

INT64DLL

WILL

株式会社ウィル

- 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
動作環境について.....	4
インストール.....	5
サンプルを見る.....	7
サポートについて.....	8
バージョンアップについて(無償).....	9
再配布について.....	11
プログラミング概要.....	12
宣言.....	13
代入.....	14
演算.....	16
取得.....	19
プロパティ.....	21
Dec プロパティ.....	22
UnsignedDec プロパティ.....	23
Hex プロパティ.....	24
Octal プロパティ.....	25
Bin プロパティ.....	26
Byte プロパティ.....	27
NetByte プロパティ.....	28
Sign プロパティ.....	29
メソッド.....	30
AddDec メソッド.....	31
SubDec メソッド.....	32
MulDec メソッド.....	33
DivDec メソッド.....	34
uDivDec メソッド.....	35
ModDec メソッド.....	36
uModDec メソッド.....	37
AndDec メソッド.....	38
OrDec メソッド.....	39
XorDec メソッド.....	40
Neg メソッド.....	41

Rev メソッド	42
ShlDec メソッド	43
ShrDec メソッド	44
ShraDec メソッド	45
GetHighLong メソッド	46
GetLowLong メソッド	47
SetHighLong メソッド	48
SetLowLong メソッド	49
注 記	50
サンプル	51
Calc64	52
索 引	53

はじめに

はじめに

動作環境について

■対応 OS

INT64DLL 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
Active Server Pages(ASP)

■開発に必要なソフトウェア

INT64DLL 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)
Microsoft Visual Basic Scripting Edition (VB Script) Ver3 以上

■参照設定

INT64DLL を利用する際に参照設定が必要になる場合があります。その場合はプロジェクトの参照設定で WILL INT64DLL にチェックしてください。なおプログラミングの宣言の(その4)の場合は参照設定は不要です。

INT64DLL 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

セットアップキット (Willware.exe) をダブルクリックします。

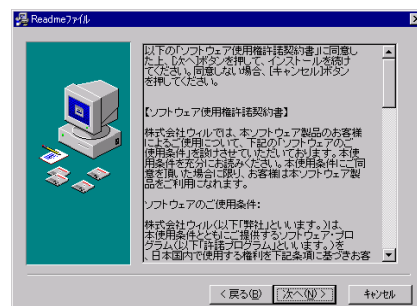


画面にしたがって、インストールを進めて下さい。

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。



はじめに

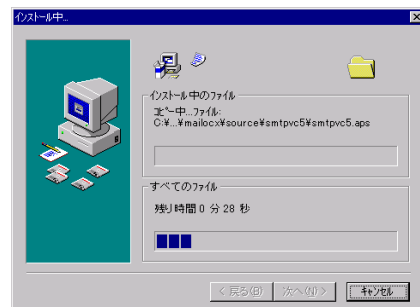
5. WILLWARE Components を登録するスタートメニュー又はプログラムマネージャのグループフォルダを指定します。初期設定では、新規に「WILLWARE Components」の名前でフォルダを作成します。特に指定する必要がなければ、初期設定をお勧めします。



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

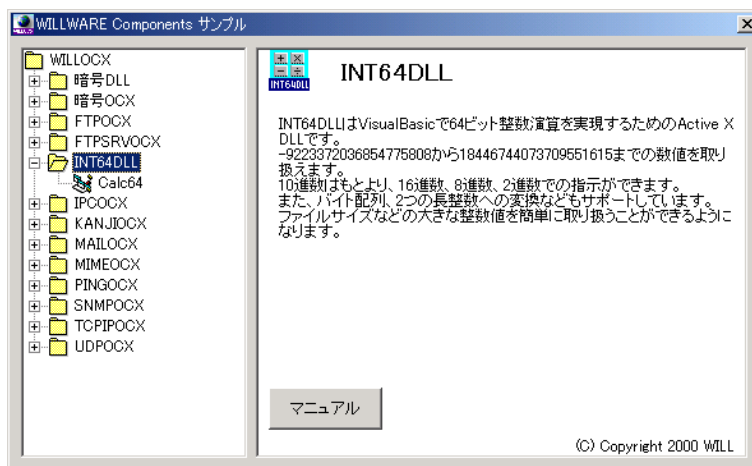


サンプルを見る

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。



はじめに

サポートについて

原則としてサポートは行いませんので、お問い合わせはご遠慮下さい。

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できます。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 : 「int64dll.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。

