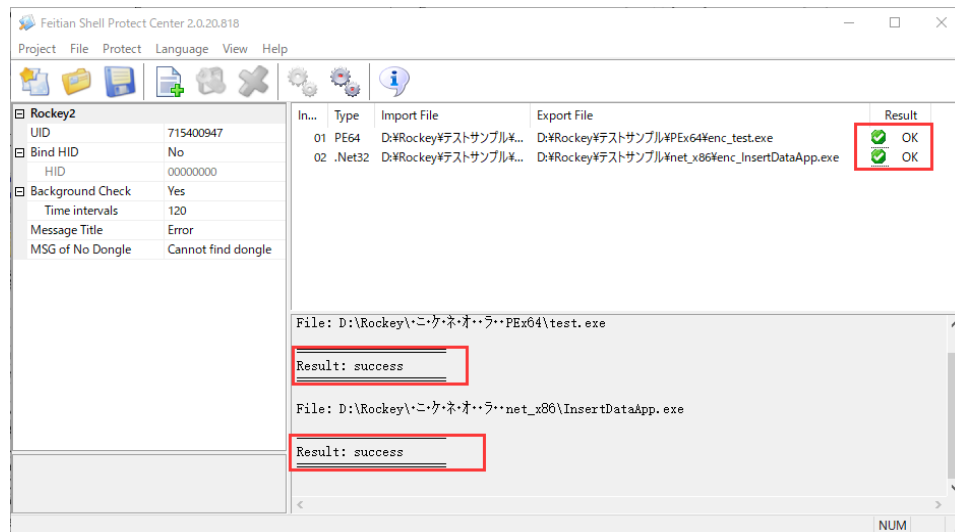


図 6-7 インベロープ実行後

正しくエンベロープされた場合、画面の右下に [Result : success] が表示され、エンベロープされた exe (enc\_XXX.exe) が自動的に生成されます。

- ※ [.Net ソフトウェア] が正しくプロテクトされた後、enc\_XXX.exe 以外にも runtime.dll が自動的に生成されます。enc\_XXX.exe と runtime.dll が同じフォルダに配置する必要があります。
- ※ [PE ソフトウェア] が正しくプロテクトされた後、enc\_XXX.exe のみ生成されます。

#### 【既知の問題】

入力ファイル (Import File) に日本語が含まれている場合、画面の右下の File 表示には文字化けが発生しますが、エンベロープには影響がないため、ソフトウェアの実行に支障がありません。

- ⑥ インベロープツールでは、設定内容を保存できます。  
設定内容を保存するには、メニューから [Project] → [Save Project] を選択します。これにより、設定内容がプロジェクトファイル (.prj) として保存されます。  
次回使用する際は、メニューから [Project] → [Open Project] を選択し、プロジェクトファイルを開いてください。

以上で、エンベロープ処理が完了しました。

- ※ インベロープツールに関するご質問は、[8.3 インベロープツールに関する FAQ] をご参照ください。

## 6.1.3 動作確認

プロテクトされる前のソフトウェア (XXX.exe) はドングルなしの状態ですそのまま実行できます。

プロテクトされた後のソフトウェア (enc\_XXX.exe) は ROCKEY2 を接続した状態で起動できます。  
ROCKEY2 を接続しない状態で起動することはできず、エンベロープツールに定義されたエラーメッセージ (MSG of No Dongle) が表示されます。

※ プロテクトされた後のソフトウェア（enc\_XXX.exe）のファイル名を変更できます。ただし、runtime.dll が生成された場合は、そのファイル名を変更しないでください。

#### 6.1.4 配布

プロテクトされたソフトウェアは、dongleとともにソフトウェアを購入した顧客に配布できます。

配布内容は以下のとおりです：

- ・ [PE ソフトウェア] の場合：プロテクトされた exe + ROCKEY2(ライセンス購入分)
- ・ [.Net ソフトウェア] の場合：[プロテクトされた exe + runtime.dll] + ROCKEY2(ライセンス購入分)

※ プロテクトされた EXE はコピーされても、dongleがないと実行できないため、不正使用を防ぐことができます。

※ ソフトウェアを追加購入する際は、dongleを設定してから配布すれば問題ありません。

## 6.2 エンベロープツールのコマンドライン

エンベロープツールは GUI 画面以外、エンベロープフォルダに含まれている RockeyCmdShell を利用して、コマンドライン方式でアプリケーションをエンベロープすることもできます。

### 6.2.1 RockeyCmdShell の種類

エンベロープツールのフォルダ（ROCKEY2 SDK Vx.xx¥Windows¥Utilities¥Envelope）には、

RockeyCmdShell.exe と RockeyCmdShell64.exe 二つ EXE があります。

エンベロープするアプリケーションの種類、ビット数によって、ご利用してください。

RockeyCmdShell の種類	保護するアプリケーション
RockeyCmdShell.exe	PE32、.NET32、.NET64 アプリケーション
RockeyCmdShell64.exe	PE64 アプリケーション

