

KANJIOCX

WILL

株式会社ウィル

白紙ページ

白紙ページ

- Microsoft、Windows、Windows NT、Visual Basic、ActiveX、Office、Access、Excel は、米国 Microsoft Corporation の米国ならびに各国における登録商標です。
- その他本書に掲載されている会社名、製品名はそれぞれ各社の商標又は登録商標です。

目次

はじめに.....	3
商品に含まれるもの.....	5
動作環境について.....	5
インストール.....	6
ライセンスの登録.....	9
サンプルを見る.....	12
サポートについて(無償).....	13
バージョンアップについて(無償).....	14
再配布について.....	16
プログラミング概要.....	17
文字列の受け渡し方法.....	19
Visual Basic における漢字コード変換例.....	20
プロパティ.....	23
KanjiIn プロパティ.....	25
KanjiOut プロパティ.....	26
KatakanaConvert プロパティ.....	27
LastEncoding プロパティ.....	28
DefaultEncoding プロパティ.....	29
Copyright プロパティ.....	30
メソッド.....	31
DetectEncoding メソッド.....	33
Any2Jis メソッド.....	34
Any2Sjis メソッド.....	35
Any2Euc メソッド.....	36
Jis2SJis メソッド.....	37
Sjis2Jis メソッド.....	38
Jis2Euc メソッド.....	39
Euc2Jis メソッド.....	40
Sjis2Euc メソッド.....	41
Euc2Sjis メソッド.....	42
RepairJis メソッド.....	43
Jis2Jis メソッド.....	44
Sjis2Sjis メソッド.....	45

Euc2Euc メソッド	46
Unicode2Euc メソッド	47
Unicode2Jis メソッド	48
Unicode2Sjis メソッド	49
Euc2Unicode メソッド	50
Jis2Unicode メソッド	51
Sjis2Unicode メソッド	52
サンプル	53
KVEIWER Ver.1.02	55
索引	57

はじめに

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

KANJIOCX 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、Microsoft Windows 2003

■開発に必要なソフトウェア

KANJIOCX 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)

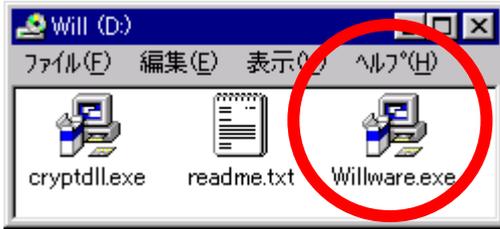
KANJIOCX は、Microsoft Visual C++ Ver 5.0 で作成しています。サンプルは、Microsoft Visual Basic Ver 5.0 で作成しています。

※ 本製品は日本語環境のみの対応となります。

はじめに

インストール

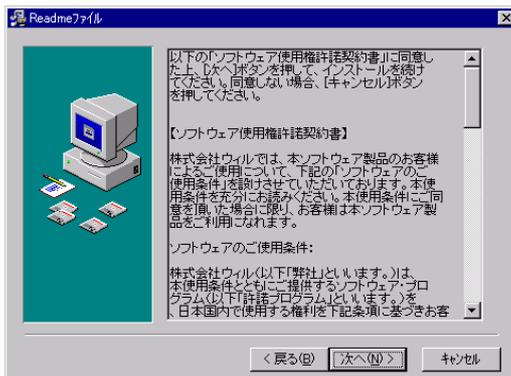
1. 製品の CD-ROM に含まれているセットアップキット (Willware.exe) をダブルクリックします。



2. 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 OCX」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。



はじめに

(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 (EKXXXXXXXXX.reg) をダブルクリックして下さい。(「XXXXXXXXXX」は、任意の数字がファイル名として付けられています。)

