

m.magic FZ dll 説明書

revision 82

目次

シェアウェア (2017.01)	4
利用条件	4
ライセンス取得	4
メイン (2015.04)	5
使用する前に	5
サポート方法	5
準備 (2015.04)	6
使い方 (2017.01)	7
概要	7
ストラテジ(戦略)A	7
ストラテジB	8
ストラテジC	9
ストラテジX	9
認証コード	9
mfzGetValidationCode	9
使用できる時間値フォーマット	10
保存すべきデータ	11
Keymart	11
Keymart(プロダクトキー生成機能)	11
Keymart(アプリケーションID移動機能)	11
サンプル	11
関数一覧 (2012.06)	12
mfzInitialize (2014.01)	13
mfzVerifyProductKey (2024.01)	14
mfzVerifyValidationCode (2012.06)	15
mfzVerifyValidationCode (2013.05)	16
mfzGetValidationCode (2012.06)	17
mfzGetTime (2012.06)	18
mfzGetTimeSpan (2012.06)	19
mfzGetSystemTime (2012.06)	20
mfzGetCode (2012.06)	21
mfzPutCode (2017.01)	22
mfzCreateTrunk (2012.06)	23
mfzGetTrunkName (2012.06)	24
mfzCustomerInfoForm_1a (2012.06)	25
mfzUninstall_1a (2012.06)	26
mfzShowInputDialog (2017.01)	27
mfzGetFileVersion (2017.01)	28
仕様 (2014.01)	29
FZ DLL	29
Keymart	30
使用条件 (2023.01)	31
追加情報 (2017.01)	32
プロダクトキー認証の概要 (2014.01)	34
変更履歴 (2017.01)	35

m.magic FZ DLL.....	35
Keymart.....	36
Keymart.dll.....	37
Hermit crab.....	38
用語集 (2014.01)	39
開発環境 (2023.03)	40

シェアウェア (2017.01)

ダウンロードありがとうございます。m.magic FZ DLL は、シェアウェアです。継続使用する場合は、下記の方法で 2,222 円 を送金してください。

利用条件

1. m.magic FZ DLL に関するプログラム及びドキュメント等の著作権は、作者が保有します。
2. mgc2fz.dll を継続使用する、または、評価の目的以外での使用する場合、必ずシェアウェア料の支払いが必要になります。
3. 試用期間中は、"4.再配布可能ファイル" に記載されるファイルの再配布はできません。
4. 再配布可能ファイル mgc2fz.dll, mfz2i.dll
5. このプログラムの使用によるいかなる損害に対して作者は責任を負いません。

ライセンス取得

A. ベクターシェアレジ利用

作者名：丸山一利

ベクターライブラリ作者番号：PA031043

作品名：m.magic FZ DLL

レジ作品番号：SR255080

B. 銀行口座

みずほ銀行 雷門前支店 支店番号 601 (普)1498221 マルマカスト

メイン (2015.04)

m.magic FZ DLL は、シェアウェア等に利用できるプロダクトキー発生、認証システムです。暗号理論を応用し、一個のアプリケーション I D で数十万個のプロダクトキーを使用できる。また、認証コードが使用できるため、ユーザーに入力させたプロダクトキーを記録する必要がない。さらに、Visual Studio Installer からプロダクトキーを診断する機能を搭載しているためインストール時にプロダクトキーを診断することも可能です。プロダクトキーを生成するプログラムキーマートは、ステップ・バイ・ステップ方式となり使いやすくなった。旧 m.magic F DLL の互換性をすべて捨て、心残りだった部分を作り直すことで洗練度が大幅に向上、新機能も追加された。

1. 数十万人を識別可能
2. 認証コード使用可能
3. 経過時間カウント可能
4. Visual Studio Installer を使用しプロダクトキー診断可能
5. x 6 4 使用可能
6. サンプルあり (MFC, C#, VB.NET)

使用する前に

使用する前に[準備](#)を参照してください。

サポート方法

e-mail のみのサポートです。回答には一週間程度かかることがあります。

e-mail アドレス: m.magic@dream.nifty.jp

準備 (2015.04)

1. 必要ならマイクロソフトダウンロードセンターから .NET Framework 3.5 Service pack 1 をダウンロードする
2. 必要なら .NET Framework 3.5 Service pack 1 をインストールする
3. ダウンロードした "mg*-.*.zip" ファイルを伸張する
4. Keymart をインストールする。
プロダクトキー : 0125eG8svZFuoCp5 または、0141twmCafY9DE6C
5. Windows SDK 付属の orca.msi (オルカ) をインストールする
6. サンプル中の D L L を ロードするクラス等をコピーして使用する

使い方 (2017.01)

概要

アプリケーション I D ごとにプロダクトキーを発生させそのプロダクトキーを診断します。アプリケーション I D、プロダクトキーを発生させるのが、キーマート(Keymart.exe) です。プロダクトキーを診断するのが、FZ DLL です。

アプリケーション I D が異なれば使用できるキーも異なる。暗号理論を使用しているためひとつのアプリケーション I D で数十万個の異なるプロダクトキーの正常性を確認できる。

プロダクトキーが正しかった場合、認証コードを作成することが可能です。プロダクトキーをそのまま保存する必要はない。認証コードを保存すればよい。認証コードを盗まれても、認証コードを作った P C でなければ、使用することはできません。