※ [PE32/64、.NET32/64] : 保護するアプリケーションをエンベロープツールに追加して、「TYPE」で表示されます。

### 6.2.2 RockeyCmdShell の起動及び基本のパラメーター

1、コマンドプロンプトを起動します。

2、Envelope フォルダ下の RockeyCmdShell64.exe を実行できます。

パラメータが設定されないため、パラメータのヘルプが表示されます。

```

C:\Users\¥ftsaf> D:\¥Rockey¥Envelope¥RockeyCmdShell64.exe
RockeyCmdShell On Windows [Version 1.0]
(C) Copyright 1998-2009 FeiTian Corp

Note: All the options part of the late, ignoring case, if the options are arguments
to use ':' partition, if parameters have spaces, use double quotes

-h      this help menu
-t      setting the rockey type
-pe     (default) the file is an pe file
-data   the file is an data file
-dotnet the file is an dotnet file

If you want to see the detailed parameters of each lock, specify the type of lock a
nd platform, Locks currently supported are:
Rockey2
Rockey3
Rockey4
Rockey4nd
Rockey4Smart
Rockey4Lm
Rockey6Smart
Et99
Et199
NetRockey4
NetRockey4nd
NetRockey6Smart
Eg: RockeyCmdShell -h -t:Rockey2 -pe, show the help of rockey2

C:\Users\¥ftsaf>
  
```

## 基本パラメータの説明

パラメータ	説明	
	英語	日本語
-h	this help menu	ヘルプを表示する。
-t	setting the rockey type	ROCKEY の種類を指定する、Rockey2 を指定してください。
-pe	(default) the file is an pe file	PE アプリケーション (デフォルト)
-data	the file is an data file	データファイル
-dotnet	the file is an dotnet file	Microsoft .NET Framework で開発された 32 ビット アプリケーション
-dotnet64		Microsoft .NET Framework で開発された 64 ビット アプリケーション

例：ROCKEY2 で PE アプリケーションを保護したい場合は、下記コマンドでヘルプを確認できます：

```
> D:\¥Rockey¥Envelope¥RockeyCmdShell.exe -h -t:Rockey2 -pe
```

実行結果：

```

コマンドプロンプト
C:\Users\¥ftsaf>D:\¥Rockey¥Envelope¥RockeyCmdShell.exe -h -t:Rockey2 -pe
RockeyCmdShell On Windows [Version 1.0]
(C) Copyright 1998-2010 Feitian Corp

Note: All the options part of the late, ignoring case, if the options are arguments to use ':' partition, if parameters have spaces, use double quotes
Pe file option:
-if the input file
-of the output file
-antidbg open anti-debugger
-chkpp open check parent process
-call2cc open instruction replacement
-protFile the protected program is used to open the protected data

Rockey option:
-UID Set User ID
-HID Set Hardware ID
-Nrkt Errors in the title no encryption locks
-Nrkm The error message is not encrypted locks
-BkChk Background timing detection

C:\Users\¥ftsaf>

```

※詳細のパラメータは下記で説明します。

## 6.2.3 ROCKEY2 製品に関するパラメータ

Rockey option	説明	
	英語	日本語
-UID	Set User ID	ユーザーID (16 進数)
-HID	Set Hardware ID	ハードウェア ID (16 進数)
-Nrkt	Errors in the title no encryption locks	ドングルがないときに、表示されたエラーメッセージのタイトル
-Nrkm	The error message is not encrypted locks	ドングルがないときに、表示されたエラーメッセージの内容
-BkChk	Background timing detection	バックグラウンドタイミングチェック機能を利用、単位 (秒、16 進数)

## 6.2.4 PE アプリケーションに関するパラメータ

PE File option	説明	
	英語	日本語
-if	the input file	入力ファイル（保護したいファイル）
-of	the output file	出力ファイル（保護されたファイル）
-AntiDbg	open anti-debugger	マルウェアのデバッグを阻止する機能を利用 ※エンベロープツールの PE Envelope Option : 「Anti Debug」
-chkpp	Open check parent process	親プロセスの確認機能の利用 ※エンベロープツールの PE Envelope Option : 「Check Parent」
-call2cc	Open instruction replacement	コードの切替機能の利用 ※エンベロープツールの PE Envelope Option : 「Replace Code」
-SctAlign		インストラクタを含むソフトウェアに対して保護を利用 ※エンベロープツールの PE Envelope Option : 「Section Align」
-protFile	The protected program is used to open the protected data	保護されたプログラムは保護されたデータを開くため使用されます ※エンベロープツールの PE Envelope Option : 「File Protect」

## 6.2.5 Microsoft .NET アプリケーションに関するパラメータ

Dotnet File option	説明	
	英語	日本語
-if	the input file	入力ファイル（保護したいファイル）
-of	the output file	出力ファイル（保護されたファイル）
-algorithm	set encryption algorithm,support 3DES and DES	暗号化アルゴリズム（3DES/DES）を選択 ※エンベロープツールの.NET Envelope Option : 「Algorithm」
-Rdsn	strong file name replacement	.NET アプリケーションの厳密名（StrongName）を設定 ※エンベロープツールの.NET Envelope Option : 「RedisposeStrongName」
-Cnsn	to create a new strong file name	新しい厳密名を作成 ※エンベロープツールの.NET Envelope Option : 「CreateNewStrongName」