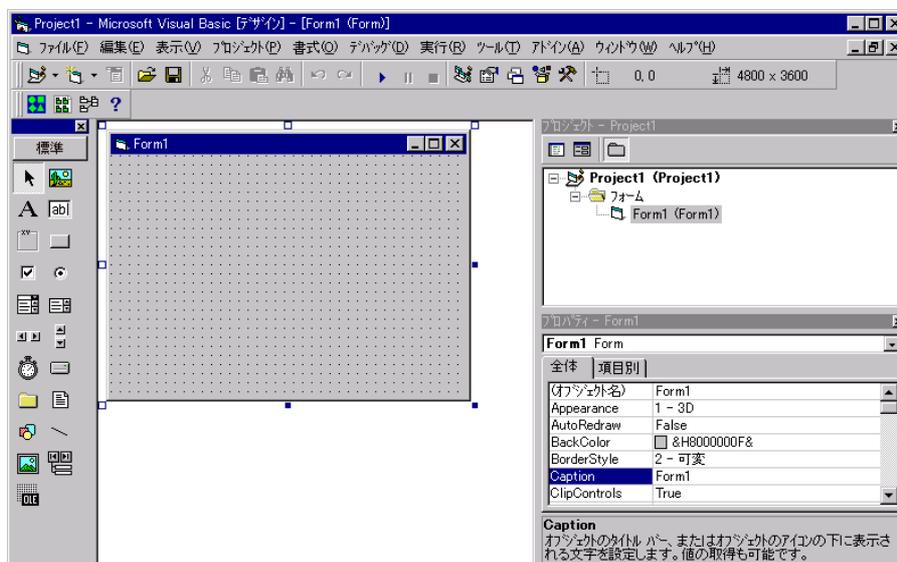


以下のメッセージボックスが表示され、ライセンスがレジストリに登録されます。



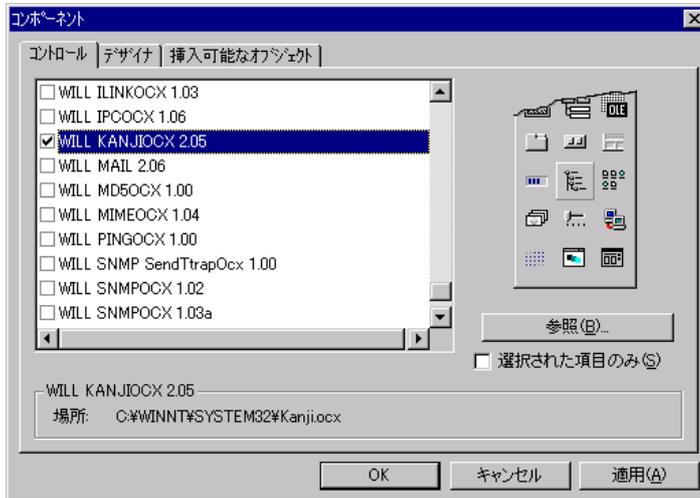
■手動で登録する

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。



はじめに

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 KANJIOCX を選択して「OK」をクリックすると、KANJIOCX がツールボックスに追加され、アイコンが表示されます。

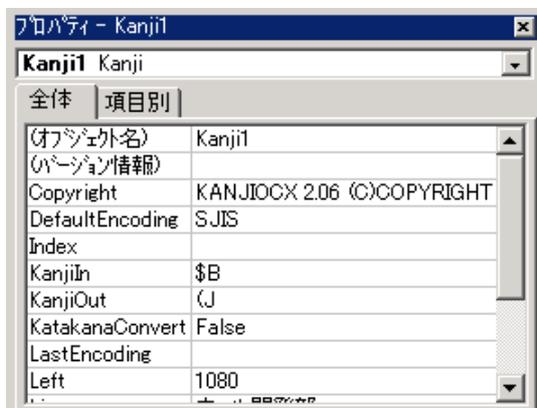


ツールボックスに追加された KANJIOCX を選択し、フォームにアイコンを貼り付けると、以下の「WILL LICENSE REGISTRATION」画面が表示されます。ここで、ユーザー名、シリアル番号、キーコードをそれぞれ入力してライセンス登録を行います。



