暗号 OCX

WILL

株式会社ウィル

白紙ページ

白紙ページ

目 次

はじめに

白紙ページ

商品に含まれるもの

1．CD-ROM
- Willware.exe
- Cryptdll.exe
  （暗号 DLL 専用・実行環境用セットアップキット）
- readme.txt

2．フロッピーディスク
- レジストリファイル
- readme.txt

3．使用許諾契約書

4．マニュアル

動作環境について

■対応 OS

暗号 OCX は、以下に示す OS で動作確認を行っております。

Microsoft Windows 95、Microsoft Windows 98、
Microsoft WindowsNT 4.0、Microsoft Windows 2000、
Microsoft Windows XP（暗号 OCX Ver 1.02 より対応）、
Microsoft Windows 2003

■開発に必要なソフトウェア

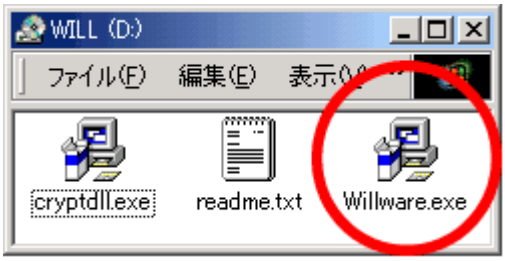
暗号 OCX をご使用いただくには、以下のいずれかのソフトウェアが必要です。

Microsoft Visual Basic Ver 5.0
Microsoft Visual Basic Ver 6.0
Microsoft Office 2000（Access、Excel）

暗号 OCX は、Microsoft Visual C++ Ver5.0 で作成しています。サンプルは、Microsoft Visual Basic Ver 5.0 で作成しています。
※ 本製品は日本語環境のみの対応となります。

インストール

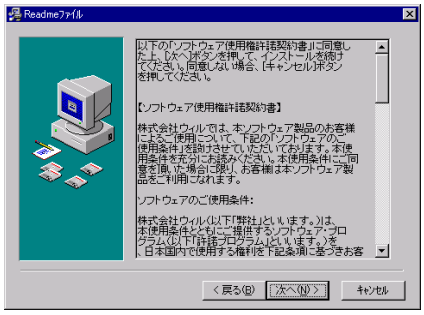
製品の CD-ROM に含まれているセットアップキット（Willware.exe）をダブルクリックします。



画面にしたがって、インストールを進めて下さい。

1. インストールを始めます。「次へ」をクリックして下さい。

2. 使用許諾契約書です。内容に同意される場合は「次へ」をクリックして下さい。



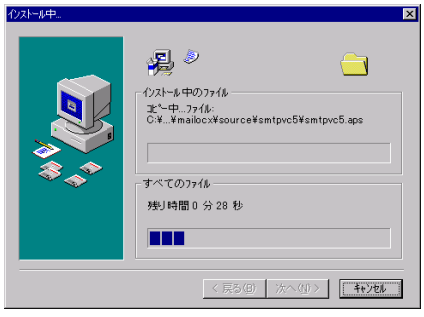
3. インストール先のフォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。別のフォルダを指定したい場合は「参照」をクリックし、フォルダを指定して下さい。

4. インストール中に置換されるファイルのバックアップを作成できます。そのバックアップファイルの保存先フォルダを指定します。初期設定でよろしければ「次へ」をクリックして下さい。

5. WILLWARE Components を登録するスタートメニュー又はプログラムマネージャのグループフォルダを指定します。初期設定では、新規に「WILLWARE Components」の名前でフォルダを作成します。特に指定する必要がなければ、初期設定をお勧めします。

6. プログラムのコピーを開始します。「次へ」をクリックして下さい。

7. プログラムのコピーをしています。中断する場合は、「キャンセル」をクリックして下さい。

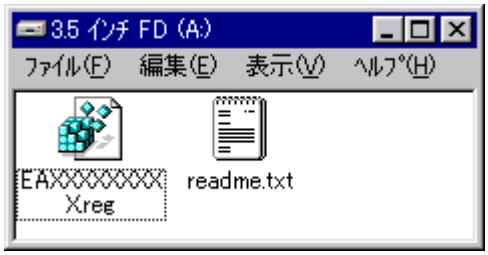


8. インストールが完了しました。「完了」をクリックし、インストールを終了して下さい

ライセンスの登録

■レジストリファイルから登録する

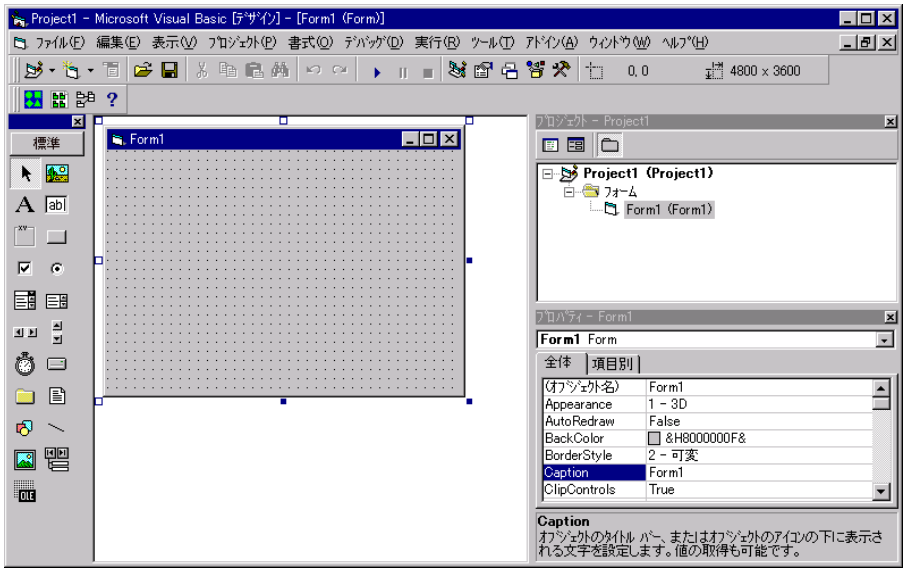
ライセンスを登録します。製品に含まれているフロッピーディスクのレジストリファイル（EAXXXXXXXXX.reg）をダブルクリックして下さい。（「XXXXXXXXX」は、任意の数字がファイル名として付けられています。）