はじめに

■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

INT64DLL を単体あるいはアプリケーションに組込んでの再配布を許可します。

■配布時に必要な配布可能ファイル

INT64DLL は ATL を用いて作成していますので、int64.dll 以外に必要なファイルはありません。

■著作権

- ・ INT64DLL 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イル(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。
- ・ INT64DLL を利用するアプリケーションは INT64DLL 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 INT64DLL 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、INT64DLL 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

宣言

宣言の方法は 4 種類あります。

その 1 : Dim i64 As New Int64

その 2 : Dim i64 As Int64
Set i64 = New Int64

その 3 : Dim i64 As Int64
Set i64 = CreateObject("int64")

その 4 : Dim i64 As Object
Set i64 = CreateObject("int64")

代 入

Int64 オブジェクトに数値を代入する方法は以下のとおりです。

■ 符号付 10 進数を代入

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
i64.Dec = "-123456789"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ 符号なし 10 進数を代入

```
Dim i64 As New int64
i64.UnsignedDec = "18446744073709551615" '-1
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ 16 進数を代入

```
Dim i64 As New Int64
i64.Hex = "1234567890ABCDEF"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ 8 進数を代入

```
Dim i64 As New Int64
i64.Octal = "12345670"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ 2 進数を代入

```
Dim i64 As New Int64
i64.Bin = "10101010"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```


■ long で代入

```
Dim i64 As New int64
```

```
i64.SetHighLong &H12345678
```

```
i64.SetLowLong &H90ABCDEF
```

```
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ バイト(リトルエンディアン)で代入

```
Dim i64 As New Int64
```

```
Dim B(7) As Byte
```

```
B(0) = 1 'LSB
```

```
B(1) = 2
```

```
B(2) = 3
```

```
B(3) = 4
```

```
B(4) = 5
```

```
B(5) = 6
```

```
B(6) = 7
```

```
B(7) = 8 'MSB
```

```
i64.Byte = B
```

```
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ ネットワークバイト(ビッグエンディアン)で代入

```
Dim i64 As New Int64
```

```
Dim B(7) As Byte
```

```
B(0) = 1 'MSB
```

```
B(1) = 2
```

```
B(2) = 3
```

```
B(3) = 4
```

```
B(4) = 5
```

```
B(5) = 6
```

```
B(6) = 7
```

```
B(7) = 8 'LSB
```

```
i64.NetByte = B
```

```
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

演算

Int64 オブジェクトの内部に格納されている数値に対して演算する方法は以下の通りです。

■加法

```
Dim i64 As New Int64
i64.Dec = "-2"
i64.AddDec "3"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■減法

```
Dim i64 As New Int64
i64.Dec = "1"
i64.SubDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■乗法

```
Dim i64 As New Int64
i64.Dec = "-1"
i64.MulDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■除法

```
Dim i64 As New Int64
i64.Dec = "-4"
i64.DivDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■符号なし除法

```
Dim i64 As New Int64
i64.Hex = "FFFFFFFFFFFFFFFF"
i64.uDivDec "256"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■剰余

```
Dim i64 As New Int64
i64.Dec = "-4"
i64.ModDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■符号なし剰余

```
Dim i64 As New Int64
i64.Hex = "FFFFFFFFFFFFFFFF"
i64.uModDec "256"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■And

```
Dim i64 As New Int64
i64.Dec = "7"
i64.AndDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■Or

```
Dim i64 As New Int64
i64.Dec = "5"
i64.OrDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■Xor

```
Dim i64 As New Int64
i64.Dec = "7"
i64.XorDec "2"
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■左シフト