■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 KANJIOCX 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、KANJIOCX 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。

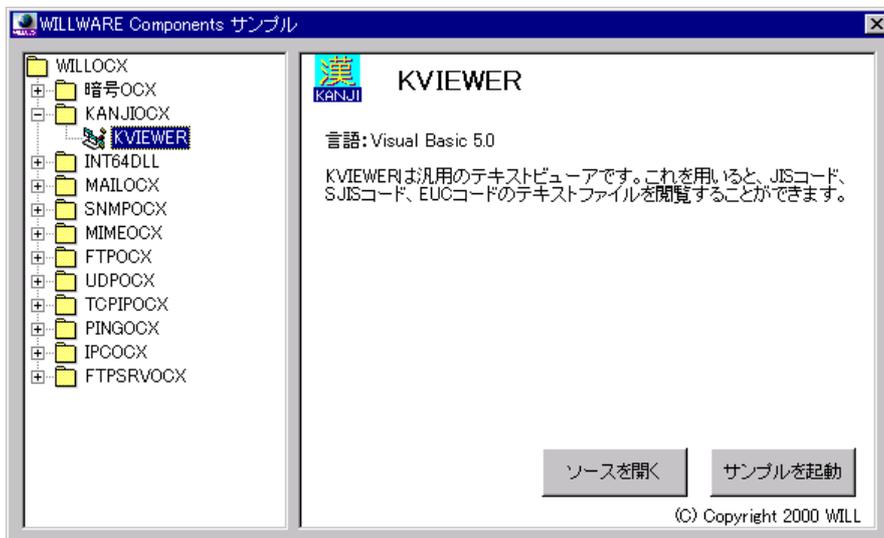
はじめに

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「WILLWARE Components」が追加されます。



「WILLWARE 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」をご覧ください。



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

バージ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ください。

- 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 : 「○○○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せん。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ください。

■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ジョンアップをする前に、以下のことにご注意ください。

● **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つをバージ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

KANJIOCX 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

KANJIOCX 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。

- ・ KANJI.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

KANJIOCX 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(Windows 95, 98, Me の場合)

C:¥WINNT¥system32(Windows NT, 2000, 2003 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

- ・ KANJIOCX 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。
- ・ 本ソフトウェ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。
- ・ 本ソフトウェ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。
- ・ KANJIOCX を利用するアプリケーションは KANJIOCX 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 KANJIOCX 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、KANJIOCX を使用していることを示してください。

プログラミング概要

白紙ページ

文字列の受け渡し方法

ファイルへのアクセスはアプリケーションレベルで行い、この OCX では文字列の受け渡しのみを行います。アプリケーションと OCX の間で文字列を受け渡す場合に、プロパティ経由では、Unicode から ANSI への自動変換が行われてしまうため、漢字コード変換 ActiveX コントロールでは、メソッドのみを提供します。文字列は必ず変数に格納して渡してください。“Text”のように文字列をそのまま渡すとエラーが発生します。メソッドの内部では、変換する文字列変数を操作します。正常に処理した場合、変数が空文字になります。変換処理を失敗した文字列は変数に残ります。

注意: VisualBasic の文字コードはすべて UNICODE を使用しますので、VisualBasic 以外のシステム(VCなど)と文字列を受け渡しする場合、文字コードを変換する必要があります。

UNICODE から ANSI へ変換

```
Text = StrConv(Text, vbFromUnicode)
```

ANSI から UNICODE へ変換

```
Text = StrConv(Text, vbUnicode)
```

Visual Basic における漢字コード変換例

■ Visual Basic でリテラル文字列を指定した場合 (Visual Basic の内部コードは Unicode)

```
Dim Sjis$, Unic$, Ansi$
```