ストラテジ(戦略)A

V S I (Visual Studio Installer) を使用しインストール時に、プロダクトキーを入力する。入れない場合インストールは失敗する。正常だった場合、トランクファイルに認証コードを書き込む。さらに起動時に、認証コードの正常性を診断する。アプリケーション I D は、インストーラー用 FZ DLL と アプリケーションに内包する。(FZ DLL は、2つ必要)

1. Keymart を使いアプリケーション I D、プロダクトキー、インストーラー用 F Z D L L (mfz2i.dll)を生成する。
2. 配布するアプリケーションに、アプリケーション I D を内包する。
3. V S I 用のセットアッププロジェクトを作り、**起動時 FZ DLL** と **インストーラー用 FZ DLL (mfz2i.dll)** を含める。この時、インストーラー用 FZ DLL の **Exclude プロパティ** を True にする。
4. ユーザー情報ダイアログを追加する。SerialNumberTemplate プロパティに <????-????-????-????> をセットする。ShowSerialNumber プロパティを True にする。
5. カスタム動作のアンインストールに、インストーラー用 FZ DLL を含め、mfzUninstall_1a を **EntryPoint プロパティ** にセットする。
6. マージモジュールを作る。
7. オルカでマージモジュールを開く。mfzCustomerInfoForm_1a 関数を呼び出せるように改良するためです。
8. 左側の Tables から CustomAction を選ぶ。下表を参考に、マゼンダ行を追加する。ブラック文字は、状況によりこの表と異なることがある。

9. 左側の Tables から ControlEvent を選ぶ。元の文字をブルー文字に変更する。ブラック文字は、状況によりこの表と異なることがある。

Dialog_	Control_	Event	Argument	Condition	order
CustomerInfoForm	NextButton	DoAction	CustomerInfoForm_1a	CustomerInfoForm_ShowSerial="1"	0
CustomerInfoForm	NextButton	EndDialog	Return	ProductID=""	1
CustomerInfoForm	NextButton	NewDialog	[CustomerInfoForm_NextArgs]	CustomerInfoForm_NextArgs<>"" AND (ProductID<>"" OR CustomerInfoForm_ShowSerial="0")	2

10. マージモジュールを保存する。
11. セットアップを開始し、ユーザーにプロダクトキーを入力させる。プロダクトキーが正しければ、mfzCustomerInfoForm_1a 関数は、認証コードを作り、トランクファイルに書き込む。正しくなければ、セットアップは失敗する。
12. アプリケーション起動ごとに、認証コードを診断する。



インストーラー用 FZ DLL の Exclude プロパティを True にするのは、インストーラー用 FZ DLL をインストールしないためです。mfz2i.dll は、アプリケーション起動時の診断には使用できない。

ストラテジ B

アプリケーション起動時に、プロダクトキーを入力する。入れない場合、機能制限、時間をカウントする。正常だった場合、トランクファイルに認証コードを書き込む。さらに起動時に、認証コードの正常性を診断する。アプリケーション ID は、FZ DLL に内包する。

1. Keymart を使いアプリケーション ID、プロダクトキー、マジックコード、FZ DLL を生成する。
2. 配布するアプリケーションに、マジックコードを内包する。
3. ユーザーにプロダクトキーを入力させる。
4. プロダクトキーが正しければ認証コードを作り、トランクファイルに書き込む。正しくなければ、機能制限、時間をカウントする。
5. アプリケーション起動ごとに、認証コードを診断する。



マジックコードがなければ F Z D L L からアプリケーション I Dを読み取れない。

ストラテジC

アプリケーション起動時に、プロダクトキーを入力する。入れない場合、機能制限、時間をカウントする。正常だった場合、トランクファイルに認証コードを書き込む。さらに起動時に、認証コードの正常性を診断する。アプリケーション I Dは、アプリケーションに内包する。

1. Keymart を使いアプリケーション I D、プロダクトキーを生成する。
2. 配布するアプリケーションに、アプリケーション I Dを内包する。
3. ユーザーにプロダクトキーを入力させる。
4. プロダクトキーが正しければ認証コードを作り、トランクファイルに書き込む。正しくなければ、機能制限、時間をカウントする。
5. アプリケーション起動ごとに、認証コードを診断する。

ストラテジX

アプリケーション起動時に、プロダクトキーを入力する。入れない場合、機能制限、時間をカウントする。正常だった場合、"ある場所" に認証コードを書き込む。さらに起動時に、認証コードの正常性を診断する。アプリケーション I Dは、"ある場所" に内包する。

認証コード

プロダクトキーが正しかった場合認証コードを作ります。作成条件は、HDDフォーマットナンバー、CPUの種類、HDDフォーマットナンバーとCPUの種類の三通り。HDDフォーマットナンバー(CPUの種類)が異なる場合、正しい認証コードにならない。従って、認証コードをHDDフォーマットナンバー(CPUの種類)の異なるPCにコピーしても使用できない。アプリケーション I DをM F Z D L Lに内包しない場合、このいずれかをイニシャライズ時に設定する。アプリケーション内包時は、自動判定される

mfzGetValidationCode

M F Z D L L バージョン 1.1200 よりプロダクトキー診断後、正しいときのみ認証コードを作るようにした。mfzVerifyProductKey 関数の使用を省ける。

時間値フォーマット

M F Z D L L バージョン 1.1201 より時間値のフォーマットBが追加された。従来、時間値を取得したHDDでのみ正常な時間として使用できたが、フォーマットBでは、HDDに関係なく正常な時間値として使用できる。M F Z D L L の初期化関数でどちらのフォーマットを使うのかが選択できる。ただし、アプリケーションIDを内包した場合、使用できるフォーマットが固定される。M F Z D L L バージョン 1.1402 以降は、**フォーマットCの使用を推奨**します。フォーマットA、フォーマットBは、互換性を保つために残してあります。