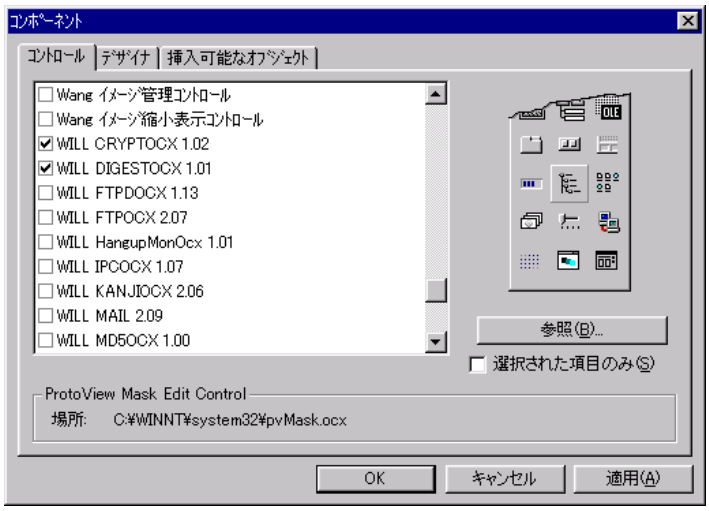
以下のメッセージボックスが表示され、ライセンスがレジストリに登録されます。

■手動で登録する

あらかじめ電子メールで通知しているライセンス情報を利用してライセンスを登録する等、レジストリファイルを利用しない場合は、VisualBasic 起動後に新規プロジェクトを選択し以下のデザイン画面を開きます。



ツールバーの「プロジェクト」から、「コンポーネント」を選択し、「コンポーネント」画面を開きます。次にコントロールタブの一覧から CRYPTOCX 及び DIGESTOCX を選択して「OK」をクリックすると、OCX がツールボックスに追加され、アイコンが表示されます。

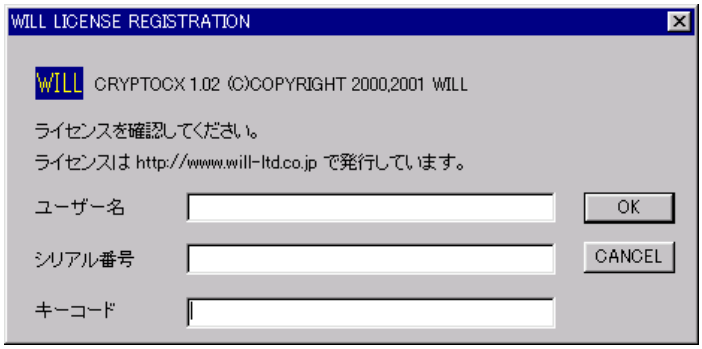


ツールボックスに追加された OCX を選択し、フォームにアイコンを貼り付けると、以下の「WILL LICENSE REGISTRATION」画面が表示されます。ここで、ユーザー名、シリアル番号、キーコードをそれぞれ入力してライセンス登録を行います。

■トライアルライセンスから正規ライセンスへの移行

既にトライアルライセンスが登録されている場合には、デザイン画面にある OCX のプロパティで「バージョン情報」をクリックして下さい。

「WILL LICENSE REGISTRATION」画面が表示されますので、ここで正規ライセンスを入力して下さい。



■ライセンス入力時のご注意

※ライセンスが入力できない！？

入力したライセンスにスペースが含まれていないか確認して下さい。(ライセンスに、スペースは使用していません。)

※登録したライセンスを認識しない！？

ライセンスを登録しても、オブジェクトが新規ライセンスを認識していない場合は、暗号 OCX のアイコンを少し動かして下さい。この作業により、オブジェクトにライセンスが記憶されます。