'Unicode 文字列を指定する。

```
Unic = "テスト文字列。This is test string."
```

'Unicode 文字列を Ansi 文字列に変換する。

```
Ansi = StrConv(Unic, vbFromUnicode)
```

Windows で使用する日本語文字コードは、Sjis なので JIS に変換するには Any2Jis または、Sjis2Jis メソッドを使います。

'Sjis 文字列を Jis 文字列に変換する。

```
Jis = Kanji1.Sjis2Jis(Ansi)
```

■ JIS コードファイルから文字列を読み込んで表示

```
Dim src As String, dist As String
Dim disp As String
Dim amari As String

' 変換結果の残りを保存する変数を初期化。
amari = ""

' 表示用の TextBox を初期化する。
Text1.Text = ""
Do Until EOF(1)
    ' ファイルからデータを読み込む。
    Line Input #1, src
    ' 直前の変換結果の余りと結合する。
    If (amari <> "") Then
        src = amari & src
    End If
    ' JIS 文字列を SJIS 文字列に変換する。
    dist = Kanji1.Any2Sjis(src)
    ' 文字列を Unicode に変換する。
    disp = StrConv(dist, vbUnicode)
    ' 表示する。
    Text1.Text = Text1.Text & disp
    ' 余りを保存する。
    amari = src
Loop
```

白紙ページ

プロパティ

プロパティ

白紙ページ

KanjiIn プロパティ

■機能

1文字あたり2バイトのエスケープシーケンスを指定します。参照も可能です。エスケープ文字を除いたものを指定します。次のような指定方法です。

■構文

Object.KanjiIn [=Value]

KanjiIn 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	KANJIIOCX オブジェクトです。
Value	エスケープシーケンスのモードを指定します。デフォルト値は"\$B"です。

■設定値

Value 設定値は次のとおりです。

(値)	(説明)
\$@	<ESC> @\$@
\$B	<ESC> \$B RFC 1468 で定義されている JIS X 0208-1983 文字セットです。
\$(D	<ESC> \$(D

■データ型

文字列(String)

KanjiOut プロパティ

■機能

1文字あたり1バイトのエスケープシーケンスを指定します。参照も可能です。エスケープ文字を除いたものを指定します。次のような指定方法です。

■構文

ESC(J がエスケープシーケンスの場合。

Object.KanjiOut [=Value]

KanjiOut 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	KANJIIOCX オブジェクトです。
Value	デフォルト値は RFC 1468 で定義されている JIS X 0201-1976 ("Roman" set)文字セットを表す"J"です。

■設定値

Value 設定値は次のとおりです。

(値)	(説明)
(J)	<ESC> (J RFC 1468 で定義されている JIS X 0201-1976 ("Roman" set)文字セットです。
(B)	<ESC> (B
(I)	<ESC> (I

■データ型

文字列(String)

KatakanaConvert プロパティ

■機能

シフト JIS から Euc、Euc からシフト JIS への変換のとき、半角カタカナを全角カタカナに変換するかどうかを指定します。参照も可能です。

また、JIS コードへの変換のとき、JIS コードから他のコードへ変換するときには、このプロパティは参照されません。

■構文

Object. KatakanaConvert [=Value]

KatakanaConvert 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	KANJIIOCX オブジェクトです。
Value	変換モードを指定します。デフォルト値は FALSE です。

■設定値

Value 設定値は次のとおりです。

(値)	(説明)
TRUE	半角カタカナを全角カタカナに変換します。
FALSE	変換されません。

■解説

KatakanaConvert の影響の有無を表にすると、次のようになります。

メソッド名	影響	メソッド名	影響
Any2Jis	×	Any2Euc	△
Euc2Jis	×	Any2Sjis	△
Jis2Euc	×	Euc2Sjis	○
Jis2Sjis	×	Sjis2Euc	○
Sjis2Jis	×		

■データ型

ブール型(Boolean)

LastEncoding プロパティ

■機能

このコントロールが処理した最後の漢字コードが設定されています。参照のみ可能です。このプロパティは、DetectEncoding メソッドあるいは、Any2～メソッドを使用したときに上書きされます。

■構文

Object. LastEncoding

LastEncoding 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	KANJI OCX オブジェクトです。