時間値フォーマット	正常な時間値と認識させるための条件
A	HDDとユーザー名
B	ユーザー名
C	ユーザー名

使用できる時間値フォーマット

M F Z D L L バージョンごとに使用できる時間値フォーマットが異なる。

M F Z D L L バージョン	アプリケーション ID	フォーマットA	フォーマットB	フォーマットC
1.1100 以降	内部	○	×	×
	外部	○	×	×
	インストーラー	○	×	×
1.1201 以降	内部	×	○	×
	外部	○	○	×
	インストーラー	×	○	×
1.1402 以降	内部	×	×	○
	外部	○	○	○
	インストーラー	×	×	○

保存すべきデータ

○が保存すべきデータ

	プロダクトキー	認証コード	時間値
プロダクトキー、認証コードを使用する	×	○	×
プロダクトキー、認証コード、時間値使用する	×	○	○
プロダクトキーのみ使用する	○	×	×
プロダクトキー、時間値を使用する	○	×	○

Keymart

Keymart64_*_*.msi または、Keymart32_*_*.msi を実行しインストールする。途中、プロダクトキー（シリアル番号）を入力する。プロダクトキーは、012GWT01WyrfDVWI または、012hSNU5pAQVDADG。

Keymart（プロダクトキー生成機能）

プロダクトキーは、%drive%:\my document\Product keys\<manufacture>\<product name>\Product keys [<manufacture>-<product name>].txt ファイルに出力される。また、x86(x64) フォルダ内の mfv2i.dll は、インストーラー用の F Z D L L である。

Keymart（アプリケーション I D 移動機能）

旧バージョンの F Z D L L に書込まれたアプリケーション I D を新バージョンの D L L に移し替えます。指示に従い必要な情報を入力する。マジックコード、プロダクトキーは、移し替えた後そのまま使える。x 6 4 用のアプリケーション I D を移し替える場合は、6 4 ビット用の W i n d o w s を使用すること。時間値については、フォーマット B に固定される。

サンプル

正しいアプリケーション I D、マジックコード等を設定しなければ正しく動作しない。また、Visual studio 2003 で作成されているため新バージョンで読込んだ場合、コンパイル時にワーニングが出ることもある。修正してください。

関数一覧 (2012.06)

関数名	機能説明
mfzInitialize	D L L を初期化する
mfzVerifyProductKey	プロダクトキーを診断する
mfzVerifyValidationCode	認証コードを診断する
mfzGetValidationCode	認証コードを作成する
mfzGetTime	時間値を取得する
mfzGetTimeSpan	経過時間を取得
mfzGetSystemTime	時間値をシステムタイムに変換する
mfzGetCode	トランクファイルから認証コード等を取得
mfzPutCode	トランクファイルに認証コード等を書き込む
mfzCreateTrunk	トランクファイルを作成
mfzGetTrunkName	トランクファイル名を取得
mfzCustomerInfoForm_1a	インストーラー用プロダクトキー診断関数
mfzUninstall_1a	アンインストーラー用関数
mfzShowInputDialog	プロダクトキー入力ダイアログを表示
mfzGetFileVersion	ファイルのバージョン情報を取得します

mfzInitialize (2014.01)

```
int mfzInitialize( const LPVOID lpInitialParams );
```

引数

const LPVOID lpInitialParams

初期化用構造体のポインタ。

構造体

```
typedef struct
{
    ULONG nSize;           // この構造体のサイズ
    ULONG nFunctions;      // プロダクトキーのバージョン
    BYTE  szIdentity[MFZ_ID_KV41_SIZE]; // アプリケーション I D
} MFZ_INI_PARAMS_A, *LPMFZ_INI_PARAMS_A;
```

```
typedef struct
{
    ULONG nSize;           // この構造体のサイズ
    ULONG nFunctions;      // プロダクトキーのバージョン
    ULONG nMagicCode;      // D L L にアクセスするための識別番号
    SYSTEMTIME sysPutTime; // D L L に I D を書き込んだ時間
} MFZ_INI_PARAMS_B, *LPMFZ_INI_PARAMS_B;
```

nFunctions

上記の構造体の *nFunctions* には、プロダクトキーバージョンと時間値のフォーマット、認証コード作成時に含めるハードウェアセットをする。

戻り値

0

解説

時間値フォーマット *MFZ_INI_TIME_FORMAT_B* は、D L L バージョン 1.1201 以降で利用できる。また、*MFZ_INI_PARAMS_B* 構造体を使用する場合、*MFZ_INI_TIME_FORMAT_B* での固定動作になる。*MFZ_INI_TIME_FORMAT_A* を選択できない。

D L L バージョン 1.1400 から認証コード作成条件に、C P U の種類を加えるかどうかを選択できる。

mfzVerifyProductKey (2024.01)

```
int mfzVerifyProductKey( const CHAR* lpszProductKey );
```

引数

const CHAR lpszProductKey*
プロダクトキー

戻り値

MFZ_TRUE	正しいキーだった
MFZ_FALSE	不正なキーだった
MFZ_FAILURE	障害が発生した

解説

指定のプロダクトキーの正常性を診断する。

mfzVerifyValidationCode (2013.05)

```
int mfzVerifyValidationCode( const CHAR szUserName[MFZ_USER_NAME_SIZE], const BYTE  
szValidationCode[MFZ_VALDATIONCODE_SIZE] );
```

