

# 赤尾の Visual Basic 6.0ユーザーに贈る イチから出直し超基本プログラミング

## “現場で勝てる”

### Visual Basic

# 6.0

第7回

## アドレス帳で学ぶ実践テクニック その3 赤尾 猫太 AKAO, Nekota

### 「APIの利用」



### はじめに

さて、いつまでも同じサンプルでは飽きてしまうので、アドレス帳のサンプルは今回で片付けてしましましょう。今回はいろいろな機能を追加して完成させますが、その前にAPI (Application Programming Interface) について説明します。

この連載ではAPIについて深く説明する予定はありませんし、まったく使わないで進めようかとも思っていました。しかし、現実を考えると必要な場合があるので最小限の範囲で説明しておきます。

#### ■ APIの必要性

なぜ説明しないつもりだったAPIを説明するのかをもう少し詳しく説明しておきましょう。

#### 本稿で前提となるもの

OS Windows 98以上  
開発環境 Visual Basic 5.0/6.0



この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥AKAOフォルダ以下に収録しています

¥ADDRESS3:住所録サンプル

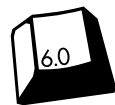
少なくとも本連載で学習を進めている人にはAPIはまだ必要ないと思います。

しかし、プログラム作りにおいて「こんな機能を付け加えたい」と思ってインターネットで調べたり、Tips集などの書籍を買ってきてサンプルのコードを見てみると、かなりの度合いでAPIが使用されています。

APIを使わずに高度な機能や便利な機能を付加するためには、市販されているコントロールを使うことが考えられますが、機能追加のために何万円も出せない場合がありますよね。

また、プログラミングの上達の早道は、いかにたくさんのコードを見て、理解して使うかだと言っても過言ではありません。そうしたことから、読者のみなさんにはAPIを使用するための最小限の知識は身に付けておいていただきたいのです。

VB.NETの解説などで「APIは使用しないほうがよい」だとか「APIを使用しなくても.NET Frameworkのクラスで実現できる」と書かれていたりしますが、実際に調査してみると、依然としてAPIなしでは実現できない機能が残っています。つまり、まだまだAPIなしではさまざまなプログラミングを行えないのも事実なのです。



### APIとは?

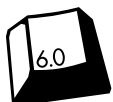
では、APIってなんなのでしょう。“Windowsの機能



の最小構成部品”だと言えればわかりやすいと思います。つまり、コンピュータのハードウェアにたとえば、基板やハードディスク、電源ユニットなどがVBのコントロールにあたり、抵抗やコンデンサやLSIなどの個々の部品がAPIにあたります。この個々の部品（API）の組み合わせによってWindowsのOSはもちろん、アプリケーションのさまざまな機能が実現されているのです。

コンピュータのハードウェアで考えると想像できますが、いちいちコンデンサなどの部品をハンダ付けして組み上げて自作パソコンを作る根性のある人は、ほとんどいませんよね。しかし、基板やハードディスク、CPUのようにある程度まとまった部品を買ってきて自作パソコンを作る人はいます。VBによるプログラミングは、この後者の「自作パソコンを作る」ということに似ています。つまり、比較的簡単に目的としたもの（プログラム）を作成できるのです。ハードウェアをコンデンサなどから作成する場合には、コンピュータというよりも電気の知識が必要になります。それと同じように、APIを使用するにはVB以外の知識が必要になるのです。

しかし、心配はいりません。ハードウェア作成と違ってAPIを利用する際には「サンプルコードをまねて使う」だけで目的を達成することができます。まねて使うだけなら、ごくわずかな知識でもAPIを使うことができます。もちろん、すべてのサンプルコードをまねるだけでは、自分の欲しい機能を実現することはできません。しかし、そのごくわずかな知識だけを記憶しておけば、APIでしか実現できない機能を組み込んだプログラムも作れる可能性が飛躍的に高くなるのです。



## APIの利用を宣言する

では、そのごくわずかな部分を覚えましょう。APIを使用するためには最初に「このAPIを使います」という宣言が必要です。サンプルコードなどを見ると、次のように記述されている部分です。

```
Declare Function GetPrivateProfileString Lib _  
"kernel32" Alias "GetPrivateProfileStringA" (_  
ByVal lpApplicationName As String, _  
ByVal lpKeyName As Any, ByVal lpDefault As String, _  
ByVal lpReturnedString As String, ByVal nSize As Long, _  
ByVal lpFileName As String) As Long
```

この場合、実際に使用するAPIの名前は「GetPrivateProfileString」です。他に、最小限理解する必要があるのは、()の中の型の宣言と、この場合では最後の「As Long」です。

()内の型の宣言部分は、最初のパラメータであるlpApplicationNameはString型、次がAny型というように、各パラメータのデータ型を宣言しています。しかし、いくらサンプルコードをまねるといっても、渡すパラメータを変えないと自分の目的のプログラムは作れないので、まねる際には注意してください。最後のAs Longは戻り値がLong型であるということを表わしています。

では、この宣言をどこに記述するのでしょうか？単一のフォームをもつプログラムならフォームのGeneralのDeclarationsに記述してください。その場合は、Declare Functionの前にPrivateと記述してください。複数のフォームをもち、それらのフォームでこのAPIを使用する場合、プログラム内に標準モジュールが存在するならPrivateは付加せずに標準モジュールのGeneralのDeclarationsに記述してください。標準モジュールの有無にかかわらず、そのAPIの使用場所が単一のフォームであればフォームのGeneralのDeclarationsにPrivateを付加して記述してかまいません。

これで準備は完了です。

## ■ API使用の注意点

次に使用方法ですが、いくつかの注意点があります。基本的にAPIのサンプルコードをまねて使用するときは、関連している部分のすべてをきちんとまねるようにしましょう。

例として次のコードを見てください。

```
d$ = String(256, 0)  
rc = GetPrivateProfileString(sect, keydata, defData, d$, 256, fnfn)  
IniGet = Left$(d$, InStr(d$, Chr$(0)) - 1)
```