-StrongName	set Strong file name	新しい厳密名を設定 ※エンベロープツールの.NET Evelope Option : 「StrongName」
-Blob	open BLOB Encryption	BLOB データを暗号化 ※エンベロープツールの.NET Evelope Option : 「Encrypt BLOB Data」
-String	open String Encryption	文字列データを暗号化 ※エンベロープツールの.NET Evelope Option : 「Encrypt String Data」
-Resource	open the resource file encryption	リソースデータを暗号化 ※エンベロープツールの.NET Evelope Option : 「Encrypt Resouce Data」
-Method	open String Encryption	Method List に選択された関数の内容に対して暗号化 ※エンベロープツールの.NET Evelope Option : 「Encrypt Method Body」
-AllToken	encrypt all the methods	全ての関数に対して暗号化 ※エンベロープツールの.NET Evelope Option : 「Method List」:Encrypt All
-Token	TOKEN value is set, you can set up several times, encryption, mul tiple TOKEN	指定された関数に対して暗号化することはできます。同時に複数関数に暗号化することもできます。例 : -token:functionA – token:functionB
- RuntimeName	set the Runtime Libaray Name	ランタイムライブラリファイル名を指定する

## 6.2.6 ROCKEY2 のコマンド実行例

例 : ROCKEY2 [UID : 715400947 (16 進数 : 2AA426F3) 、HID : 0x74A76F0D] を利用

### 6.2.6.1 ROCKEY2 で PE32 を保護する場合

```
D:¥Rockey¥Envelope¥RockeyCmdShell.exe -PE -if:"D:¥test.exe" -of:"D:¥enc_test.exe" -
AntiDbg -Call2CC -ChkPP -t:Rockey2 -UID:2AA426F3 -HID:74A76F0D -BkChk:78 -
Nrkt:"Error" -Nrkm:"Cannot find dongle"
```

### 6.2.6.2 ROCKEY2 で PE64 を保護する場合

```
D:¥Rockey¥Envelope¥RockeyCmdShell64.exe -PE -if:"D:¥test.exe" -of:"D:¥enc_test.exe" -
AntiDbg -Call2CC -ChkPP -t:Rockey2 -UID:2AA426F3 -HID:74A76F0D -BkChk:78 -
Nrkt:"Error" -Nrkm:"Cannot find dongle"
```

### 6.2.6.3 ROCKEY2 で.NET32 を保護する場合

```
D:¥Rockey¥Envelope¥RockeyCmdShell.exe -t:Rockey2 -UID:2AA426F3 -HID:74A76F0D -
BkChk:78 -Nrkt:"エラー" -Nrkm:"ドングルが見つかりません" -Dotnet -if:"D:¥test.exe" -
of:"D:¥enc_test.exe" -Algorithm:3DES -AllToken
```

### 6.2.6.4 ROCKEY2 で.NET64 を保護する場合

```
D:¥Rockey¥Envelope¥RockeyCmdShell.exe -t:Rockey2 -UID:2AA426F3 -HID:74A76F0D -
BkChk:78 -Nrkt:"エラー" -Nrkm:"ドングルが見つかりません" -Dotnet64 -if:"D:¥test.exe" -
of:"D:¥enc_test.exe" -Algorithm:3DES -AllToken
```

エンベロープツールのコマンドライン方式を使用して bat ファイルを作成すると、ソフトウェアのバージョンアップ時などに、以前と同様にソフトウェアを保護することができます。ぜひご活用ください。

## 6.3 注意事項

- 1、エンベロープツールの GUI 方式およびコマンドライン方式のいずれを利用する場合も、ROCKEY2 ドングルの接続が必要です。手元にドングルがない場合、エンベロープツールは使用できませんのでご注意ください。
- 2、一部のアプリケーションは特別な方法で作成・コンパイルされているため、エンベロープツールに対応できない可能性があります。評価の際は、実際に使用するアプリケーションで動作確認を行うことをお勧めします。
- 3、エンベロープツールの GUI 画面では、複数のソフトウェアを追加し、一括でエンベロープ処理を行うことが可能です。  
一方、コマンドライン方式を利用する場合は、各ソフトウェアごとに 1 行ずつコマンドを設定する必要があります。
- 4、保護対象のソフトウェアを複数回エンベロープすることは可能ですが、すでにエンベロープされたファイルを再度エンベロープすることはできませんので、ご注意ください。
- 5、エンベロープ方式と API 方式の両方を使用してソフトウェアを保護する場合は、まずソースコードに ROCKEY2 の API を組み込み（『第 7 章 ROCKEY2 の関数（API）』を参照）、EXE を生成してください。その後、生成された EXE をエンベロープツールで保護してください。
- 6、エンベロープツールに関するご質問は、[9.3 エンベロープツールに関する FAQ] をご参照ください。



## 第7章 ROCKEY2 の関数（API）

ROCKEY2 ソフトウェアプログラミングインターフェース（API）は、基本的かつ柔軟な暗号化手段を提供します。ROCKEY2 の API を利用することで、 dongle の暗号化機能を最大限に活用できます。API セットは簡素化されており、ROCKEY2 によるプログラミング効果を最大化するよう設計されています。

ROCKEY2 API による暗号化では、セキュリティレベルを向上させるために、ソフトウェアから ROCKEY2 の API を呼び出します。ソフトウェアの任意の部分で dongle の存在を確認し、その状態に基づいた処理を実行することが可能です。また、UDZ（User Data Zone）に保存されたデータをチェックすることもできます。

API の呼び出し方法について、C 言語によるインターフェース例を使用して説明しますが、他のプログラミング言語からもほぼ同じ方法で API を呼び出すことが可能です。

### 7.1 Rockey2 API インターフェース

#### 7.1.1 RY2\_Find：接続された ROCKEY2 を検索する

EXTERN C int WINAPI RY2\_Find();