引数

const CHAR szUserName[MFZ_USER_NAME_SIZE]

ユーザー名

const BYTE szValidationCode[MFZ_VALDATIONCODE_SIZE]

認証コード

戻り値

MFZ_TRUE 正しい認証コード

MFZ_FALSE 認証コードが正しくない

解説

認証コードの正常性を判断します。

mfzGetValidationCode (2012.06)

```
int mfzGetValidationCode( const CHAR* lpszProductKey, const CHAR* lpszUserName, BYTE*  
lpszValidationCode );
```

引数

const CHAR lpszProductKey*

プロダクトキー

const CHAR lpszUserName*

ユーザー名

BYTE lpszValidationCode*

認証コード

戻り値

MFZ_SUCCESS 成功

MFZ_FAILURE 失敗

解説

プロダクトキーが正しいときのみ認証コードを作成する。正しくない場合 MFZ_FAILURE を返す。従って、mfzVerifyProductKey を使用する手間を省ける。ユーザー名は、必須。

mfzGetTime (2012.06)

```
int mfzGetTime( LPCSTR lpszUserName, BYTE* lpszTime );
```

引数

LPCSTR lpszUserName

サイズが MFZ_USER_NAME_SIZE のユーザー名

BYTE lpszTime*

サイズが MFZ_TIME_SIZE の時間値

戻り値

MFZ_SUCCESS 成功

MFZ_FAILURE 失敗

解説

現在時間を F Z D L L 固有の値で返す。時間値は、ユーザー名で固有となるので、ユーザー名は必須。

mfzGetTimeSpan (2012.06)

```
int mfzGetTimeSpan( LPCSTR lpszUserName, const BYTE* lpszTime );
```

引数

LPCSTR lpszUserName

サイズが MFZ_USER_NAME_SIZE のユーザー名

const BYTE lpszTime*

サイズが MFZ_TIME_SIZE の時間値

戻り値

正の整数	成功、日数
MFZ_FAILURE	失敗
MFZ_TMS_INVALID_TIME	不正な時間値になった

解説

指定の F Z D L L 固有の時間値と現在時間との日数を返す。

mfzGetSystemTime (2012.06)

```
int mfzGetSystemTime( LPCSTR lpszUserName, BYTE* lpszTime, SYSTEMTIME* lpsysTime );
```

引数

LPCSTR lpszUserName

ユーザー名

const BYTE lpszTime*

F Z D L L フォーマットの時間値

SYSTEMTIME lpsysTime*

M F C で使用しているシステムタイム構造体

戻り値

MFZ_SUCCESS	成功
MFZ_FAILURE	失敗
MFZ_TMS_INVALID_TIME	無効な時間値になった
MFZ_TMS_SUCCESS_AND_TIME_MADE	成功し、時間値のフォーマットを A から B に変換した

解説

F Z D L L 固有の時間値を一般的なシステムタイムに変換する。ユーザー名は、mfzGetTime 関数を使用したときのユーザー名と同一にする。時間値フォーマット B は、D L L バージョン 1.1201 以降で利用できる。イニシャライズ時に、時間値フォーマット B を選択し、フォーマットが A だった場合、可能ならフォーマット変換を行い lpszTime に結果を返す。

mfzGetCode (2012.06)

```
int mfzGetCode( LPCSTR lpszFileName, BYTE* lpszValidationCode, BYTE* lpszTime, CHAR* lpszUserName );
```

引数

LPCSTR lpszFileName

トランクファイル名

BYTE lpszValidationCode*

サイズが MFZ_VALDATIONCODE_SIZE の認証コード

BYTE lpszTime*

サイズが MFZ_TIME_SIZE の時間値

CHAR lpszUserName*

サイズが MFZ_USER_NAME_SIZE のユーザー名

戻り値

MFZ_SUCCESS	成功
MFZ_FAILURE	失敗
MFZ_COD_ERR_INITIAL_STATE	トランクファイルが初期状態
MFZ_COD_ERR_LOAD	トランクファイルのロードに失敗した
MFZ_COD_ERR_CRC	トランクファイル内のデータをCRCチェックしたが異常だった
MFZ_COD_ERR_NO_TRUNK	既定のトランクファイルではなかった
MFZ_COD_ERR_RESOURCE	トランクファイルからリソースを読み込めなかった
MFZ_COD_ERR_FILE_NOT_FOUND	指定のファイルが見つからなかった
MFZ_COD_ERR_INVALID_FILE_NAME	不正なファイル名だった

解説

認証コード、時間値、ユーザー名をトランクファイルから読み出す。

mfzPutCode (2017.01)

```
int mfzPutCode( LPCWSTR lpszFileName, const BYTE* lpszValidationCode, const BYTE* lpszTime, const  
CHAR* lpszUserName );
```

引数

LPCWSTR lpszFileName

トランクファイル名

BYTE lpszValidationCode*

サイズが MFZ_VALDATIONCODE_SIZE の認証コード

BYTE lpszTime*

サイズが MFZ_TIME_SIZE の時間値

CHAR lpszUserName*

サイズが MFZ_USER_NAME_SIZE のユーザー名

戻り値

MFZ_SUCCESS	成功
MFZ_FAILURE	失敗
MFZ_COD_ERR_INITIAL_STATE	トランクファイルが初期状態
MFZ_COD_ERR_LOAD	トランクファイルのロードに失敗した
MFZ_COD_ERR_CRC	トランクファイル内のデータをCRCチェックしたが異常だった
MFZ_COD_ERR_NO_TRUNK	既定のトランクファイルではなかった
MFZ_COD_ERR_RESOURCE	トランクファイルからリソースを読み込めなかった
MFZ_COD_ERR_FILE_NOT_FOUND	指定のファイルが見つからなかった
MFZ_COD_ERR_INVALID_FILE_NAME	不正なファイル名だった