```
Dim i64 As New Int64
i64.Dec = "1"
i64.ShlDec 32
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■右論理シフト

```
Dim i64 As New Int64
i64.Dec = "-1"
i64.ShrlDec 8
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■右算術シフト

```
Dim i64 As New Int64
i64.Dec = "-1"
i64.ShraDec 8
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■ビット反転

```
Dim i64 As New Int64
i64.Dec = "-1"
i64.Rev
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■符号反転

```
Dim i64 As New Int64
i64.Dec = "-1"
i64.Neg
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

■int64 オブジェクト同士の演算

```
Dim i64 As New Int64 , j64 As New Int64
i64.Hex = "FFFFFFFF"
j64.Hex = "100000000"
i64.AddDec j64.Dec
Debug.Print i64.Dec, i64.UnsignedDec, i64.Hex, i64.Octal, i64.Bin
```

取得

Int64 オブジェクトの数値を取得する方法は以下の通りです。

■符号付 10 進数フォーマット

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.Dec
```

■符号なし 10 進数フォーマット

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.UnsignedDec
```

■16 進数フォーマット

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.Hex
```

■8 進数フォーマット

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.Octal
```

■2 進数フォーマット

```
Dim i64 As New Int64
i64.Dec = "123456789"
Debug.Print i64.Bin
```

■long フォーマット

```
Dim i64 As New Int64
i64.Hex = "1234567890abcdef"
Debug.Print Hex$(i64.GetHighLong), Hex$(i64.GetLowLong)
```

■ バイトフォーマット

```
Dim i64 As New Int64
Dim B() As Byte, i As Integer
i64.Hex = "1234567890abcdef"
B = i64.Byte
For i = LBound(B) To UBound(B)
    Debug.Print i, Hex$(B(i))
Next
```

■ ネットワークバイトフォーマット

```
Dim i64 As New Int64
Dim B() As Byte, i As Integer
i64.Hex = "1234567890abcdef"
B = i64.NetByte
For i = LBound(B) To UBound(B)
    Debug.Print i, Hex$(B(i))
Next
```

■ 符 号

```
Dim i64 As New Int64
i64.Dec = "123"
Debug.Print i64.Sign
i64.Dec = "0"
Debug.Print i64.Sign
i64.Dec = "-123"
Debug.Print i64.Sign
```

プロパティ

Dec プロパティ

■機 能

符号付 10 進数を扱います。

■構 文

Object.Dec[=Val]

Dec プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■データ型

文字列(String)

■戻り値

符号付 10 進数。

■解 説

“-9223372036854775808”から“18446744073709551615”までの値を設定できます。内部値 9223372036854775807(&H7fffffffffffffff)を超える値を取り出すと、負の値として扱われます。すなわち、“-9223372036854775808”から、“9223372036854775807”の値が戻ります。

UnsignedDec プロパティ

■機能

符号なし 10 進数を扱います。

■構文

Object.UnsignedDec[=Val]

UnsignedDec プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	INT64DLL オブジェクトです。
Val	符号なし 10 進数を示す文字列です。

■データ型

文字列(String)

■戻り値

符号なし 10 進数。

■解説

“-9223372036854775808”から“18446744073709551615”までの値を設定できます。

“-9223372036854775808”から“-1”の設定値は、

内部値 9223372036854775808(&H7fffffffffffffff)から、

18446744073709551615(&Hfffffffffffffff)となります。

取り出した値は、“0”から“18446744073709551615”の値になります。

Hex プロパティ

■機能

符号なし 16 進数を扱います。

■構文

Object.Hex[=Val]

Hex プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	INT64DLL オブジェクトです。
Val	符号なし 16 進数を示す文字列です。

■データ型

文字列(String)

■戻り値

符号なし 16 進数。

■解説

“0”から“FFFFFFFFFFFFFFFF”までの値を設定できます。16 進数を構成しない文字が含まれる場合、その文字を読み飛ばして処理します。符号を付けることはできません。負の値を得たい場合は、Neg メソッドと組み合わせてください。