戻り値	< 0	エラー
	= 0	ROCKEY2 が接続されてません
	> 0	コンピュータに接続された ROCKEY2 の個数

#### 7.1.2 RY2\_Open：指定された ROCKEY2 をオープンする

EXTERN C int WINAPI RY2\_Open(int mode, DWORD uid, DWORD\* hid);

入力パラメータ	mode	<p>ROCKEY2 のオープンモードを示す</p> <p>mode &gt; 0：UID 方式で ROCKEY2 をオープンします。 mode の設定値は dongle の番号です。 例えば：uid=12345678、mode=2 は、uid が 12345678 で、2 番目の dongle をオープンすることを意味します。 ※注：ここでの「2 番目」とは、ソフトウェアが稼働中に 2 番目に接続された ROCKEY2 を指します。複数の ROCKEY2 を使用するソフトウェアで利用されます。</p> <p>mode = -1：HID 方式で ROCKEY2 をオープンします。 条件：*hid は 0 以外である必要があります。</p> <p>mode = 0：最初に見つかった dongle をオープンします。 ROCKEY2 の UID と HID は使用しません ※注：mode = 0 を使用すると、UID や HID に関係なく最初に見つかった dongle を開き、データを読み書きできます。</p> <p>また、以下の事前に決められたモードのパラメータを利用することも可能です。 AUTO_MODE=0 HID_MODE=-1</p>
---------	------	---

	uid(UserID)	UID 方式でオープンする際は、このパラメータに UID を指定します。具体的なユーザーID の値は、RY2_GenUID を使用して取得できます
	hid	ROCKEY2 のハードウェア ID を指定や表示するパラメーターです。この値は入/出力データとなります。 HID 方式で dongle をオープンする際は、このパラメータに HID を指定します。 どちらの方式で ROCKEY2 を開く場合でも、成功すればこのパラメータを通じて dongle のハードウェア ID を取得できます。
戻り値	>=0	成功。パラメータには dongle のハンドルアドレスが書き込まれます。
	< 0	エラー

### 7.1.3 RY2\_Close : 指定された dongle を閉じる

EXTERN C void WINAPI RY2\_Close(int handle);

入力パラメータ	指定のハンドルで ROCKEY2 を閉じます。 handle : dongle のハンドル。RY2_Open の戻り値のハンドルと同じものを指定します。
戻り値	無し。

### 7.1.4 RY2\_GenUID : ユーザーID を生成する

EXTERN C int WINAPI RY2\_GenUID(int handle, DWORD\* uid, char\* seed, int isProtect)

入力パラメータ	handle	dongle のハンドル。RY2_Open の戻り値のハンドルと同じものを指定します。
	uid	出力パラメータ。生成されたユーザーID はこのパラメータに格納されます。
	seed	ユーザーID を生成するための Seed コード。最大 64 バイトの文字列です。
	isProtect	<保留> ※「ROCKEY2_DISABLE_WRITE_PROTECT」のみ指定可能です。
戻り値	エラーコード。詳細は、7.1.7 エラーコード で説明されています。	

### 7.1.5 RY2\_Read : dongle のデータを読み出し

EXTERN C int WINAPI RY2\_Read(int handle, int block\_index, char\* buffer512);

入力パラメータ	handle	dongle のハンドル。RY2_Open の戻り値のハンドルと同じものを指定します。
	block_index	操作したいユーザーデータゾーンを指定します。0~4 の中からどれかを指定できます。
	buffer512	ユーザーデータゾーンの読み出しバッファ。各ユーザーデータゾーンのサイズは 512 バイトであり、このバッファも 512 バイトです。
戻り値	エラーコード。詳細は、7.1.7 エラーコード で説明されています。	

### 7.1.6 RY2\_Write : dongle にデータを書き込み

EXTERN C int WINAPI RY2\_Write(int handle, int block\_index, char\* buffer512);

入力パラメータ	handle	dongle のハンドル。RY2_Open の戻り値のハンドルと同じものを指定します。
	block_index	操作したいユーザーデータゾーンを指定します。0~4 の中からどれかを指定できます。
	buffer512	ユーザーデータゾーンの読み出しバッファ。各ユーザーデータゾーンのサイズは 512 バイトであり、このバッファも 512 バイトです。
戻り値	エラーコード。詳細は、7.1.7 エラーコード で説明されています。	