解説

認証コード、時間値、ユーザー名をトランクファイルに書込む。

mfzCreateTrunk (2012.06)

```
int mfzCreateTrunk( LPCSTR lpszFileName, const DWORD dwAttributes );
```

引数

LPCSTR lpszFileName

トランクファイルの名前とパス。

const DWORD dwAttributes

この属性が作られたファイルに設定される。使えるのは FILE_ATTRIBUTE_HIDDEN / FILE_ATTRIBUTE_NORMAL / FILE_ATTRIBUTE_READONLY です。

戻り値

MFZ_SUCCESS 成功

MFZ_FAILURE 失敗

解説

トランクファイルを作成する。第一引数に名前と場所を設定する。

mfzGetTrunkName (2012.06)

```
int mfzGetTrunkName( const ULONG nFunction, LPCSTR lpszManufacturer, LPCSTR lpszProductName,
const INT nSize, LPSTR lpszReturn );
```

引数

const ULONG nFunction

この関数の動作を指定する変数です

MFZ_GTN_GET_APPDATA_PATH

Common App path 取得

MFZ_GTN_GET_TRUNK_PATH

トランクファイルがある規定のパス取得

MFZ_GTN_GET_TRUNK_PATH_AND_NAME

トランクファイル名と既定のパス取得

MFZ_GTN_GET_TRUNK_NAME

トランクファイル名取得

MFZ_GTN_MAKE_DIRECTORY

ディレクトリ作成

LPCSTR lpszManufacturer

マニファクチャラ名です

LPCSTR lpszProductName

製品名です

const INT nSize

LPSTR lpszReturn 変数のバイト数です

LPSTR lpszReturn

結果を返す文字列変数です

戻り値

MFZ_SUCCESS 成功

MFZ_FAILURE 失敗

解説

この関数を必ず使う必要はない。パスを含むトランクファイル名を返す。パスは、Common App path になる。

mfzCustomerInfoForm_1a (2012.06)

インストーラー用プロダクトキー診断関数

mfzUninstall_1a (2012.06)

アンインストーラー用関数

mfzShowInputDialog (2017.01)

```
int mfzShowInputDialog( LPCWSTR lpszTitle, LPCWSTR lpszMsg, const int nFigure, CHAR* lpszUserName,  
CHAR* lpszProductKey );
```

引数

LPCWSTR lpszTitle

ダイアログ枠に表示される文字

LPCWSTR lpszMsg

ダイアログに表示される文字

const int nFigure

プロダクトキーの桁数。16 か 18 桁。

CHAR lpszUserName*

ユーザー名です。サイズは、MFZ_USER_NAME_SIZE

CHAR lpszProductKey*

プロダクトキー。ここに、ユーザーが、入力したプロダクトキーが入る。

戻り値

MFZ_SUCCESS	成功、OK ボタンが押された
MFZ_FAILURE	失敗
MFZ_CANCEL	キャンセルボタンが押された

解説

この関数を必ず使う必要はない。ユーザー名に、スペース文字を入れるとエディットコントロールにログイン名が表示される。先頭が、NULL 文字の場合コンピュータ名が表示される。文字列を入れた場合は、編集できない状態でその文字が表示される。

mfzGetFileVersion (2017.01)

ファイルのバージョン情報を取得します

仕様 (2014.01)

FZ DLL

二重起動

二重起動はできない。

mzfInitialize 関数

イニシャライズは、1 回しかできない。必ず行うこと。

MfzVerifyProductKey & mfzVerifyValidationCode 関数

プロダクトキー、認証コードの正常性診断は、3 回以内に行うこと。

FZ DLL へのアプリケーション ID 書込み

アプリケーション ID は、FZ DLL に内包できる。ただし、一度しか書き込めない。書き換えはできない。
読み書きするには、マジックコードが必要。

アプリケーション ID

アプリケーションを識別するためのアプリケーション ID のサイズは、192 バイト。

プロダクトキー

プロダクトキーが使用する文字は、半角 "A"-"Z", "a"-"z", "0"-"9" です。大文字と小文字は区別される。

認証コードの許可範囲

ユーザー名、HDD のフォーマットナンバー、CPU の種類を使用し認証コードを作る。ユーザー名、HDD のフォーマットナンバー、CPU の種類が異なる場合、正常な認証コードにならない。認証コード診断で、不正な認証コードになる。

認証コード作成時の CPU の種類

使用できる CPU の種類は、インテルと AMD。

時間値の許可範囲

ユーザー名を使用し、時間値を作る。（時間値フォーマット B）ユーザー名が異なる場合、正常な時間値戻せない。アプリケーション ID を FZ DLL に内包しない場合、HDD のフォーマットナンバーを含めるかどうかを選択できる。（選択できるのは、DLL バージョン 1.1201 以降の機能、それ以前は、必ず HDD フォーマットナンバーが含まれる）