-- 例 --

```
i64.Dec = "-256"
```

```
If i64.Sign < 0 Then
```

```
    i64.Neg
```

```
    Debug.Print "-" & i64.Hex
```

```
Else
```

```
    Debug.Print i64.Hex
```

```
End If
```

Octal プロパティ

■機 能

符号なし 8 進数を扱います。

■構 文

Object.Octal[=Value]

Octal プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	INT64DLL オブジェクトです。
Value	符号なし 8 進数を示す文字列です。

■データ型

文字列(String)

■戻り値

符号なし 8 進数。

■解 説

“0”から“17777777777777777777”までの値を設定できます。8 進数を構成しない文字が含まれる場合、その文字を読み飛ばして処理します。符号はつけることができません。負の値を得たい場合は、Neg メソッドと組み合わせてください。

-- 例 --

```
i64.Dec = "-256"  
If i64.Sign < 0 Then  
    i64.Neg  
    Debug.Print "-" & i64.Octal  
Else  
    Debug.Print i64.Octal  
End If
```


Byte プロパティ

■機能

内部データをバイト配列を使って扱います。

■構文

Object.Byte[=Val]

Byte プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	INT64DLL オブジェクトです。
Val	バイト配列(8 バイト)です。

■データ型

バイト配列またはバイナリ文字列。

■戻り値

8 バイトのバイト配列、または、8 バイトのバイナリ文字列。

■解説

配列の要素は 0 から始まり、7 まであります。0 の要素に、LSB のデータが格納され、7 の要素に MSB のデータが格納されます。(リトルエンディアン)

NetByte プロパティ

■機 能

内部データをネットワークオーダーのバイト配列を使って扱います。

■構 文

Object.NetByte[=Val]

NetByte プロパティの構文の指定項目は次の通りです。

(指定項目)	(内 容)
Object	INT64DLL オブジェクトです。
Val	ネットワークオーダーのバイト配列(8 バイト)。

■データ型

バイト配列またはバイナリ文字列。

■戻り値

8 バイトのバイト配列、または、8 バイトのバイナリ文字列。

■解 説

配列の要素は 0 から始まり、7 まであります。0 の要素に、MSB のデータが格納され、7 の要素に LSB のデータが格納されます。(ビッグエンディアン)

Sign プロパティ

■機能

内部値の符号を得ます。

■構文

Object.Sign[=Val]

Sign プロパティの構文の指定項目は次の通りです。

(指定項目)	(内容)
Object	INT64DLL オブジェクトです。

■データ型

長整数(Long)

■戻り値

(内部値)	(戻り値)
正	1
0	0
負	-1

■解説

内部値を符号付数値とみなして、正か負か 0 を判断し、それに応じて 1,-1,0 の値を返します。

メソッド

AddDec メソッド

■機 能

符号付 10 進数を足します。

■構 文

Object.AddDec(Val As String)

AddDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値に Val で指定された値を加えます。

SubDec メソッド

■機 能

符号付 10 進数を引きます。

■構 文

Object.SubDec(Val As String)

SubDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値から Val で指定された値を引きます。

MulDec メソッド

■機 能

符号付 10 進数をかけます。

■構 文

Object.MulDec(Val As String)

MulDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値に Val で指定された値をかけます。

DivDec メソッド

■機 能

符号付 10 進数で割ります。

■構 文

Object.DivDec(Val As String)

DivDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値を Val で指定された値で割ります。

uDivDec メソッド

■機 能

符号なし 10 進数で割ります。

■構 文

Object.uDivDec(Val As String)

uDivDec 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	INT64DLL オブジェクトです。
Val	符号なし 10 進数を示す文字列です。

■解 説

内部 64 ビット値を符号なし数値とみなし、これを Val で指定された符号なし数値で割ります。

ModDec メソッド

■機 能

符号付 10 進数で割ったあまりを内部値に設定します。

■構 文

Object.ModDec(Val As String)

ModDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値を Val で指定された値で割ったあまりを内部値に設定します。

uModDec メソッド

■機 能

符号なし 10 進数で割ったあまりを内部値に設定します。

■構 文

Object.uModDec(Val As String)

uModDec 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	INT64DLL オブジェクトです。
Val	符号なし 10 進数を示す文字列です。

■解 説

内部 64 ビット値を符号なし数値とみなし、これを Val で指定された符号なし数値で割ったあまりを内部値に設定します。

AndDec メソッド

■機 能

符号付 10 進数で And 演算を行います。

■構 文

Object.AndDec(Val As String)

AndDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値を Val で指定された値で And をとりその結果を内部値に設定します。

OrDec メソッド

■機 能

符号付 10 進数で Or 演算を行います。

■構 文

Object.OrDec(Val As String)

OrDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値を Val で指定された値で Or をとりその結果を内部値に設定します。

XorDec メソッド

■機 能

符号付 10 進数で Xor 演算を行います。

■構 文

Object.XorDec(Val As String)

XorDec 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	INT64DLL オブジェクトです。
Val	符号付 10 進数を示す文字列です。

■解 説

内部 64 ビット値を Val で指定された値で Xor をとりその結果を内部値に設定します。

Neg メソッド

■機 能

符号変換を行います。

■構 文

Object.Neg()

Neg 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	INT64DLL オブジェクトです。

■解 説

内部 64 ビット値の符号を変換し、その結果を内部値に設定します。

Rev メソッド

■機 能

ビット反転を行います。

■構 文

Object.Rev()

Rev 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	INT64DLL オブジェクトです。

■解 説

内部 64 ビット値のビットを反転してその結果を内部値に設定します。

ShlDec メソッド

■機 能

左シフトを行います。

■構 文

Object.ShlDec(ShiftCount As Long)

ShlDec 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	INT64DLL オブジェクトです。
ShiftCount	シフトするビット数(0 から 64)です。

■解 説

内部 64ビット値を ShiftCount で指定されたビット数だけ左論理シフトしその結果を内部値に設定します。

ShrDec メソッド

■機 能

右論理シフトを行います。

■構 文

Object.ShrDec(ShiftCount As Long)

ShrDec 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	INT64DLL オブジェクトです。
ShiftCount	シフトするビット数(0 から 64)です。

■解 説

内部 64ビット値を ShiftCount で指定されたビット数だけ右論理シフトしその結果を内部値に設定します。

ShraDec メソッド

■機 能

右算術シフトを行います。

■構 文

Object.ShraDec(ShiftCount As Long)

ShraDec 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	INT64DLL オブジェクトです。
ShiftCount	シフトするビット数(0 から 64)です。

■解 説

内部 64ビット値を ShiftCount で指定されたビット数だけ右算術シフトしその結果を内部値に設定する。

GetHighLong メソッド

■機 能

内部データの上位ロングワードを取り出します。

■構 文

Object.GetHighLong()

GetHighLong 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	INT64DLL オブジェクトです。

■データ型

長整数(Long)

GetLowLong メソッド

■機 能

内部データの下位ロングワードを取り出します。

■構 文

Object.GetLowLong()

GetLowLong 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	INT64DLL オブジェクトです。

■データ型

長整数(Long)

SetHighLong メソッド

■機 能

内部データの上位ロングワードに設定します。

■構 文

Object.SetHighLong(longVal As Long)

SetHighLong 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	INT64DLL オブジェクトです。
longVal	設定する long 値です。

SetLowLong メソッド

■機 能

内部データの下位ロングワードに設定します。

■構 文

Object. SetLowLong(longVal As Long)

SetLowLong 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	INT64DLL オブジェクトです。
longVal	設定する long 値です。