■設定値

LastEncoding プロパティの設定値は以下の通りです。

(値)	(説明)
NEW_JIS	新 JIS
OLD_JIS	旧 JIS
NEC_JIS	NEC JIS
SJIS	シフト JIS
Euc	Euc
ASCII	ASCII
UNKNOWN	特定できない

■データ型

文字列(String)

DefaultEncoding プロパティ

■機能

Any2Sjis、Any2Jis、Any2Euc メソッドを使用するとき元の漢字コードを自動検出し変更を行います。しかし、シフト JIS と Euc 漢字コードが区別できない場合があります。(注: SJIS、Euc の漢字コードの中には、同じコードを割り当てる文字があります。)その時 DefaultEncoding に設定された文字コードを参照します。

■構文

Object. DefaultEncoding [=Value]

DefaultEncoding 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	KANJIIOCX オブジェクトです。
Value	変換する文字コードモードを指定します。デフォルトは“SJIS”です。

■設定値

Value 設定値は次のとおりです。

(値)	(説明)
SJIS	変換する文字をシフト JIS に指定します。
Euc	変換する文字を Euc に指定します。

■データ型

文字列(String)

Copyright プロパティ

■機 能

KANJIOCX のコピーライトが設定されています。参照のみ可能です。

■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	KANJIOCX オブジェクトです。

■データ型

文字列(String)

メソッド

メソッド

白紙ページ

DetectEncoding メソッド

■機能

日本語テキストで使用されている符号化方式を判別します。
Source の設定について、「文字列の受け渡し方法」を参照してください。

■構文

Code = Object. DetectEncoding (Source)
DetectEncoding 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	KANJIIOCX オブジェクトです。
Source	ANSI 文字列を指定します。 ※ANSI = American National Standard Institute 米国規格協会の呼称。アメリカの工業分野の標準化を行う機関。コンピュータなどに関する数万の標準規格を制定している。ISO規格をアメリカ内部で承認する機能もある。日本のJISに相当する。
Code	判別の結果を示す数値です。

■戻り値

整数(Short)。戻り値が示す値は以下の通りです。

(値)	(説明)
1	符号化方式を特定できない
2	符号化方式は新 JIS(*1)
3	符号化方式は旧 JIS(*2)
4	符号化方式は NEC 漢字 (*3)
5	符号化方式は SJIS
6	符号化方式は Euc
7	符号化方式は ASCII

*1 新 JIS とは JIS X 0208-1983 を差します。
*2 旧 JIS とは JIS X 0208-1978 を差します。
*3 ESC \$ K を検出すると「NEC 漢字」と認識します。
また、この判別結果は LastEncoding にも設定されます。

Any2Jis メソッド

■機 能

JIS コードに変換します。

Source に指定された文字列を JIS コードに変換します。JIS コードのエスケープシーケンスは KanjiIn、KanjiOut プロパティに設定されている値を使用します。

Source に指定された文字列の漢字コードは自動的に検出されます。自動検出で特定できなかった場合は、DefaultEncoding プロパティに設定されている漢字コードが使用されます。

戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Any2Jis (Source)

Any2Jis 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	KANJI OCX オブジェクトです。
Source	文字列を指定します。
Code	JIS コードに変換された文字列です。

■戻り値

文字列(String)

Any2Sjis メソッド

■機 能

シフト JIS コードに変換します。

Source に指定された文字列をシフト JIS コードに変換します。Any2Sjis、Any2Jis、Any2Euc メソッドを使用するとき元の漢字コードを自動検出し変更を行います。しかし、シフト JIS と Euc 漢字コードが区別できない場合があります。(注: SJIJ、Euc の漢字コードの中には、同じコードを割り当てる文字があります。)その時には DefaultEncoding に設定された文字コードを参照します。

戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Any2Sjis (Source)

Any2Sjis 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	KANJI OCX オブジェクトです。
Source	文字列を指定します。
Code	シフト JIS コードに変換された文字列です。