動作スピード

正常なプロダクトキーを入力しなかった場合等、ウェイトがかかる。

mfzCustomerInfoForm_1a 関数

VSI 専用です。V S I 以外では使えません。

キーが正しかった場合、トランクファイルが、%drive%:[CommonAppDataFolder][マニファクチャラ名][プロダクト名]*.trk に作られる。

mfzUninstall_1a 関数

VSI 専用です。V S I 以外では使えません。

アンインストール時、%drive%:[CommonAppDataFolder][マニファクチャラ名][プロダクト名]*.trk を消去する。ディレクトリ内に、ファイルがない場合、%drive%:[CommonAppDataFolder][マニファクチャラ名][プロダクト名] と %drive%:[CommonAppDataFolder][マニファクチャラ名] のディレクトリは消去される。

Keymart

二重起動

二重起動はできない。

アプリケーションID

アプリケーションIDを指定してプロダクトキーを生成することはできない。

プロダクトキー

プロダクトキーの桁数は、プロダクトキーの発生数により、16、18桁が自動的に決定される。

アプリケーションID移動機能

M F Z D L L内のアプリケーションIDを移動する機能を有する。移動するためには、認証のためマニファクチャラ名、プロダクトキー等を入力する必要あり。

使用条件 (2023.01)

FZ DLL 仕様

- ・ M F C をスタティックライブラリとして使用
- ・ 文字は U N I C O D E
- ・ .NET Framework は不使用

Keymart

- ・ M F C をスタティックライブラリとして使用
- ・ NET Framework 3.5 を使用

必要ソフト

- ・ .NET Framework 3.5
- ・ Visual C++ or Visual C# or Visual Basic.NET

動作可能 OS

○ 可 - 未確認 × 不可

OS	x86	x64(WOW64)	x64
Windows10	-	○	○
Windows 8.1	○	-	-
Windows 8	○	○	○
Windows 7	○	○	○
Windows Vista	○	-	-
Windows XP	-	-	-

追加情報 (2017.01)

M F Cをスタティックとした(2017.01)

バージョン 2.1600 よりM F Cをスタティックとした。すなわちランタイムをインストールする必要はない。

アプリケーション I Dの移動(2015.04)

異なるバージョンの F Z D L L にアプリケーション I Dを移動した場合 時間値フォーマットは、Cになる。

時間値フォーマット変換(2015.04)

mfzGetSystemTime 関数が TMS_SUCCESS_AND_TIME_MADE を返した場合時間値フォーマットが変換されています。時間値を保存してください。

認証コード(2014.01)

F Z D L L バージョン 1.1400 よりC P Uの種類を作成条件に加えた。使えるC P Uは、インテルとA M D。

Hermit Crab(2013.10)

Hermit Crab は、廃止となり、その機能を Keymart に統合した。

サンプルプログラム (2013.01)

C # & V B . N E T のサンプルプログラムは、Visual Studio 2005 で作成されている。 .NET Framework 2.0 の機能を使用している。

アクティブロード (2012.12)

C # & V B . N E T のサンプルプログラムでは、D L Lをスタティックロードとしていたが、アクティブロードに変更した。従って、ロードしたプログラムが動作している状態で F Z D L Lをアンロードできる。

時間値フォーマット (2012.05)

正しくフォーマットを指定しない場合正しい時間値にはならない。

起動失敗 (2012.02)

Keymart と Hermit crab は、同時使用はできない。どちらか一方は、起動に失敗する。

インストーラマージモジュール (2012.02)

インストーラマージモジュール から mfzCustomerInfoForm_1a 関数を呼び出すには、オルカを使いマージモジュールを修正する必要がある。

F Z D L L に内包したアプリケーション I D (2012.02)

F Z D L L バージョン 1.1100-1.1101 と 1.1102 では、アプリケーション I D の書込み書式が異なる。新バージョンの F Z D L L を使いたい場合は、付属のハミットクラブを使用しアプリケーション I D を移動する。

プロダクトキーバージョン (2012.02)

Keymart Version 1.1102 は、プロダクトキーバージョン 4.1 のみを生成する。F Z D L L は、プロダクトキーバージョン 4.0 & 4.1 双方診断できる。イニシャライズ時に、診断したいプロダクトキーのバージョンをセットする。

x 6 4 ビット用インストーラマージモジュール (2011.07)

x 6 4 ビット用インストーラマージモジュールを作ると "警告: エントリ ポイント 'mfzUninstall_1a' がカスタム動作 'mgc2fz.dll' のモジュール 'D:¥mgc2fz.dll' で見つかりません。" というメッセージが出ることもある。V S I に問題がある? 詳細は不明です。そこで、シャチ!? (オルカ) を使ってみます。警告の出たマージモジュールをオルカで開きます。左にある "Tables" から "CustumAction" を見つけ出す。選択する。右側の "Target" から _mfzUninstall_1a@4 のような文字を見つける。mfzUninstall_1a に書き換える。マージモジュールを保存する。正常にインストールできるようになります。

オルカ (2011.07)

インストーラマージモジュールを編集するソフトです。Windows SDK に含まれています。%drive %¥Program Files¥Microsoft SDKs¥*¥¥bin ディレクトリに orca.msi というファイルがあるのでインストールする。

プロダクトキーバージョン (2011.07)

F Z D L L (Ver1.1100) は、プロダクトキーバージョン 4.0 専用です。

再配布 (2011.07)

配布する実行ファイルと同じディレクトリに F Z D L L を配布してください。また、配布先の環境に、.NET Framework は、必要ありません。

ランタイム D L L (2011.07)