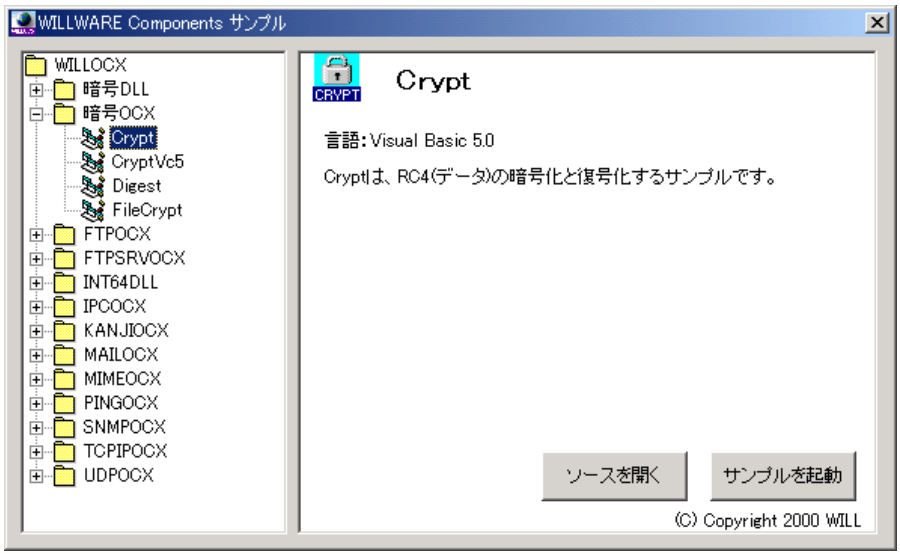
※トライアルライセンスで作成したアプリケーションはどうする！？

既にトライアルライセンスで作成したアプリケーションは、正規ライセンスを登録した後、再コンパイルする必要があります。

サンプルを見る

インストールが完了すると、スタートメニューに「WILLWALE Components」が追加されます。

「WILLWALE Components」の「サンプル」を起動すると「WILLWARE Components サンプル」画面が表示されます。サンプルの起動、またはそれぞれのソースを開くことができます。但し、ソースを開くにはライセンスが必要です。トライアルライセンス又は、正規ライセンスを登録してご利用下さい。（ライセンスの登録方法は前項の「ライセンスの登録」をご覧下さい。）



サポートについて（無償）

サポートは基本的に電子メールで受け付けております。サポートは無償でご利用いただけます。

■お問い合わせの前に

サポート作業を円滑に行うために、お問い合わせの際には以下の情報をご用意下さい。

1．製品名及びバージョン
2．開発環境（OS の種類及びバージョン、サービスパッケージの種類）
3．開発ツール及びバージョン
4．サーバの種類
5．問題点
   (1) エラー内容又は、エラー状況のハードコピー
   (2) 問題点となる部分のサンプルソースコード。

■FAQ

弊社ホームページの「サポート」のページで、キーワードを入力して FAQ を検索できます。休業日などサポートの対応が遅れる場合もありますので、まずはこちらをご確認下さい。

■お問合せ先

info@will-ltd.co.jp

バージョンアップについて（無償）

製品のバージョンアップは、すべて無償です。

■バージョンアップ情報の入手方法

バーションアップの情報は、弊社ホームページの新着情報で通知し、各商品のページの更新履歴で更新内容を掲示致します。

■最新バージョンの入手方法

最新バージョンのプログラムは、弊社ホームページ（http://www.will-ltd.co.jp/）のダウンロードのページよりダウンロードすることが出来ます。ダウンロードするファイルは、以下のバージョンアップの目的により異なりますのでご注意下さい。

● **WILLWARE Components（全製品用）セットアップキットを利用してバージョンアップ**
ファイル名　：「Willware.exe」

WILLWARE Components（全製品用）セットアップキットは全ての製品をインストールするためのものです。そのため本製品以外の製品及びサンプル、マニュアルも同時にバージョンアップされます。

● **各コンポーネント毎のセットアップキットを利用してバージョンアップ**
ファイル名　：「〇〇〇ocx.exe」

各コンポーネントのファイル(ocx、dll)及び、依存ファイルのみバージョンアップされます。サンプル及びマニュアルはバージョンアップされませんのでご注意下さい。

■バージョンアップをする前に

各セットアップキットを利用してバーションアップをする前に、以下のことにご注意下さい。

● **WILLWARE Components（全製品用）セットアップキット**を利用してバージョンアップする場合は、古いバージョンをアンインストールしてから、最新バージョンをインストールすることをお勧めいたします。
※ アンインストールの方法は、スタートメニューから「設定」→「コントロールパネル」→「アプリケーションの追加と削除」の画面で、「WILLWARE Components」を選択し、画面の指示に従って行って下さい。

● **各コンポーネント毎のセットアップキット**を利用してバージョンアップする場合は、最新バージョンをそのままインストールして下さい。古いファイルは上書きされます。
※ 弊社製品を複数ご利用いただいている場合、いずれか１つをバーションアップしても他の製品に影響はありません。

■バージョンアップの方法

セットアップキットをダブルクリックし、画面の指示に従ってインストールを進めて下さい。

再配布について

■作成したアプリケーションの配布時

暗号 OCX を利用して作成したアプリケーションの配布時のランタイムライセンスはフリーです。但し、開発ライセンスの配布はできません。

■再配布時に必要な配布可能ファイル

暗号 OCX を利用して作成したアプリケーションを配布する場合には、以下のファイルを添付する必要があります。( )内は推奨バージョンです。

- CRYPT.OCX
- DIGEST.OCX
- MFC42.DLL （Ver 4.21.7022）
- MFC42LOC.DLL （Ver 4.21.7022）
- MSVCRT.DLL （Ver 5.00.7022）
- OLEPRO32.DLL （Ver 5.0.4118）
- OLEAUT32.DLL （Ver 2.20.4118）

※ セットアップウィザードを使用する場合

暗号 OCX をインストールすると、自動的に OCX の依存ファイルが以下のディレクトリにインストールされます。

C:¥Windows¥system （Windows95, Windows98 の場合）
C:¥WINNT¥system32 (WindowsNT4.0, Windows2000, Windows2003 の場合）
C:¥Windows¥system32 (WindowsXP の場合）