## 7.1.7 エラーコード

エラーコード	値	説明
RY2ERR_SUCCESS	0	成功
RY2ERR_NO_SUCH_DEVICE	0xA0100001	指定された ROCKEY2 が見つからない (パラメータエラー)
RY2ERR_NOT_OPENED_DEVICE	0xA0100002	この関数を実行する前に、RY2_Open で ROCKEY2 を開く必要がある (操作エラー)
RY2ERR_WRONG_UID	0xA0100003	UID 間違い (パラメータエラー)
RY2ERR_WRONG_INDEX	0xA0100004	読み出し/書き込みユーザデータゾーンが間違い (パラメータエラー)
RY2ERR_TOO_LONG_SEED	0xA0100005	GenUID 機能をコールする際、指定された Seed コードの長さが 64 バイトを超えている (パラメータエラー)
RY2ERR_WRITE_PROTECT	0xA0100006	書き込み保護ハードウェアに対してデータを修正しようとしている (操作エラー)
RY2ERR_OPEN_DEVICE	0xA0100007	ROCKEY2 オープンエラー (Windows エラー)
RY2ERR_READ_REPORT	0xA0100008	読み出しエラー (Windows エラー)
RY2ERR_WRITE_REPORT	0xA0100009	書き込みエラー (Windows エラー)
RY2ERR_SETUP_DI_GET_DEVICE_INTERFACE_DETAIL	0xA010000A	内部エラー (Windows エラー)
RY2ERR_GET_ATTRIBUTES	0xA010000B	内部エラー (Windows エラー)
RY2ERR_GET_PREPARSED_DATA	0xA010000C	内部エラー (Windows エラー)
RY2ERR_GETCAPS	0xA010000D	内部エラー (Windows エラー)
RY2ERR_FREE_PREPARSED_DATA	0xA010000E	内部エラー (Windows エラー)
RY2ERR_FLUSH_QUEUE	0xA010000F	内部エラー (Windows エラー)
RY2ERR_SETUP_DI_CLASS_DEVS	0xA0100010	内部エラー (Windows エラー)
RY2ERR_GET_SERIAL	0xA0100011	内部エラー (Windows エラー)
RY2ERR_GET_PRODUCT_STRING	0xA0100012	内部エラー (Windows エラー)
RY2ERR_TOO_LONG_DEVICE_DETAIL	0xA0100013	内部エラー
RY2ERR_WRONG_REPORT_LENGTH	0xA0100020	未知のデバイス (ハードウェアエラー)
RY2ERR_VERIFY	0xA0100021	操作検証エラー (ハードウェアエラー)
RY2ERR_UNKNOWN_ERROR	0xA010FFFF	予期せぬエラー (ハードウェアエラー)

## 7.2 基本的なサンプル

以下のサンプルは、VC 6.0 で作成されたコントロールパネルプログラムです。オリジナルプログラムを暗号化する手順として、ROCKEY2 の API を組み込む方法をステップバイステップで説明します。特別な C 言語の技術は使用していないため、他の開発言語でソフトウェアを開発する場合でも、このセクションを参考にするだけで API の組み込み方法を理解できます。

## 7.2.1 Step0--オリジナルプログラム

以下は、ROCKEY2 の API がまだ追加されていない状態のオリジナルプログラムです。

```
#include <windows.h>
#include <stdio.h>

void main()
{
    printf("Hello ROCKEY2!\n");
}
```

## 7.2.2 Step1-- dongle の初期化（ユーザーID の生成）

任意の Seed コードを指定して ROCKEY2 を初期化し、ユーザーID（UID）を生成します。

※注意：Seed コードの長さは 64 バイト以下で指定してください。

```
#include <windows.h>
#include <stdio.h>
#include "rockey2.h"

void main()
{
    int retcode, handle;
    DWORD uid, hid;
    char buffer[512];

    // dongle を検索する
    retcode = RY2_Find();
    if (retcode < 0)
    {
        printf("ERROR: %08x\n", retcode);
        return;
    }
    if (retcode == 0)
    {
        printf("ERROR: Not found any rockey2\n");
        return;
    }
    else
        printf("Found rockey2\n");

    // dongle の UID と HID が関係なく、初めて見つかった dongle を利用する
    retcode = RY2_Open(AUTO_MODE, 0, &hid);
    if (retcode < 0)
    {
        printf("ERROR: %08x\n", retcode);
        return;
    }
    handle = retcode;

    // Seed を指定し、ユーザーID を生成する
    printf("\n");
    printf("Please input seed : ");
    scanf("%s", buffer);
```

```

        retcode = RY2_GenUID(handle, &uid, buffer,
                               ROCKEY2_DISABLE_WRITE_PROTECT);
        if (retcode < 0)
        {
            printf("ERROR: %08x¥n", retcode);
            return;
        }
        printf("¥nUID is %d¥n", uid);

        RY2_Close(handle);

        printf("¥n");
        printf("You must plug out Rockey2 and re-plug to enable new uid.¥n");
    }

```

### 7.2.3 Step2-- 指定された UID で dongle を開く

サンプル Step1 で生成した UID を使用して ROCKEY2 をオープンし、戻り値でオープンの成否を判断します。

```

#include <windows.h>
#include <stdio.h>
#include "rockey2.h"

void main()
{
    int retcode, handle;
    DWORD uid, hid;

    // dongle を検索する
    retcode = RY2_Find();
    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    if (retcode == 0){
        printf("ERROR: Not found any rockey2¥n");
        return;
    }

    // Step1 に Seed が 12345 の場合、生成された UID は 715400947 となります。
    // UID モードで dongle を開きます。
    uid=715400947;
    retcode = RY2_Open(1, uid, &hid);

    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    } else
        printf("Hello ROCKEY2!¥n");

    handle = retcode;
    RY2_Close(handle);
}

```

## 7.2.4 Step3-- ドングルのユーザデータゾーンにデータを書き込む

UID 方式でdongleを開き、データゾーンにデータを書き込みます。  
以下のサンプルでは、データゾーン 0 に「Hello ROCKEY2!」を書き込みます。

```
#include <windows.h>
#include <stdio.h>
#include "rockey2.h"

void main()
{
    int retcode, handle, block_index;
    DWORD uid, hid;
    char buffer[512];

    // ドングルを検索する
    retcode = RY2_Find();
    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    if (retcode == 0){
        printf("ERROR: Not found any rockey2¥n");
        return;
    }

    //Step1 に Seed が 12345 の場合、生成された UID は 715400947 となります。
    //UID モードでdongleを開きます。
    uid=715400947;
    retcode = RY2_Open(1, uid, &hid);

    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }

    //ユーザーデータゾーン 0 にデータを書き込む
    handle = retcode;
    block_index=0;
    strcpy((char*)buffer,"Hello ROCKEY2 !");
    retcode = RY2_Write(handle, block_index, buffer);
    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }

    printf("Write OK¥n");
    RY2_Close(handle);
}
```

