

- 「事前ハイディング」と「実行時ハイディング」、および、その相手先ファイルの「新規ファイルの作成」と「既存ファイルのオープン」の関係の一覧表
注意書きが多いので、それを全部消して、プログラムコードだけに見てみるとうっかりやしません。
(ただし、初心者の方にとっては、この注意書きは、「★重要事項」ですので必ず全て理解する必要があります。)

	事前ハイディング (VBEのツール-参照設定)	実行時ハイディング	
事前設定	VBEのツール-参照設定で手動設定。 あるいはVBAにて、「ActiveWorkbook.VBProject.References」などで自動設定。	なし	
有効期間	「操作元」となる「自ソフト」を開いている間じゅうずっと。	ハイディングするコードを書いた場所から、プログラムが終わるまで。終わったら無効になりますが不都合はありません。	
新規ファイルの作成に使うもの	<p>★ New キーワードとAddメソッド</p> <p>まず、New キーワードにて、自ソフトの、空の新しい「ガワ」(空のアプリケーションウィンドウ)を作ります。つまり、別個の新しいウィンドウにて、「空の状態のソフト」を起動させます。 起動させた後には、「非表示状態」なので、「××××Visible=True」にて表示させることが必要です。 そのあと、Addメソッドで、その「ガワ」の中に新しいファイルを作ります。 (仮装子は既定設定のとおりの前装子となります。)</p> <p>基本、別のアプリケーションウィンドウを開かないなら、New キーワードを使っても処理や変数宣言は変わりませんが、以下のサンプルではあく前提で書いてしまいました。</p> <pre>Dim xlApp As Excel.Application Dim bk01 As Workbook Set xlAppGawa = New Excel.Application xlAppGawa.Visible = True Set bk01 = xlAppGawa.Workbooks.Add</pre> <p>なお、新しい別個のアプリケーションウィンドウを「作らず」に、現在のアプリケーションウィンドウに自ソフトのファイルを表示させたい場合は、「New」キーワードは代わりに「Add」だけを使います。具体的には、上記の赤い行以外は全部削除して、かつ、「xlAppGawa」を「Application」に書き換えで実行します。</p> <p>※プログラムが終わったあとでも、作成したファイルを(閉じずに開いたまま)再度リモコン操作したい場合は、変数設定を、「Dim」ではなくて「Public」にして、プログラムの外で宣言します。(但し・なんらかの標準モジュールで、クラスオブジェクトモジュールでの宣言は不可。)とのプログラムからも再設定せずにリモコン操作ができます。 あるいは、都度、GetObject関数で再取得します。</p>	<p>★ New キーワードとAddメソッド</p> <p>「事前ハイディング」とまったく同じ方法です。</p>	
	<p>★ New キーワードとAddメソッド</p> <p>基本的には同上。 例えはExcelからWordに参照設定している場合は、=ExcelからWord3ファイルを新規作成した場合は、以下のように書き換えで実行します。</p> <ul style="list-style-type: none"> 「Excel.Application」を「Word.Application」に(2行あります)。 「As Workbook」を「As Document」に。 「Workbooks.Add」を「Documents.Add」に。 <p>→ 前述に「ドット」がなければそれを消さないように注意して。</p> <p>※本来は変数名も書き換えたほうがいいですが、初心者の方は、まずは、書き換えなくてやってみるほうで理解しやすいかもしれません。</p> <p>※また、ここでは、自ソフトと異なるソフトのファイルを新規作成するわけですので、「New」キーワードを使っている「アプリケーションウィンドウ=ガワ」を作る作業は省略できない必須の作業となります。</p> <pre>Dim Access01 As Access.Application Set Access01 = New Access.Application Access01.Visible = True Access01.UserControl = True Access01.NewCurrentDatabase = mdb ファイルのフルパス</pre> <p>(ただし、mdbをオブジェクトとして選んでくれない場合は、いったんフルで「GetObject関数で開くほうがいいか?」 「OpenCurrentDatabase」メソッドも、オブジェクトとして選んでくれないほうがいいです。)</p> <pre>Dim AccAppGawa As Access.Application Set AccAppGawa = New Access.Application AccAppGawa.Visible = True AccAppGawa.UserControl = True AccAppGawa.NewCurrentDatabase = True AccAppGawa.NewCurrentDatabase = mdb ファイルのフルパス</pre>	<p>★ CreateObject関数とAddメソッド</p> <p>以下、プログラム例。