セットアップウィザードを実行すると自動的にアプリケーション配布時に必要な OCX（内部で利用している OCX）と、DLL ファイルが Setup.lst ファイルに追加されます。

■著作権

- 暗号 OCX およびこれに付随するマニュアルの著作権は株式会社ウィル（横浜市保土ヶ谷区）にあります。
- 本ソフトウエアおよびマニュアルを運用した結果については、当社は一切責任を負いません。
- 本ソフトウエアの仕様またはマニュアルに記載されている事項は予告無く変更することがあります。
- マニュアルなどに記載されている会社名、製品名は、各社の商標および登録商標です。
- 暗号 OCX を利用するアプリケーションは暗号 OCX の著作権表示を行わなければなりません。Copyright プロパティに暗号 OCX の著作権を示す文字列があります。アプリケーションまたはドキュメントのいずれかにこの文字列を表示して、暗号 OCX

を使用していることを示してください。

製 品 概 要

白紙ページ

特 徴

暗号 OCX は、データの暗号と復号、およびデータの要約値の計算を行なうための 32 ビット Active X コントロールです。

CRYPTOCX および DIGESTOCX の 2 つのコントロールで構成されます。これらのコントロールは、Microsoft RSA Base Provider によってサポートされるアルゴリズムを用いて暗号化、復号化、要約を行ないます。

CRYPTOCX

CRYPTOCX は、データの暗号と復号を行なうコントロールです。CRYPTOCX で行なうことのできる暗号方式は、RC4(Revest Cipher 4)および DES と呼ばれるものです。RC4,DES の鍵はともに秘密鍵方式で、1 つの鍵を暗号時と復号時に利用します。RC4 の鍵の強度は、40 ビット又は、128 ビットですべての暗号プロバイダで利用可能です。DES の鍵強度は 56 ビット、112 ビット、168 ビットの 3 種類がありますが、56 ビット以外の暗号強度は利用する暗号プロバイダが Strong または Enhanced の場合に限られます（Microsoft RSA Base Provider では動作しません）。

RC4 は、ストリームタイプの暗号方式で、1 バイト単位で暗号化が可能です。データを任意の大きさに区切って暗号化できるのでデータの取り扱いが非常に簡単です。通信中のデータの暗号化、復号化に非常に向くもので、弊社製品である TCPIPOCX、FTPOCX、MAILOCX などと組み合わせると効果的です。

DES は、ブロック暗号方式で 8 バイト単位で暗号を施していきます。UNIX のパスワードの暗号として有名です。56 ビットの暗号は強度として問題があるといわれていますが、112 ビットおよび 168 ビットのものはトリプル DES と呼ばれ、充分な強度をもっているとされています。

【DES 動作環境】

OS	InternetExplorer	56 ビット	112 ビット	168 ビット
Widows 95	Ver 5.5 以上	○	×	×
Widows 98	Ver 5.5 以上	○	×	×
Widows NT 4.0	Ver 5.5 以上	○	×	×
Widows 2000	――――――	○	○	○
Widows XP	――――――	○	○	○
Widows 2003	――――――	○	○	○

DIGESTOCX

DIGESTOCX は、データの要約を行なうコントロールです。データの要約とは、いわゆるチェックサムのようなもので、データが改竄（かいざん）されたかどうかを確認するために利用されるものです。

DIGESTOCX は、MD5(Message Digest 5)と SHA(Secure Hash Algorithm)の 2 種類の方式を選択できます。MD5 は 16 バイト長の要約データを生成し、SHA は 20 バイトの要約データを生成します。

CRYPTOCX コントロール

- ▼ プログラミング概要
- ▼ プロパティ
- ▼ メソッド

白紙ページ

▼ プログラミング概要

白紙ページ

暗号化

暗号化する前のデータを平文と呼びます。
CRYPTOCX では平文を文字列変数に格納しておきます。

```
Dim X$

X = "平文"
```

Unicode ではなく ANSI コードで暗号化したければ、

```
X = StrConv("平文" ,vbFromUnicode)
```

とします。
VB の文字列変数には、文字列だけでなくバイナリデータも格納できます。バイナリデータは次のようにして文字列変数に格納します。

```
Dim X$, B() As Byte,I%

ReDim B(99) '100 バイトのバイナリデータ
For I=0 To 99     'B(0)---(B(99)にデータを格納する
    B(I) = I
Next
X = B             'B()の内容がそのまま X にコピーされる
                  'X は UNICODE データではないので
                  表示などはできない
```

平文データが用意できたらこれを暗号化します。
OpenRC4 メソッドで、パスワードおよびパスワードから抽出する暗号キー方式および初期データ（SALT）を生成するかどうかを指示します。ここでは、MD5 を用いて暗号キーを抽出し、さらに初期データ（SALT）を生成することを指示します。なお、パスワードは、Unicode 文字列で指示してください。

```
Crypt1.OpenRC4 "Password", RC4HashMD5, True
```

これで準備ができました。後はデータを暗号化するだけです。
暗号化を行なうには、Encrypt メソッドを用います。引数に平文を含む文字列変数を渡します。CRYPTOCX はこの変数の内容をバイナリ値として 1 バイトづつ暗号化し、同じ変数に暗号化した値を書き込みます。暗号化前と暗号化後の変数のサイズは同じになります。データが大量にある場合は、Encrypt メソッドを複数回呼び出します。この際、データは任