外部ランタイム方式で動作している。Visual C++ 2008 SP1 再頒布可能パッケージをインストールしてください。

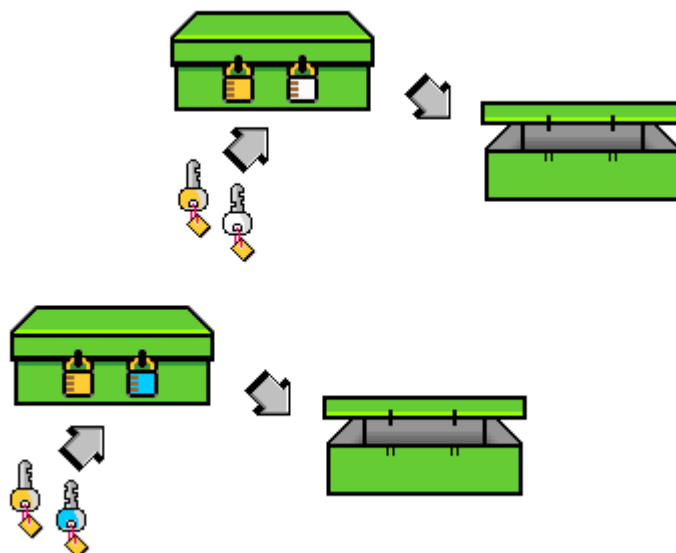
サンプルプログラム (2011.07)

VisualStudio 2003 で作成されている。新しいバージョンで読み込んだ場合変換後、コンパイル時にワーニングが出ることもある。修正してください。

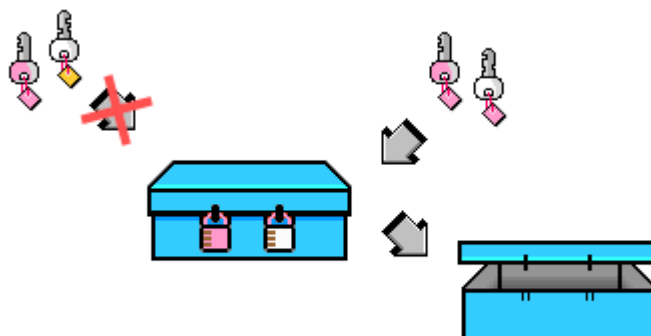
プロダクトキー認証の概要 (2014.01)

概要

暗号理論を応用することで、複数のプロダクトキーの使用を可能にしている。鍵を2つ使える点が重要です。例えば、ある緑色の箱があったとする。緑色の箱には、二つの錠が設置できる。黄色と白の錠がついている。従って、緑の箱を二人に配るとき、それぞれに異なる色の鍵を渡すことが可能になる。



青い箱の場合はどうだろうか。青い箱には、黄色ではなく、桃色の錠がついている。従って、桃色の白鍵を使用しなければならない。黄色の白鍵を使うことはできない。



変更履歴 (2017.01)

m.magic FZ DLL

Version	日付	主要内容
2.1600	2017.01	<ul style="list-style-type: none"> ● Visual Studio 2015 でコンパイルした ● M F C をスタティックライブラリとした ● 文字をユニコードとした
1.1402	2015.05	<ul style="list-style-type: none"> ● 時間値のフォーマット C を追加した ● 技術的な改良
1.1401	2015.03	<ul style="list-style-type: none"> ● 技術的な改良
1.1400	2014.02	<ul style="list-style-type: none"> ● 認証コード作成時に、C P U の種類を加えるために必要な個所を修正
1.1300	2013.05	<ul style="list-style-type: none"> ● CustomerInfoForm_1a 関数内でトランクファイルを見つけた時消去するようにした
1.1201	2012.05	<ul style="list-style-type: none"> ● mfzGetSystemTime 関数に時間値フォーマット A を B に変換する機能追加 ● 時間値のフォーマット変更
1.1200	2012.03	<ul style="list-style-type: none"> ● プロダクトキー診断後、正しいときのみ認証コードを作るようにした ● 不正動作時のウェイト機構等の変更
1.1102	2012.02	<ul style="list-style-type: none"> ● プロダクトキーが正しいときのみトランクファイルを作るように mfzCustomerInfoForm_1a 関数を変更 ● mfzCustomerInfoForm_1a 関数にプロダクトキーの文字数診断およびバージョン診断機構追加 ● 不正動作時のウェイト機構等の変更 ● F Z D L L に書き込んだアプリケーション I D を C R C チェックするときのサイズが間違っているのを修正 ● F Z D L L に書き込むアプリケーション I D の書式変更 ● プロダクトキーバージョンを改良し 4.1 とした
1.1101	2011.09	<ul style="list-style-type: none"> ● 二重起動防止機構変更
1.1100	2011.07	<ul style="list-style-type: none"> ● 初期リリース