## 7.2.5 Step4-- ドングルのユーザーデータゾーンの内容を読み出す

UID モードで ROCKEY2 を開くと、データゾーンの内容を読み取ることができます。  
以下のサンプルでは、データゾーン 0 の内容を読み取ります。

```
#include <windows.h>
#include <stdio.h>
#include "rockey2.h"

void main()
{
    int retcode, handle, block_index;
    DWORD uid, hid;
    char buffer[512];

    // ドングルを検索する
    retcode = RY2_Find();
    if (retcode < 0)
    {
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    if (retcode == 0)
    {
        printf("ERROR: Not found any rockey2¥n");
        return;
    }

    //Step1 に Seed が 12345 の場合、生成された UID は 715400947 となります。
    //UID モードでドングルを開きます。
    uid=715400947;
    retcode = RY2_Open(1, uid, &hid);
    if (retcode < 0)
    {
        printf("ERROR: %08x¥n", retcode);
        return;
    }

    //ユーザーデータゾーン 0 の内容を読み出す
    handle = retcode;
    block_index=0;
    retcode = RY2_Read(handle, block_index, buffer);
    if (retcode < 0)
    {
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    printf("Readed data: %s¥n", buffer);

    RY2_Close(handle);
}
```

## 7.3 総合応用サンプル

### 7.3.1 Step5-- 一つの ROCKEY2 の複数データゾーンへの書き込み/読み出し

以下のサンプルは、UID モードで dongle を開き、5 つのデータゾーンに順番にデータを書き込み、読み取る手順を示しています。

実際の開発では、要件に合わせて適宜調整してください。

```
#include <windows.h>
#include <stdio.h>
#include <conio.h>
#include "rockey2.h"

void main()
{
    int retcode, handle, block_index;
    DWORD uid, hid;
    char buffer[512];

    //dongleを検索
    retcode = RY2_Find();
    if (retcode < 0)
    {
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    if (retcode == 0)
    {
        printf("ERROR: Not found any rockey2¥n");
        return;
    }

    //Step1 に Seed が 12345 の場合、生成された UID は 715400947 となります。
    //UID モードで dongle を開きます。
    uid=715400947;
    retcode = RY2_Open(1, uid, &hid);
    if (retcode < 0)
    {
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    handle = retcode;

    //ユーザーデータゾーン 0 に"Module A"を書き込む
    block_index=0;
    strcpy((char*)buffer,"Module A");
    retcode = RY2_Write(handle, block_index, buffer);
    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }
}
```



```

//ユーザーデータゾーン 1 に"Module B"を書き込む
block_index=1;
strcpy((char*)buffer,"Module B");
retcode = RY2_Write(handle, block_index, buffer);
if (retcode < 0){
    printf("ERROR: %08x¥n", retcode);
    return;
}

//ユーザーデータゾーン 2 に"Module C"を書き込む
block_index=2;
strcpy((char*)buffer,"Module C");
retcode = RY2_Write(handle, block_index, buffer);
if (retcode < 0){
    printf("ERROR: %08x¥n", retcode);
    return;
}

//ユーザーデータゾーン 3 に"Module D"を書き込む
block_index=3;
strcpy((char*)buffer,"Module D");
retcode = RY2_Write(handle, block_index, buffer);
if (retcode < 0){
    printf("ERROR: %08x¥n", retcode);
    return;
}

//ユーザーデータゾーン 4 に"Module E"を書き込む
block_index=4;
strcpy((char*)buffer,"Module E");
retcode = RY2_Write(handle, block_index, buffer);
if (retcode < 0){
    printf("ERROR: %08x¥n", retcode);
    return;
}

//ユーザーデータゾーン 0~4 の内容を読み出す
for(block_index=0;block_index<5;block_index++)
{
    retcode = RY2_Read(handle, block_index, buffer);
    if (retcode < 0){
        printf("ERROR: %08x¥n", retcode);
        return;
    }
    printf("Readed data: %s¥n", buffer);
}

RY2_Close(handle);
}

```

## 第8章 実用前に検討すべき事項

ROCKEY2 を実用する前に、以下の事項を検討頂ければ幸いです。

### 8.1 ソフトウェアと ROCKEY2 の紐付けを計画

実際に運用する前に、事前に計画を立てる必要があります。保護するソフトウェアの種類や、今後のソフトウェアのバージョンアップなどの拡張性を考慮し、ソフトウェアと ROCKEY の関係を適切に設定してください。

例えば、ソフトウェア A を制御し、将来的にどのバージョンでも ROCKEY2 があれば使用できるようにしたい場合、ソフトウェア A と ROCKEY2 の SeedCode (例：softA0001)に関連付けることができます。この設定を行えば、今後どのバージョンのソフトウェア A でも SeedCode (softA0001)の ROCKEY2 で保護でき、ソフトウェアの購入者には SeedCode (softA0001)の ROCKEY2 を提供すれば問題ありません。

一方で、ソフトウェア A の異なるバージョンごとに個別の料金が発生する場合（例えば、V1.0 を購入しても、新バージョン V2.0 は別途料金が必要な場合）、バージョンごとに異なる SeedCode を設定することが可能です。