意の大きさで区切ることができます。

Crypt1.Encrypt　X '暗号化の結果はものと変数に戻る

暗号化が終わると、Close メソッドを呼び出してください。

Crypt1.Close

復号化

暗号化されたデータは、文字列変数に格納しておきます。

OpenRC4 メソッドで、パスワードおよびパスワードから抽出する暗号キー方式および初期データ(SALT)を生成するかどうかを指示します。
暗号化で用いた条件と同じ条件を指定してください。
ここでは、MD5 を用いて暗号キーを抽出し、さらに初期データ(SALT)を生成することを指示します。なお、パスワードは、Uncode 文字列で指示してください。

```
Crypt1.OpenRC4 "Password", RC4HashMD5, True
```

ここまでは、暗号化と同じです。
これで準備ができました。後はデータを復号するだけです。
復号を行なうには、Decrypt メソッドを用います。引数に暗号文を含む文字列変数を渡します。CRYPTOCX はこの変数の内容をバイナリ値として 1 バイトづつ復号し、同じ変数に復号した値を書き込みます。復号前復号後の変数のサイズは同じになります。データが大量にある場合は、Decrypt メソッドを複数回呼び出します。この際、データは任意の大きさで区切ることができます。バイト単位で処理されるので、暗号化の際の区切る大きさが異なってもかまいません。

```
Crypt1.Decrypt   X '復号化の結果はものと変数に戻る
```

復号化が終わると、Close メソッドを呼び出してください。

```
Crypt1.Close
```

白紙ページ

▼ プロパティ

白紙ページ

Copyright プロパティ

■機　能

著作権およびバージョン情報を表示します。

■構　文

Object.Copyright

Copyright プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。

■データ型

文字列(String)

Licensee プロパティ

■機　能

CRYPTOCX のライセンス情報です。バージョン情報画面からいつでもライセンスを切り替えることが可能です。
トライアルライセンスから正規ライセンスに切り替えた場合、このプロパティで正規ライセンスが組み込まれたことを確認して、再度アプリケーションを作成してください。

■構　文

Object.Licensee
Licensee プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。

■データ型

文字列(String)。

▼ メソッド

白紙ページ

OpenRC4 メソッド

■機　能

RC4 アルゴリズムで暗号あるいは復号処理を開始します。

■構　文

Object.OpenRC4(PASSWORD As String, HashAlgorithm As RC4HashALG, CreateSalt As Boolean)
OpenRC4 メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
PASSWORD	パスワードを含む Unicode 文字列(String)です。
HashAlgorithm	Hash アルゴリズムを指定する、長整数(Long)です。次の設定値を参照してください。
CreateSalt	暗号強度を指定する、ブール型(Boolean)です。次の設定値を参照してください。

■設定値

HashAlgorithm の設定値は次の通りです。		
(定　数)	(値)	(内　容)
RC4HashMD5	0	MD5 アルゴリズム。
RC4HashSHA	1	SHA アルゴリズム。

CreateSalt の設定値は次の通りです。	
(値)	(内　容)
True	128 ビット暗号。
False	40 ビット暗号。

■戻り値

なし

■解　説

このメソッドにより暗号化処理のための準備を行ないます。
このメソッドが成功すると暗号化および復号化が可能になります。RC4 暗号化では、パ

スワードから生成される Hash 値の先頭 5 バイト(40 ビット)がキーとして利用されます。CreateSalt を True にすると Hash 値の先頭 16 バイト(128 ビット)をキーとするようになります。

OpenDES メソッド

■機　能

DES アルゴリズムで暗号あるいは復号処理を開始します。

■構　文

Object.OpenDES(PASSWORD As String, HashAlgorithm As RC4HashALG, bits As DESBITS)

OpenDES メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内 容)
Object	CRYPTOCX オブジェクトです。
PASSWORD	パスワードを含む Unicode 文字列です。
HashAlgorithm	Hash アルゴリズムを指定する、長整数(Long)です。次の設定値を参照してください。
bits	フラグです。次の設定値を参照してください。

■設定値

HashAlgorithm の設定値は次の通りです。

(定 数)	(値)	(内 容)
RC4HashMD5	0	MD5 アルゴリズム。
RC4HashSHA	1	SHA1 アルゴリズム。

(注意)
暗号化と復号化で、異なるハッシュアルゴリズムを指定すると正しい結果が得られません。

bits の設定値は次の通りです。

(定 数)	(値)	(内 容)
DES56	0	DES56 ビット CBC モード
DES112	1	トリプル DES112 ビット CBC モード
DES168	2	トリプル DES168 ビット CBC モード
DES56ECB	4	DES56 ビット ECB モード
DES112ECB	5	トリプル DES112 ビット ECB モード
DES168ECB	6	トリプル DES168 ビット ECB モード

(注意)
トリプル DES を利用できるのは、Strong または Enhanced の Cryptographic Provider の場合だけです。トリプル DES が使えるかどうかは、GetCSPName メソッド使って確認してください。

■戻り値

なし

■解　説

DES はブロック暗号です。

Encrypt メソッドおよび Decrypt メソッドにデータを分割して渡す場合は、8 の倍数のバイト数の大きさのブロックに分割する必要があります。また、途中のブロックの場合には Final を False に、最終のブロックには Final を True にする必要があります。また、暗号化されたデータは、最大で 8 バイトのパッディングデータが付加されます。

パッディングデータを付加したくない場合は、元のデータを 8 の倍数の大きさにします。このようにすると、最終ブロックでも Final を False にすることができます。