Keymart

Version	日付	主要内容
6.3	2017.01	<ul style="list-style-type: none"> ● Visual Studio 2015 でコンパイルした
6.2	2015.05	<ul style="list-style-type: none"> ● Ver.1.1402 の Keymart.dll を使用した
6.1	2015.04	<ul style="list-style-type: none"> ● アプリケーション I D 設置場所を非選択式とした ● 図柄変更
6.0	2014.02	<ul style="list-style-type: none"> ● 認証コード作成時に、C P U の種類を加えるために必要な個所を修正 ● 注釈を入れるためのテキストボックスの I M E をオンできるようにした ● マニファクチャラ名とプロダクトキーの文字数は 3 1 文字が正しい。3 2 文字で機能停止する問題修正 ● マニファクチャラ名とプロダクトキーの入力フォームを分離した ● 視覚デザイン変更
5.1303	2013.10	<ul style="list-style-type: none"> ● インストーラー用 FZ DLL を無条件で出力するようにした ● Hermit crab の機能を Keymart に統合した
5.1302	2013.08	<ul style="list-style-type: none"> ● プログラムの洗練度向上 ● 実行ステップ数を増やした ● プロダクトキーの生成数を桁数で判断するようにした ● プロダクトキー出力ファイル内のマジックコードを出力する条件変更
5.1301	2013.06	<ul style="list-style-type: none"> ● 各 Step フォームをコントロールとして表示するようにした ● プロダクトキーの出力先フォルダを変更
5.1300	2013.05	<ul style="list-style-type: none"> ● 全面改良。ステップ・バイ・ステップ式に変更した
4.1201	2012.05	<ul style="list-style-type: none"> ● コントロールのレイアウト見直し ● ユーザー名等を表示するようにした
4.1102	2012.02	<ul style="list-style-type: none"> ● プロダクトキーの取出しに問題があるのを修正
4.1101	2011.09	<ul style="list-style-type: none"> ● 起動時の診断機構修正 ● 障害時の説明を表示するようにした ● 保存ファイル名の初期値変更
4.1100	2011.07	<ul style="list-style-type: none"> ● 初期リリース

Keymart.dll

Version	日付	主要内容
2.1600	2017.01	<ul style="list-style-type: none"> ● Visual Studio 2015 でコンパイルした ● M F C をスタティックライブラリとした ● 文字をユニコードとした
1.1402	2015.05	<ul style="list-style-type: none"> ● アプリケーション I D を移動するとき Validation Code の作成条件に C P U を指定している場合 条件が H D D に代わってしまう問題修正。 ● 時間値フォーマット C を追加した ● 技術的な改良
1.1401	2015.03	<ul style="list-style-type: none"> ● 技術的な改良
1.1400	2014.02	<ul style="list-style-type: none"> ● 認証コード作成時に、C P U の種類を加えるために必要な個所を修正
1.1301	2013.06	<ul style="list-style-type: none"> ● kmtDeleteFileAndDirectory 関数を追加した
1.1300	2013.05	<ul style="list-style-type: none"> ● F Z D L L Version 1.1300 にアプリケーション I D を移動できるようにした ● キー生成時に呼び出されるコールバック関数の呼び出し回数を多くした ● kmtGetDirectory 関数を追加した ● kmtMakeDirectory 関数を追加した ● kmtCheckFile 関数を追加した
1.1201	2012.05	<ul style="list-style-type: none"> ● F Z D L L Version 1.1201 にアプリケーション I D を移動できるようにした ● 時間値のフォーマット変更に伴いフォーマット識別用関数を追加
1.1102	2012.02	<ul style="list-style-type: none"> ● 使用しない乱数を判定する機構の改良 ● プロダクトキーの発生数を二択とした ● プロダクトキーの取出しに問題があるのを修正 ● 不正動作時のウェイト機構等の変更 ● F Z D L L に書き込んだアプリケーション I D の書式変換をする関数の追加 ● F Z D L L に書き込んだアプリケーション I D を C R C チェックするときのサイズが間違っているのを修正 ● F Z D L L に書き込むアプリケーション I D の書式変更 ● プロダクトキーバージョンを改良し 4.1 とした
1.1100	2011.07	<ul style="list-style-type: none"> ● 初期リリース

Hermit crab

Version	日付	主要内容
-.-	2013.10	<ul style="list-style-type: none">● Keymart と統合したため廃止
1.1302	2013.08	<ul style="list-style-type: none">● プログラムの洗練度向上● MagicCode フォームの改良
1.1301	2013.06	<ul style="list-style-type: none">● 各 Step フォームをコントロールとして表示するようにした● 処理に失敗したときのファイル削除動作の変更● 変換した D L L の出力先フォルダを変更
1.1300	2013.05	<ul style="list-style-type: none">● 変換した D L L の出力先を固定化した● F Z D L L Version 1.1300 にアプリケーション I D を移動するように変更
1.1201	2012.05	<ul style="list-style-type: none">● F Z D L L Version 1.1201 にアプリケーション I D を移動するように変更
1.1102	2012.02	<ul style="list-style-type: none">● 初期リリース

用語集 (2014.01)

アプリケーション I D

アプリケーションを識別するための番号。

プロダクトキー

ユーザーに入力させるキーです。

認証コード

プロダクトキーが正しかったことを示す証明コード。

マジックコード

アプリケーション I D を FZ DLL に内包した場合に、読出しに必要な番号。

マニファクチャラ名

アプリケーション I D を FZ DLL に内包した場合に、読出しに必要な生産者名。

プロダクト名

アプリケーション I D を FZ DLL に内包した場合に、読出しに必要なアプリケーション名。

時間値

アプリケーションをインストールしたときの時間。通常の時間値と異なるフォーマット。

ユーザー名

トランクファイルに時間値、認証コードを書き込む際必要。認証コードを生成するためにも必要。

トランク (Trunk) ファイル

時間値、認証コード、ユーザー名を書き込むファイル。

V S I

Visual Studio Installer。Visual Studio に、付属のインストーラー。

開発環境 (2024.01)

動作確認環境

PC	CPU	RAM	OS
ASrock X300	Ryzen 3200G	8.0G	Windows 10
ASUS M5A88-M EVO	Athlon II 615e	4.0G	Windows 8.1/8/7
ASUS A7S8X-MX	Sempron 3000+	1.0G	Windows Vista

開発環境

Windows 10

Windows8.1

Windows8

Windows7 (SP1)

Windows Vista (SP2)

Visual Studio 2015 Community (Update3)

Visual Studio 2008 (SP1)