例：ソフトウェア A V1.0 ↔ ROCKEY2 ( SeedCode: softA0001-v1.0 )  
ソフトウェア A V2.0 ↔ ROCKEY2 ( SeedCode: softA0001-v2.0 )

SeedCode が同じで、UserDataZone で区別することも可能です。

例：ソフトウェア A V1.0 ↔ ROCKEY2 ( SeedCode: softA0001、UserDataZone0=Ver1.0 )  
ソフトウェア A V2.0 ↔ ROCKEY2 ( SeedCode: softA0001、UserDataZone0=Ver2.0 )

このように設定することで、各バージョンのライセンス管理を柔軟に行うことができます。

各ソフトウェアと ROCKEY2 の SeedCode の対応関係は、重要な情報のため慎重に管理・保管してください。

### 8.2 自社社内専用初期化ツールの作成

ROCKEY2 の SDK に含まれている初期化ツールを使用すると、ROCKEY2 の UID 生成やユーザーデータゾーンの書き込みが可能です。ただし、初期化ツールには SeedCode の保存機能がないため、毎回 SeedCode の入力や UID の検証が必要となり、手間がかかります。

開発能力のある企業には、ROCKEY2 の API を活用し、自社専用の ROCKEY2 初期化ツールを作成することをお勧めします。SeedCode をソースコードに直接記述したり、簡易的なデータベースに保存したりすることで、ソフトウェアに対応する ROCKEY2 を効率的に初期化できます。

具体的なサンプルは、ROCKEY2 SDK Vx.xx¥Windows¥Samples フォルダ内にありますので、ご参照ください。

### 8.3 プロテクト方式の選択及び注意点

ROCKEY2 のエンベロープツール方式は、その利便性の高さから多くのユーザーに利用されています。しかし、エンベロープツール方式はすべてのソフトウェアに対応できるわけではないため、対応が難しい場合は API 方式を検討することを推奨します。

また、エンベロープツールを使用する際は、ROCKEY2 ドングルが接続されている必要があります。そのため、最低でも 1 本のdongleを社内に保管することを推奨します。

もしdongleが手元にない場合、追加で 1 本購入することを検討してください。

## 第9章 FAQ

---

ここでは、まず ROCKEY2 を使用しているときに発生する可能性のある一般的な問題を解決する際に役立つと考えられるチェック項目をいくつか列挙します。また、FAQ ではよくある質問に関してご案内します。

### 9.1 トラブルシューティングの確認事項

- 最新版の ROCKEY2 SDK をご使用になれていますか？  
⇒弊社のウェブサイト：<https://ftsafe.co.jp/support/download/rockey2/> より最新の SDK をダウンロードできます。
- 問題が発生した際に、そのdongleを他のコンピュータでも試してみましたか？  
⇒コンピュータの環境（OS のバージョン等）に起因している可能性があります。

### 9.2 基本の運用に関する FAQ

#### 9.2.1 プロテクトをかけるプログラム数に制限はあるでしょうか？

プロテクトをかけるプログラム数に制限はありません。

#### 9.2.2 プロテクトされたソフトウェアは複数のdongleに対応できますか？

はい、対応可能です。

複数の ROCKEY2 に同じ SeedCode を設定すると、同じ UID が生成されます。この UID を利用してエンベロップ方式や API 方式でソフトウェアを保護すれば（HID を使用しない限り）、どの ROCKEY2 を使用してもソフトウェアを実行できます。

※ 各 ROCKEY2 の HID（ハードウェア ID）は異なります。そのため、HID を使用すると、保護されたソフトウェアは特定の ROCKEY2 でのみ実行可能になります。一方で、HID を利用せず UID のみを使用すれば、複数のdongleで同じソフトウェアを使用できます。

※ 初期化作業を簡略化するために、API 方式を利用し、自社専用の初期化ツールを開発することを推奨します。

#### 9.2.3 プロテクトをかけた EXE や DLL 自体の不正コピーは制限できますか？

プロテクトされたEXEやDLL自体のコピーを物理的に制限することはできません。

しかし、それらのファイルをコピーされても、ROCKEY2dongleを接続しない限り実行できません。そのため、不正コピーによるソフトウェアの不正利用をdongleによって抑制することが可能です。

## 9.3 インベロープツールに関する FAQ

### 9.3.1 [Result: failed, error code:100, error message Find Dongle Failed] エラー

【問題】：

ROCKEY2 のエンベロープツールでソフトウェア（EXE ファイル）をエンベロープすると、[Result: failed, error code:100, error message Find Dongle Failed] エラーが発生します。

【原因】：

エンベロープ作業時に、ROCKEY2 が PC に接続されていない可能性があります。

【解決策】：

ROCKEY2 を PC に接続してから、エンベロープ作業を行ってください。

### 9.3.2 [Result: failed, error code:100, error message Open Dongle Failed] エラー

【問題】：

ROCKEY2 のエンベロープツールでソフトウェア（EXE ファイル）をエンベロープすると、[Result: failed, error code:100, error message Open Dongle Failed] エラーが発生します。