ブロックに分割しない場合は、データを最終ブロックとして扱います。データの大きさが 8 の倍数でなければ Final を True にします。8 の倍数であれば Final を True または False に指定することができます。

■データを分割して暗号化する例

```
Dim x$, a$ ,b$
Crypt1.OpenDES "パスワード", RC4HashMD5, DES112
x = "暗号化するデータ"
a = "" '暗号化されたデータを蓄積する
Do While LenB(x) > 0
    b = LeftB$(x, 8) 'ブロックは 8 の倍数バイト
    x = MidB$(x, 9)
    If LenB(x) = 0 Then
        Crypt1.Encrypt b, 1 '最終ブロック
    Else
        Crypt1.Encrypt b, 0 '途中のブロック
    End If
    a = a & b
Loop
Crypt1.Close
```

■バイナリデータを暗号化する方法

バイト配列 B() As Byte にバイナリデータが格納されている場合、次のように文字変数にバイナリのまま格納してから暗号化します。

```
Dim B() As Byte, S As String, i&
ReDim B(4) '5 バイト
For i = 0 To UBound(B)
    B(i) = i
Next
S = B
Debug.Print "S(暗号前)-->" & Crypt1.STR2HEX(S)
Crypt1.OpenDES "ogawa", RC4HashMD5, DES112
Crypt1.Encrypt S, True
Crypt1.Close
B = S
Debug.Print "S(暗号後)-->" & Crypt1.STR2HEX(S)
For i = 0 To UBound(B)
    Debug.Print "B(" & i & ")-->" & Right$("0" & Hex$(B(i)), 2)
Next
```

■参　考

ECB　:　Electric CodeBook
CBC　:　Cipher Block Chaining

Encrypt メソッド

■機　能

暗号処理を行ないます。

■構　文

Object.Encrypt(Data As String, [Final])
Encrypt メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
Data	文字列変数。暗号化前のデータを格納します。

■戻り値

なし

■解　説

このメソッドによりデータを暗号化します。
暗号化した結果は Data 変数に書き戻します。

Decrypt メソッド

■機　能

復号処理を行ないます。

■構　文

Object.Decrypt(Data As String, [Final])
Decrypt メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
Data	文字列変数。暗号前のデータを格納します。

■戻り値

なし

■解　説

このメソッドによりデータを復号します。
復号した結果は Data 変数に書き戻します。

Close メソッド

■機　能

暗号化処理を終了します。

■構　文

Object.Close()
Close メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。

■戻り値

なし

■解　説

このメソッドにより暗号化処理のための資源を解放します。
このメソッドはエラーを発生することはありません。

STR2HEX メソッド

■機　能

バイナリ文字列の内容を Unicode で表示可能な 16 進文字列に変換します。

■構　文

X = Object.STR2HEX(Data As String)
STR2HEX メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
S	バイナリ文字列(String)
X	Unicode 文字列(String)。VB で表示可能。

■戻り値

文字列(String)。

■解　説

暗号化したデータを確認するために、文字列変数に格納されているバイナリデータを VB などで表示可能になるように、Unicode 文字列に変換します。すべてのデータを表示できるよう、16 進表記します。

HEX2STR メソッド

■機　能

16 進表記された Unicode 文字列からバイナリ文字列を生成します。

■構　文

S = Object.HEX2STR(Str As String)
HEX2STR メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
X	Unicode 文字列(String)。16 進数。偶数文字数です。
S	バイナリ文字列(String)です。

■戻り値

文字列(String)

■解　説

16 進表記された Unicode 文字列をバイナリ文字列に変換する。
Unicode 文字列は、"0"から"9"と"A"から"F"までの文字だけで形成され、文字数は偶数でなければなりません。

LastError メソッド

■機　能

API で発生したエラーコードです。

■構　文

ErrCode = Object.LastError()
LastError メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
ErrCode	長整数のエラーコードです。

■戻り値

長整数(Long)。

(値)	(内　容)
0	正常
上記以外	エラー

■解　説

Err.Number が 29806 であるエラーが発生した場合に詳細なエラーコードを知ることができます。

LastErrorMessage メソッド

■機　能

API で発生したエラーメッセージです。

■構　文

ErrMsg = Object.LastErrorMessage()
LastErrorMessage メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	CRYPTOCX オブジェクトです。
ErrMsg	文字列のエラーメッセージです。

■戻り値

文字列(String)。

(値)	(内　容)
空文字	エラーメッセージなし
上記以外	エラーメッセージあり

■解　説

Err.Number が 29806 であるエラーが発生した場合に詳細なエラーメッセージを知ることができます。

DIGESTOCX コントロール

- ▼ プログラミング概要
- ▼ プロパティ
- ▼ メソッド

白紙ページ

▼ プログラミング概要

白紙ページ

要 約

データの要約を得るには、データを文字列変数に格納して用意してください。

```
Dim X$

X = "データ"
```

データは Unicode 文字列だけでなくバイナリ文字列でもかまいません。

データの準備ができたら、要約を開始します。要約を開始するには、Open メソッドを利用します。Open メソッドには 1 つの引数があり、要約のためのアルゴリズムを指定します。アルゴリズムには MD5 と SHA の 2 種類があります。MD5 は 16 バイトの要約を、SHA は 20 バイトの要約を生成します。ここでは、MD5 での要約を指示します。

```
Digest1.Open HashMD5
```