注 記

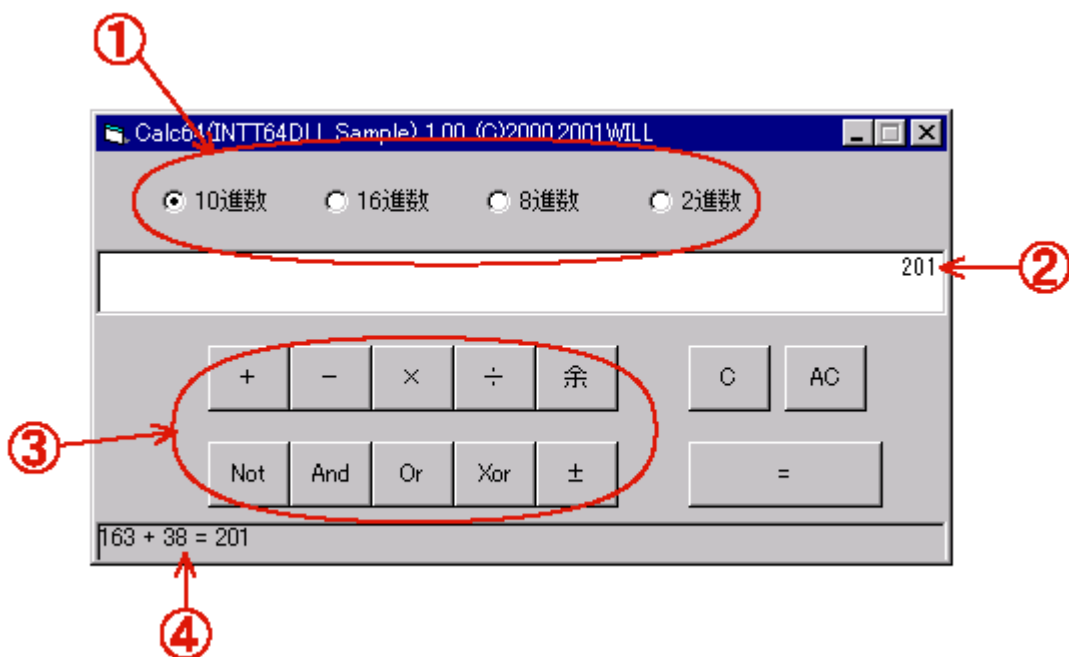
※ オーバーフローエラー及びアンダーフローエラーは検出しません。

※ 0 で除算するとトラップ可能なエラーが発生します。

サンプル

Calc64

Calc64 は、INT64DLL を利用した、64 ビット整数演算を行う計算プログラムです。
10 進数、16 進数、8 進数、2 進数の 4 種類で計算できます。



1. 取り扱う進数を①で選択します。
2. 計算する値を②に入力し、計算方法を③で指定します。再び②に値を入力し「=」を押すと、計算結果が②に表示されます。その計算結果に対して、続けて計算することができます。
3. 入力した計算式が④に表示されます。
4. 計算式全てを削除する場合は「AC」を押して下さい。②に入力した値だけを削除する場合は「C」を押して下さい。

索引

AddDec メソッド	31
AndDec メソッド	38
Bin プロパティ	26
Byte プロパティ	27
Dec プロパティ	22
DivDec メソッド	34
GetHighLong メソッド	46
GetLowLong メソッド	47
Hex プロパティ	24
ModDec メソッド	36
MulDec メソッド	33
Neg メソッド	41
NetByte プロパティ	28
Octal プロパティ	25
OrDec メソッド	39
Rev メソッド	42
SetHighLong メソッド	48
SetLowLong メソッド	49
ShlDec メソッド	43
ShraDec メソッド	45
ShrlDec メソッド	44
Sign プロパティ	29
SubDec メソッド	32
uDivDec メソッド	35
uModDec メソッド	37
UnsignedDec プロパティ	23
XorDec メソッド	40

INT64DLL マニュアル

1999年1月14日 初版第1版
2001年5月15日 第2版
2002年1月30日 第3版
2003年5月8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。