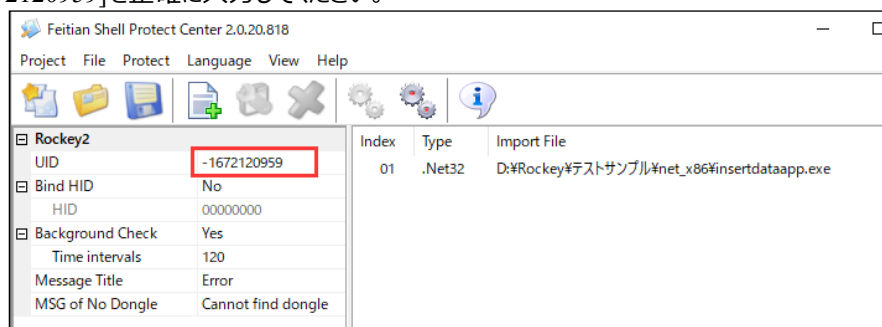
【原因】：

ROCKEY2 の UID が、エンベロープツールに指定された UID と異なるため、オープンエラーが発生します。

【解決策】：

ROCKEY2 の UID をエンベロープツールの左上に正確に入力してください。

※UID がマイナス数値で生成される場合、「-」も含めてエンベロープツールに入力してください。例えば、SeedCode を [test] と設定すると、生成された UID が[-1672120959]となります。エンベロープツールの UID 欄には[-1672120959]を正確に入力してください。



### 9.3.3 [Result: failed, error code:57, error message ACCESS TARGE FILE FAIL] エラー

【問題概要】：

ROCKEY2 のエンベロープツールでソフトウェア（EXE ファイル）をエンベロープすると、[Result: failed, error code:57, error message ACCESS TARGE FILE FAIL] エラーが発生します。

【原因】：

出力パスに書き込み権限がない可能性があります。

【解決策】：

エンベロープされたソフトウェアの出力パスに書き込み権限があるかどうかを確認してください。必要であれば、出力先フォルダーの書き込み権限を設定してください。

### 9.3.4 [Result: failed, error code:48, error message PE SAVE FAIL] エラー

【問題】：

ROCKEY2 のエンベロープツールを使用してソフトウェア（EXE ファイル）をエンベロープすると、[Result: failed, error code:48, error message PE SAVE FAIL] エラーが発生します。

【原因】：

エラーコード 48 は、エンベロープツール実行時に中間ファイルが作成できないときに発生します。

【解決策】：

以下の項目を確認してください。

1. エンベロープツールを CD から起動している場合、中間ファイルが作成できない可能性があります。そのため、エンベロープツールフォルダ（SDK¥Windows¥Utilities¥Envelope）をローカルの HDD にコピーしてから実行してください。
2. エンベロープツールが保存されているフォルダに書き込み権限があるか確認してください。

### 9.3.5 エンベロープされたアプリを実行すると [runtime.dll が見つからない] エラー

【問題】：

エンベロープされたソフトウェアを実行すると、[runtime.dll が見つからない] エラーが発生して、起動できません。

【原因】：

.Net ソフトウェアをエンベロープすると、エンベロープされた EXE 以外にも runtime.dll が自動的に生成されます。この.dll ファイルは、エンベロープされた EXE が実行する際に必要なファイルです。当該ファイルが見つからない場合、エラーが発生します。

【解決策】：

エンベロープされた .Net ソフトウェアのフォルダに runtime.dll が存在するかどうかを確認してください。もし存在しない場合は、エンベロープツールを使用して再度ソフトウェアをエンベロープし、生成された runtime.dll を利用してください。

### 9.3.6 ソフトウェアがエンベロープされた後、起動できません

【問題】：

Microsoft .NET Framework を利用したソフトウェアがエンベロープツールで保護された後、起動できません。

【原因】：

エンベロープツールが Microsoft .NET Framework の一部の機能に適応できない可能性があります。

【回避策】：

1. エンベロープ対象を減らしてみてください。

エンベロープツール⇒ [.NET Envelope Option] ⇒「Method List」がデフォルトで [Encrypt All] に設定されており、すべてのメソッドがエンベロープ対象になります。

[Method List] 右側の [...] をクリックし、一部の関数やコンポーネントのチェックを外してから再度エンベロープを実行し、問題が改善するかどうかを確認してください。

2、API 方式を検討してください。

エンベロープツールが十分な対応をできない場合は、第 7 章の API 方式を検討してください。

### 9.3.7 Linux のエンベロープツールについて

【問題】：

エンベロープツールは、Linux 用の実行ファイルもプロテクトできますか？

【回答】：

ROCKEY2 製品は Linux 版のエンベロープツールが提供されておりません。

Linux 版のエンベロープツールを利用したい場合、同シリーズの ROCKEY4ND 製品をご検討ください。

## 9.4 API に関する FAQ

### 9.4.1 ROCKEY2 の API をどのように呼び出しますか？

ROCKEY2 の SDK には、各開発言語に対応したライブラリが用意されています。  
基本的には、そのライブラリを利用して ROCKEY2 の API 関数を呼び出します。

### 9.4.2 .NET アプリケーションではどのライブラリを利用できますか？

Visual Studio .NET アプリケーション向けに、以下の 4 つの保護方式を提供しています。

- ① Class ライブラリ
- ② COM コンポーネントライブラリ
- ③ DLL 動的ライブラリ
- ④ LIB 静的にライブラリ

SDK にサンプルが格納されています：

ROCKEY2 SDK V2.43¥Windows¥Samples¥VS2017

※詳細な使用方法については、各サンプルの Readme ファイルを参照してください。

どの方式を利用するかは、ご要望に応じて選択できますが、③の動的ライブラリ（DLL）を推奨します。

※ ほかの FAQ については、弊社のウェブサイト（ <https://ftsafe.co.jp/support/rockey/> ）もご参照ください。