つぎに、Add メソッドを用いてデータを要約していきます。

データを分割して Add メソッドを呼び出すことも可能です。このとき分割するデータのサイズは任意です。たとえば、100 バイトのデータを 10 バイトづつ 10 回にわけて Add しても 50 バイトづつ 2 回に別けても 100 バイト 1 回でも同じ要約結果が選られます。

```
Digest1.Add X
```

要約結果は、DIGESTOCX 内部に保存されています。これを取り出すには GetHashValue メソッドを用います。GetHashValue メソッドで取り出した値は、バイナリ文字列で、MD5 アルゴリズムでは 16 バイト、SHA アルゴリズムでは、20 バイトの長さがあります。

このメソッドを1度呼び出すと API エラーが発生し Add メソッドは使用できなくなります。

```
d = Digest1.GetHashValue()
```

必要な要約データをとりだしたら、Close します。

```
Digest1.Close
```

要約データはバイナリ文字列です。Unicode 文字列ではないので、VB でそのまま表示することはできません。DIGESTOCX には、バイナリ文字列を Unicode により 16 進表示す

るためのメソッド STR2HEX が用意されています。

```
Debug.Print "MD5 DIGEST=" & Digest1.Str2Hex(d)
```

により、要約データの内容を確認することができます。

▼ プロパティ

白紙ページ

Copyright プロパティ

■機　能

著作権およびバージョンを表示します。

■構　文

Object.Copyright

Copyright プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。

■データ型

文字列(String)。

Licensee プロパティ

■機　能

　DIGESTOCX のライセンス情報です。バージョン情報画面からいつでもライセンスを切り替えることが可能です。
　トライアルライセンスから正規ライセンスに切り替えた場合、このプロパティで正規ライセンスが組み込まれたことを確認して、再度アプリケーションを作成してください。

■構　文

Object.Licensee
Licensee プロパティの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。

■データ型

文字列(String)。

▼ メソッド

白紙ページ

Open メソッド

■機　能

要約処理を開始します。

■構　文

Object.Open(Algorithm As HASHALG)
Open メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
HashAlgorithm	Hash アルゴリズムを指定する長整数(Long)です。

■設定値

HashAlgorithm の設定値は次の通りです。

(定　数)	(値)	(内　容)
HashMD5	0	MD5 アルゴリズム
HashSHA	1	SHA アルゴリズム

■戻り値

なし

■解　説

このメソッドにより要約処理のための準備を行ないます。

Add メソッド

■機　能

要約処理を行ないます。

■構　文

Object.Add(Data As String)
Add メソッドの構文の指定項目は次の通りです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
Data	文字列変数。要約データを格納します。

■戻り値

なし

■解　説

このメソッドによりデータを要約します。
要約した値は、DIGESTOCX 内部で保持します。
データが多い場合は任意に分割して Add メソッドを呼び出してください。一度に呼び出しても分割して呼び出しても要約の値は同じです。

GetHashValue メソッド

■機　能

要約データを取り出します。

■構　文

S = Object.GetHashValue()
GetHashValue メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
S	文字列の要約データです。

■戻り値

文字列(String)

■解　説

このメソッドにより要約データを取り出します。
Close メソッドを呼び出す前にこのメソッドで要約データを取り出してください。
MD5 では、16 バイト、SHA では 20 バイトの要約データを受け取ります。
要約データが同じ場合は、元のデータは同じであると推測できます。
このメソッドを1度呼び出すと API エラーが発生し Add メソッドは使用できなくなります。

Close メソッド

■機　能

要約処理を終了します。

■構　文

Object.Close()
Close メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。

■戻り値

なし。

■解　説

このメソッドにより要約処理のための資源を解放します。
このメソッドはエラーを発生することはありません。

STR2HEX メソッド

■機　能

バイナリ文字列の内容を Unicode で表示可能な 16 進文字列に変換します。

■構　文

X = Object.Str2Hex(Data As String)
Str2Hex メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
S	バイナリ文字列(String)
X	Unicode 文字列(String)。VB で表示可能。

■戻り値

文字列(String)。

■解　説

要約データを確認するために、文字列変数に格納されているバイナリデータを VB などで表示可能になるように、Unicode 文字列に変換します。すべてのデータを表示できるよう、16 進表記します。

LastError メソッド

■機　能

API で発生したエラーコードです。

■構　文

ErrCode = Object.LastError()
LastErro メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
ErrCode	長整数のエラーコードです。

■戻り値

長整数(Long)

(値)	(内　容)
0	正常
上記以外	エラー

■解　説

Err.Number が 29806 であるエラーが発生した場合に詳細なエラーコードを知ることができます。

LastErrorMessage メソッド

■機　能

API で発生したエラーメッセージです。

■構　文

ErrMsg = Object.LastErrorMessage()
LastErrorMessage メソッドの構文の指定項目は次のとおりです。

(指定項目)	(内　容)
Object	DIGESTOCX オブジェクトです。
ErrMsg	文字列のエラーメッセージです。

■戻り値

文字列(String)

(値)	(内　容)
空文字	エラーメッセージなし
上記以外	エラーメッセージあり

■解　説

Err.Number が 29806 であるエラーが発生した場合に詳細なエラーメッセージを知ることができます。

白紙ページ

エラーコード

白紙ページ

発生しうるエラーコード

メソッドまたはプロパティ	発生しうるトラップ可能なエラーコード
<CRYPTOCX>	
Encrypt	29801, 29806
Decrypt	29801, 29806
HEX2STR	29803, 29804
OpenRC4	29802, 29806, 29807
<DIGESTOCX>	
Open	29802, 29804, 29806, 29807
GetHashValue	29801, 29803, 29803, 29806

