

# One Point Lesson

## 第五回「WindowsAPI を使ってみよう」

では、今回の OnePointLesson を始めましょう。

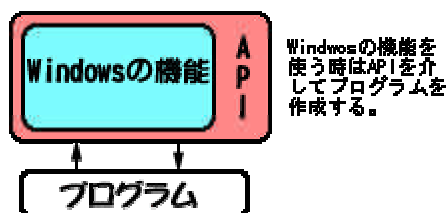
今回は、「WindowsAPI を使ってみよう」です。ちょっと高度なプログラム知識が必要ですが、F-BASIC では 比較的簡単に WindowsAPI を使うことができます。

そして、今週のプログラムでは WindowsAPI を使ってパソコンに入っている OS の種類を判別するプログラムを作ります。

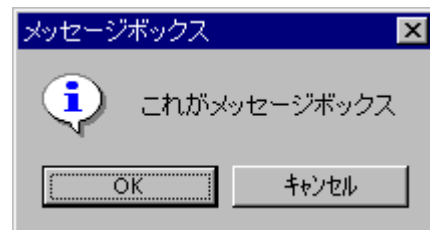
では、まず"API とは何か?"というところから始めましょう。

API とは、Application Program Interface の略で OS のさまざまな基本機能をまとめたものです。MS-DOS で言えばシステム・コールに相当するものです。イメージとしては 図 1 のような感じになります。

この代表的な例が、メッセージボックスです(図 2)。



(図 1)

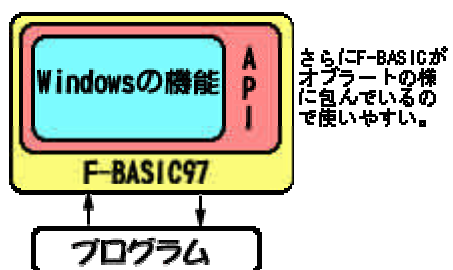


(図 2)

メッセージボックスを自分の力だけで表示しようとする大変です。(線をひいたり、色を塗ったり、文字を書いたり...) ところが、API を使うと `int MessageBox(HWND, Text, Title, Style)` と指定してやればメッセージボックスを 表示してくれます。楽ですね。

ここまでで気づかれた方もいるかと思いますが、F-BASIC の拡張命令にも MESSAGEBOX 命令があります。そうです！F-BASIC では MESSAGEBOX 命令を使えば WindowsAPI を使わなくてもメッセージボックスを作ることができます。しかも、MESSAGEBOX のパラメータは、MESSAGEBOX(タイトル, メッセージ, ボタン種, アイコン種) というように具体的な形で指定ができます。もっと楽ですね(図 3)。

では、このメッセージボックスのメッセージの部分にパソコンに入っている OS の種類とバージョンの情報を表示させてみましょう。



さらにF-BASICが  
オペラートの機  
能に包んでいるの  
で使いやすい。

(図 3)

## One Point Lesson

### 第五回「WindowsAPI を使ってみよう」

OS の情報を獲得するための API は GetVersionEX という API です。まず、この API を F-BASIC のプログラム内で使えるようにしてやらなくてはなりません。下の命令を使います。

```
declare sub プロシージャ名 lib "ライブラリ名" alias エイリアス名  
(パラメータ名)
```

プロシージャ名	F-BASIC プログラム内で使う名前を書きます。(GETVERSIONEX)
ライブラリ名	API が実際に定義されている DLL の名前を書きます。(kernel32・・・この中に GetVersionEx があります。)
エイリアス名	API の名前を書きます。(GetVersionEx・・・大文字・小文字は正確に！)
パラメータ名	引数を書きます。(_OSVERSIONINF)

これで、API の宣言ができました。

では、さっそく GETVERSIONEX を使って・・・といきたいところですが、その前に一つだけやる必要があります。この API "GetVersionEX"は引数の構造体のサイズを 構造体のメンバー DWOSVERSIONINFOSIZE に 入れる 初期化 処理を なければいけないのです。OS.DWOSVERSIONINFOSIZE = len (OS) この 1 文がその処理をしている部分です。では、GETVERSIONEX を使って OS の情報を獲得しましょう。

#### GetOS プログラムリスト

```
#include "windows.bi"  
  
' 構造体を宣言します。  
type _OSVERSIONINFO  
    DWOSVERSIONINFOSIZE as long
```

```
DWMAJORVERSION as long
DWMINORVERSION as long
DWBUILDNBER as long
DWPLATFORMID as long
SZCSDVERSION as string * 128
end type
```

'Windows API GetVersionEx の定義です。

```
declare sub API_GETVERSIONEX lib "kernel32" alias "GetVersionExA" (A as _OSVERSIONINFO)
```

```
SHOWWINDOW 0      '実行画面を消します。
```

'初期化処理

```
var OS as _OSVERSIONINFO
```

```
OS.DWOSVERSIONINFOSIZE = len (OS)
```

```
API_GETVERSIONEX OS      'OS のバージョンを獲得します。
```

'OS のバージョンによってメッセージを決めます。

```
if OS.DWPLATFORMID = 1 then
```

```
    MSG$ = "この OS は Windows95 です。"
```

```
else if OS.DWPLATFORMID = 2 then
```

```
    MSG$ = "この OS は WindowsNT です。"
```

```
else
```

```
    MSG$ = "この OS は Windows3.1 です。"
```

```
endif
```

'数値から文字型に変換します。

```
MAJOR$ = str$(OS.DWMAJORVERSION)
```

```
MINOR$ = str$(OS.DWMINORVERSION)
```

'メッセージにバージョン情報を追加します。

```
MSG$ = MSG$ + chr$(13)      '改行
```

```
MSG$ = MSG$ + chr$(10)
```

```
MSG$ = MSG$ + "バージョンは、"
```

```
MSG$ = MSG$ + MAJOR$
```

```
MSG$ = MSG$ + ". "
```

```
MSG$ = MSG$ + MINOR$
```

```
'メッセージボックスを表示します。
```

```
B=MESSAGEBOX("OS 情報", MSG$, 0, 3)
```

```
end
```

## One Point Lesson

### 第五回「WindowsAPI を使ってみよう」

さて、OS の情報が獲得できました。

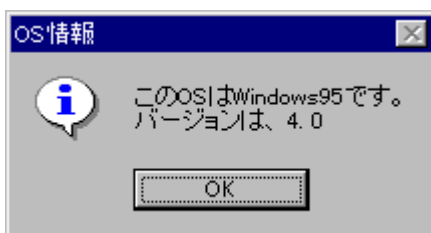
その情報を使って、今使っている OS を判別しましょう。

使用 OS	DWPLATFORMID の値
Windows95	1
WindowsNT	2
Windows3.1	0

この表のように DWPLATFORMID は使用している OS によって違った値を獲得するのでこれを使って OS を判別します。さらに、DWMAJORVERSION にはバージョンの上の桁、DWMINORVERSION にはバージョンの下を桁を獲得していますので これも使って、バージョン情報も表示するようにしましょう。

さあ、これで表示する情報はセットできました。あとは、MESSAGEBOX 命令を使って表示するだけです。

どうですか？ 図 4 の様なメッセージボックスが表示されましたよね？



(図 4)

どうですか、API の使い方分かってもらえましたか？ この API が自由に使えるようになれば F-BASIC にもない機能でも実現させることができるのです。（API は 1000 以上あります。API については Windows の関連書籍を参考にしてください）

さて、今回の OnePointLesson はここまでにしたいと思います。