■戻り値

文字列(String)

Any2Euc メソッド

■機 能

Euc コードに変換します。

Source に指定された文字列を Euc コードに変換します。Source に指定された文字列の漢字コードは自動的に検出されます。自動検出で特定できなかった場合は、DefaultEncoding プロパティに設定されている漢字コードが使用されます。

戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Any2Euc (Source)

Any2Euc 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	KANJI OCX オブジェクトです。
Source	文字列を指定します。
Code	Euc コードに変換された文字列です。

■戻り値

文字列(String)

Jis2Sjis メソッド

■機 能

JIS コードをシフト JIS コードに変換します。

Source に指定された JIS コード文字列をシフト JIS コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Jis2Sjis (Source)

Jis2Sjis 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	KANJIOCX オブジェクトです。
Source	JIS コードの文字列を指定します。
Code	JIS コードからシフト JIS コードに変換された文字列です。

■戻り値

文字列(String)

Sjis2Jis メソッド

■機 能

シフト JIS コードを JIS コードに変換します。

Source に指定されたシフト JIS コード文字列を JIS コードに変換します。戻り値に変換された文字列が設定されます。JIS コードのエスケープシーケンスは KanjiIn プロパティ、KanjiOut プロパティに設定されている値を使用します。Source に含まれている半角カタカナは、全角カタカナに変換されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Sjis2Jis (Source)

Sjis2Jis 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	KANJI OCX オブジェクトです。
Source	シフト JIS コードの文字列を指定します。
Code	シフト JIS コードから JIS コードに変換された文字列です。

■戻り値

文字列(String)

Jis2Euc メソッド

■機 能

JIS コードを Euc コードに変換します。

Source に指定された JIS コード文字列を Euc コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Jis2Euc (Source)

Jis2Euc 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	KANJIIOCX オブジェクトです。
Source	JIS コードの文字列を指定します。
Code	JIS コードから Euc コードに変換された文字列です。

■戻り値

文字列(String)

Euc2Jis メソッド

■機 能

Euc コードを JIS コードに変換します。

Source に指定された Euc コード文字列を JIS コードに変換します。戻り値に変換された文字列が設定されます。JIS コードのエスケープシーケンスは KanjiIn プロパティ、KanjiOut プロパティに設定されている値を使用します。Source に含まれている半角カタカナは、全角カタカナに変換されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Euc2Jis (Source)

Euc2Jis 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	KANJI OCX オブジェクトです。
Source	Euc コードの文字列を指定します。
Code	Euc コードから JIS コードに変換された文字列です。

■戻り値

文字列(String)

Sjis2Euc メソッド

■機 能

シフト JIS コードを Euc コードに変換します。

Source に指定されたシフト JIS コード文字列を Euc コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Sjis2Euc (Source)

Sjis2Euc 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	KANJIIOCX オブジェクトです。
Source	シフト JIS コードの文字列を指定します。
Code	シフト JIS コードから Euc コードに変換された文字列です。

■戻り値

文字列(String)

Euc2Sjis メソッド

■機 能

Euc コードをシフト JIS コードに変換します。

Source に指定された Euc コード文字列をシフト JIS コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Euc2Sjis (Source)

Euc2Sjis 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	KANJI OCX オブジェクトです。
Source	Euc コードの文字列を指定します。
Code	Euc コードからシフト Jis コードに変換された文字列です。

■戻り値

文字列(String)

RepairJis メソッド

■機 能

破損している JIS コードを修復します。

エスケープ文字が欠落してしまった JIS コードを修復します。修復できるのは、エスケープシーケンスを構成するエスケープ文字以外の文字が完全な状態で残されていないければなりません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. RepairJis (Source)

RepairJis 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	KANJI OCX オブジェクトです。
Source	Jis コードの文字列を指定します。
Code	修復された文字列です。

■戻り値

文字列(String)