■エラーの内容

エラーコード	内　　容
29801	オープンしていません。
29802	すでにオープンしています。
29803	メモリ不足です。
29804	引数の値が不正です。
29806	API エラーです。
29807	暗号サービスプロバイダーが利用できません。

白紙ページ

サンプル

白紙ページ

FileCrypt

（Ver1.00）

FileCrypt は、ファイルを暗号化、複合化するサンプルです。

■使い方

1. パスワードを指定して下さい。
2. 「SALT 使用」にチェックをすると暗号強度が増します。
3. 暗号化するタイプを「暗号ファイルモード」で指定して下さい。
4. 暗号化したいファイルを  にドロップして下さい。ファイルが暗号化されます。
5. 暗号化したファイルを  にドロップして下さい。ファイルが復号化されます。

Digest

（Ver1.00）

Digest は、ファイルを要約するサンプルです。

■使い方

1. 要約したいファイルをドロップして下さい。ファイルの要約値が「ダイジェスト（要約）の値」に表示されます。
2. 元のファイルが少しでも変更されると、要約値は全く違ってきます。

Crypt

（Ver1.00）

Crypt は、RC4(データ)の暗号化と復号化するサンプルです。

■使い方

1. 暗号化したい文を「平文」に入力して下さい。
2. パスワードを指定して下さい。
3. 「暗号化」ボタンを押すと。入力した文が暗号化されます。
4. 「複合化」ボタンを押すと暗号化された文が複合化されます。

CryptVc5

（Ver1.00）

Crypt は、RC4(データ)の暗号化と復号化するサンプルです。

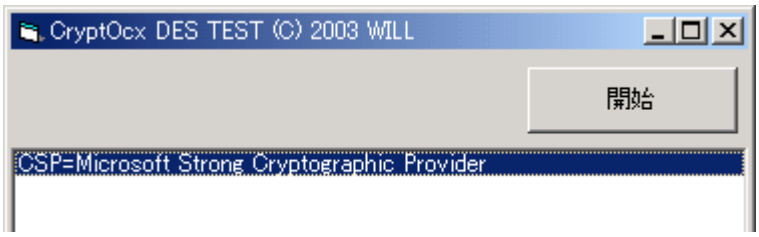
■使い方

1. 暗号したい文を「平文」に入力して下さい。
2. パスワードを指定して下さい。
3. 「暗号化」ボタンを押すと。入力した文が暗号化されます。
4. 「複合化」ボタンを押すと暗号化された文が複合化されます。

Cryptocx DES TEST

（Ver1.00）

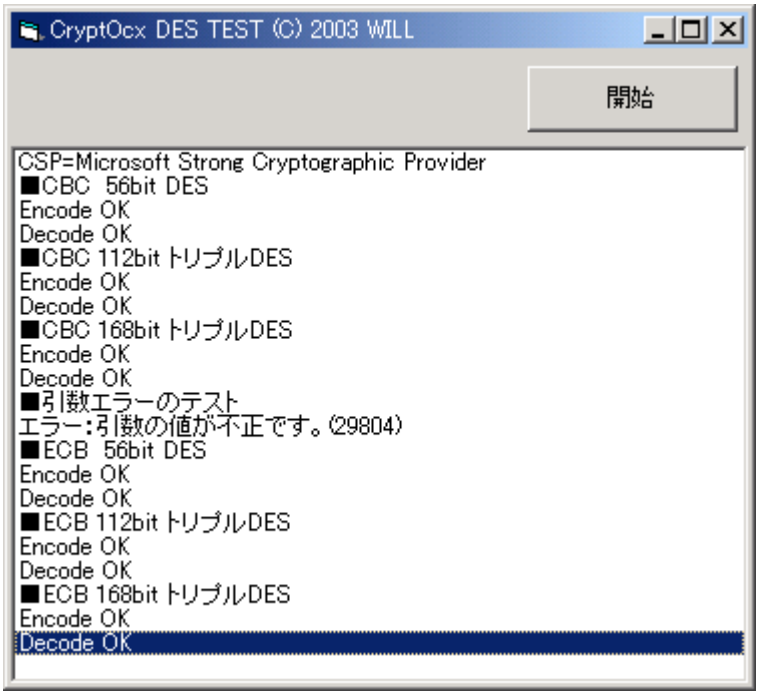
Cryptocx DES TEST は、DES(データ)の暗号化と復号化を行える環境かどうかをテストするサンプルです。



■使い方

Cryptocx DES TEST を起動して、開始ボタンを押してください。以下の様に、各暗号化及び複合化のテスト結果が表示されます。

※　引数エラーのテスト結果は、「エラー：引数の値が不正です。(29804)」という結果が正常となります。



索 引

白紙ページ

暗号 OCX マニュアル

2000 年 6 月 1 日 初版 第 1 版
2002 年 8 月 1 日 第 2 版
2003 年 8 月 28 日 第 2 版
発行所 株式会社ウィル
住所 神奈川県横浜市保土ヶ谷区西久保町 15
グランディシンヤ 302
〒240－0022
TEL：045－338－3525
FAX：045－338－3526
Mail-Address：info@will-ltd.co.jp
URL：http://www.will-ltd.co.jp /
発行者 小川 史彦