</p> <pre>Dim wdApp As Object Dim Doc01 As Object Set wdApp = CreateObject("Word.Application") wdApp.Visible = True Set Doc01 = wdApp.Documents.Add</pre> <p>※プログラムが終わったあとでも、作成したファイルを(閉じずに開いたまま)再度リモコン操作したい場合は、変数設定を、「Dim」ではなくて「Public」にして、プログラムの外で宣言します。(但し・なんらかの標準モジュールで、クラスオブジェクトモジュールでの宣言は不可。)とのプログラムからも再設定せずにリモコン操作ができます。 あるいは、都度、GetObject関数で再取得します。</p> <p>※また、ここでは、自ソフトと異なるソフトのファイルを新規作成するわけですので、「GetObject関数」を使っている「アプリケーションウィンドウ=ガワ」を作る作業は省略できない必須の作業となります。</p>	<p>「変数宣言」は明示的なオブジェクト型が指定できず、「As Object」や「As Variant」しか使えません。 「As Document」とか「As Workbook」とかが使えないのは、「事前ハイディング」されている時だけです。</p> <p>「CreateObject関数」は、 (1)全く何も無い状態から新しくWordやAccess等々の「別個のガワ(=空のアプリケーションウィンドウ)」を作ろう、 (2)プログラムが実行中のあいだのみ・それらに一併閉じハイディングをしてくれつつ、 (3)同時に、その作った「ガワ」を「オブジェクト」として返してくれませう。 (=Applicationオブジェクト)として返してくれませう。)</p> <p>そのような「返すこと」をしてくれます。 (事前ハイディングの「New」の動作を、1つの関数でやってくれます。)</p> <p>(ハイディングはファイルに対してではなく、ソフトウェアに対して行われます。)</p>
ハイディング先のソフトのファイルを新規作成	<p>★ New キーワードとOpenメソッド</p> <p>ここでの「オブジェクト」は、自ソフトを表す「オブジェクト式」のことです。 例えはExcelからWordに参照設定している場合は、=ExcelからWord3ファイルを新規作成した場合は、xlAppGawa.Workbooks が「オブジェクト式」に相当します。(下の例参照)</p> <pre>Dim xlAppGawa As Excel.Application Dim bk01 As Workbook Set xlAppGawa = New Excel.Application xlAppGawa.Visible = True Set bk01 = xlAppGawa.Workbooks.Open("D:\1\1.xlsx")</pre> <p>なお、新しい別個のアプリケーションウィンドウを「作らず」に、現在のアプリケーションウィンドウに自ソフトのファイルを表示させたい場合は、「New」キーワードは代わりに「Add」だけを使います。具体的には、上記の赤い行以外は全部削除して、かつ、「xlAppGawa」を「Application」に書き換えで実行します。</p> <pre>Dim bk01 As Workbook Set bk01 = Application.Workbooks.Open("D:\1\1.xlsx")</pre> <p>なお、「Application.Workbooks」は、階層構造を省略しないオブジェクト式の書き方ですので、「Application」を省略した場合は、「Workbooks」が「オブジェクト式」に相当する・・・ということになります。(ちなみにですが、階層構造を省略する書き方はおろそか、VBAは一向に上進しませんので、ぜひ、省略しない書き方を学んでください。)</p> <p>※ただし、Accessファイル(mdb)をオブジェクトとして操作したい場合は、「事前ハイディング」がしてあって「GetObject関数」を使っても大丈夫じゃないです。というが、これか、mdbを「オブジェクト」として扱えるほうが無いって感じです。「OpenCurrentDatabase」メソッドはあくだけ使っちゃってしまい、mdbをオブジェクトとして選んでくれないって感じで、オブジェクト変数への代入に使えませんが、別のメソッドでオブジェクトとして選んでくれるものがあればいいのですが・・・。</p>	<p>★ New キーワードとオブジェクト.Openメソッド</p> <p>以下、プログラム例。事前ハイディングよりも簡単かも。</p> <pre>Dim Doc01 As Object Set Doc01 = CreateObject("D:\1\1.docx") Doc01.Application.Visible = True</pre> <p>※「GetObject関数」は、「NewVCreateObject」で「別個のガワ」新しく作る作業も、同時にしてくれませう。 ※プログラムが終わったあとでも、開いているファイルを(そのまま閉じずに)再度リモコン操作したい場合は、変数設定を、「Dim」ではなくて「Public」にして、プログラムの外で宣言します。(但し・なんらかの標準モジュールで、クラスオブジェクトモジュールでの宣言は不可。)とのプログラムからも再設定せずにリモコン操作ができます。 あるいは、都度、GetObject関数で再取得します。</p> <p>※また、ここでは、自ソフトと異なるソフトのファイルを新規作成するわけですので、「GetObject関数」を使っている「アプリケーションウィンドウ=ガワ」を作る作業は省略できない必須の作業となります。</p> <p>AccessWordなどからExcelのファイルを開くような場合は、2010以前のバージョン「字ウィンドウ」があるのを、その場合は、ガワだけは見えるけど・ブックが表示されないことがあります。(字ウィンドウが非表示のまま開いてしまっている状態です。)</p> <p>その時は、さらに「Application.Opens(「ファイル名」).Visible = True」というコードが有効です。</p> <p>ただし、このとき、「Windows(1)」と書いてしまうと、もし「Personal_xls」が自動生成されていたら場合はそれが表示されてしまうので、インテックス番号は使わないようにしましょう。素直にファイル名だけを使います。</p> <p>※また、ブックのオブジェクト変数がPublicでなければ、プログラム終了と同時にオブジェクト変数も解放されます。そのためにも、プログラム途中で「再表示する準備」を行って・今一度、目的のファイルを出してきますが、プログラムが終わってしまおうと再表示のメニューの中には目的のファイル名が残っています。ブックのオブジェクト変数がPublicだと、プログラムが終了しても、再表示のメニューの中には目的のファイル名が残ってしまいます。</p>	<p>こちらも、すでに、自ファイルに対しては「事前ハイディング(参照設定)」が完了済みのため、「事前ハイディング」でのNewとAddを使う方法とまったく同じです。もちろん「変数宣言」と「ObjectやVariant」以外しか使えないのは「事前ハイディング」されている時だけです。</p> <p>「Openメソッド」も「CreateObject関数」も必要なく、GetObject関数だけで、アプリケーションウィンドウ(ガワ)とファイルの2つともが開きます。</p> <p>「GetObject関数」は、 (1)全く何も無い状態から新しくWordやAccess等々の「別個のガワ(=空のアプリケーションウィンドウ)」を作ろう、 (2)プログラムが実行中のあいだのみ・それらに一併閉じハイディングをしてくれつつ、 (3)同時に、その開いたファイルを「オブジェクト」として返してくれませう。 (=Applicationオブジェクト)として返してくれませう。 (=Applicationオブジェクト)といったオブジェクトとして返してくれませう。)</p> <p>そのような「返すこと」をしてくれます。 (事前ハイディングの「New」と「Open」の動作・・・つまり、 =「別個のガワ作成と既存ファイルオープンの2つの動作」を、1つの関数でやってくれます。)</p> <p>(こちらもハイディングはファイルに対してではなく、ソフトウェアに対して行われます。)</p> <p>また、「既存ファイル」が「すでに開いている時」もそのファイルを開くだけで開くことができます。 その時、</p> <p>何ファイル(ブック)からガワが開く場合は・・・ 「Set ×× = GetObject(「フルパス」)」と取得して、ガワは「Set = ××.Application」と取得します。</p> <p>何ガワが開く場合は・・・ 「Set ×× = GetObject(「Excel.Application」)」(カンマに注意。)と取得して、ブックは「Set = ××.Workbooks(「ファイル名」)」などと取得します。 「Set ×× = GetObject(「フルパス」)」と取得して、ガワは「Set = ××.Application」と取得して、ブックは「Set = ××.Workbooks(「ファイル名」)」などと取得します。</p> <p>ただし、いずれも、「字ウィンドウ」に先に自分のファイルが開いていると、例えはオブジェクト変数の書き換えで「ガワ」を開いてしまったときに、それらのファイルも全部・消滅してしまったり、そこで注意します。</p>

注意書きが多いので、それを全部消して、プログラムコードだけに見てみるとうっかりやしません。

(ただし、初心者の方にとっては、この注意書きは、「★重要事項」ですので必ず全て理解する必要があります。)

★★重要！！

「ファイル」に相当するオブジェクト変数の宣言を、標準モジュールにて、「Dim」ではなくて「Public」で書くと、操作元のExcelファイルを開いている間じゅう、オートメーションでの操作ができるようになります。