Jis2Jis メソッド

■機 能

Source から JIS コードとして完全な部分を取り出します。

Source に指定された JIS コード文字列を別のエスケープシーケンスの JIS コードに変換します。戻り値に変換された文字列が設定されます。JIS コードのエスケープシーケンスは KanjiIn プロパティ、KanjiOut プロパティに設定されている値を使用します。Source に含まれている半角カタカナは、全角カタカナに変換されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Jis2Jis (Source)

Jis2Jis 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	KANJI OCX オブジェクトです。
Source	JIS コードの文字列を指定します。
Code	JIS コードとして完全な文字列です。

■戻り値

文字列(String)

Sjis2Sjis メソッド

■機 能

Source からシフト JIS コードとして完全な部分を取り出します。

Source に指定されたシフト JIS コード文字列からシフト JIS コードとして完全な部分を取り出し、戻り値に設定します。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Sjis2Sjis (Source)

Sjis2Sjis 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	KANJIIOCX オブジェクトです。
Source	シフト JIS コードの文字列を指定します。
Code	シフト JIS コードとして完全な文字列です。

■戻り値

文字列(String)

Euc2Euc メソッド

■機 能

Source から Euc コードとして完全な部分を取り出します。

Source に指定された Euc コード文字列から Euc コードとして完全な部分を取り出し、戻り値に設定します。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Euc2Euc (Source)

Euc2Euc 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	KANJIIOCX オブジェクトです。
Source	Euc コードの文字列を指定します。
Code	Euc コードとして完全な文字列です。

■戻り値

文字列(String)

Unicode2Euc メソッド

■機 能

Unicode を Euc コードに変換します。

Source に指定された Unicode 文字列を Euc コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

```
Code = Object. Unicode2Euc (Source)
```

Unicode2Euc 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	KANJIIOCX オブジェクトです。
Source	Euc コードの文字列を指定します。
Code	Euc コードに変換された文字列です。

■戻り値

文字列(String)

Unicode2Jis メソッド

■機 能

Unicode を JIS コードに変換します。

Source に指定された Unicode 文字列を JIS コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Unicode2Jis (Source)

Unicode2Jis 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	KANJI OCX オブジェクトです。
Source	Unicode の文字列を指定します。
Code	JIS コードに変換された文字列です。

■戻り値

文字列(String)

Unicode2Sjis メソッド

■機 能

Unicode をシフト JIS コードに変換します。

Source に指定された Unicode 文字列をシフト JIS コードに変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Unicode2Sjis (Source)

Unicode2Sjis 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	KANJIIOCX オブジェクトです。
Source	Unicode の文字列を指定します。
Code	JIS コードに変換された文字列です。

■戻り値

文字列(String)

Euc2Unicode メソッド

■機 能

Euc コードを Unicode に変換します。

Source に指定された Euc コード文字列を Unicode に変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Euc2Unicode (Source)

Euc2Unicode 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	KANJI OCX オブジェクトです。
Source	Euc コードの文字列を指定します。
Code	Unicode に変換された文字列です。

■戻り値

文字列(String)

Jis2Unicode メソッド

■機 能

JIS コードを Unicode に変換します。

Source に指定された JIS コード文字列を Unicode に変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分に変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Jis2Unicode (Source)

Jis2Unicode 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	KANJIOCX オブジェクトです。
Source	JIS コードの文字列を指定します。
Code	Unicode に変換された文字列です。

■戻り値

文字列(String)

Sjis2Unicode メソッド

■機 能

シフト JIS コードを Unicode に変換します。

Source に指定されたシフト JIS コード文字列を Unicode に変換します。戻り値に変換された文字列が設定されます。

メソッドが終了したときに、Source が空文字列に設定されている場合は、すべての文字の変換が行われたこととなります。空文字列でない場合は、その部分は変換されていません。

Source の設定について、「文字列の受け渡し方法」を参照してください。

■構 文

Code = Object. Sjis2Unicode (Source)

Sjis2Unicode 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	KANJI OCX オブジェクトです。
Source	シフト JIS コードの文字列を指定します。
Code	Unicode に変換された文字列です。

■戻り値

文字列(String)

サンプル

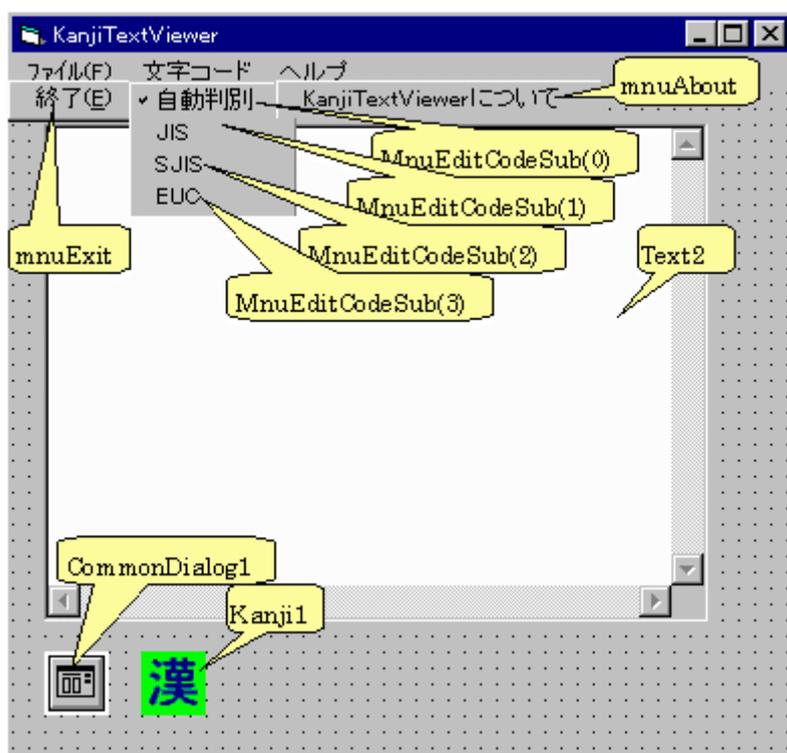
サンプル

白紙ページ

KVIEWER Ver.1.02

KVIEWER は汎用のテキストビューアーです。これを用いると、JIS コード、ShiftJIS コード、Euc コードのテキストファイルを開覧する事が出来ます。

開くメニューで指定したテキストファイルを ShiftJIS コードに変換して表示します。KVIEWER のパッケージにはサンプルファイルとして、SAMPLE_JIS.TXT、SAMPLE_Euc.TXT、SAMPLE_SJIS.TXT が同梱されています。



サンプル

白紙ページ

索引

Any2Euc メソッド	36
Any2Jis メソッド	34
Any2Sjis メソッド	35
Copyright プロパティ	30
DefaultEncoding プロパティ	29
DetectEncoding メソッド	33
Euc2Euc メソッド	46
Euc2Jis メソッド	40
Euc2Sjis メソッド	42
Euc2Unicode メソッド	50
Jis2Euc メソッド	39
Jis2Jis メソッド	44
Jis2Sjis メソッド	37
Jis2Unicode メソッド	51
KanjiIn プロパティ	25
KanjiOut プロパティ	26
KatakanaConvert プロパティ	27
LastEncoding プロパティ	28
RepairJis メソッド	43
Sjis2Euc メソッド	41
Sjis2Jis メソッド	38
Sjis2Sjis メソッド	45
Sjis2Unicode メソッド	52
Unicode2Euc メソッド	47
Unicode2Jis メソッド	48
Unicode2Sjis メソッド	49

白紙ページ

KANJIOCX マニュアル

1998年1月31日 初版第1版

1999年6月1日 第2版

1999年7月1日 第3版

2001年4月15日 第4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

本紙の内容を許可なく複写、転載、データファイル化することを禁じます。

本紙の内容に関するご質問は、上記のメールアドレス宛にお問い合わせください。