

AddinBox

【 RefEdit コントロール 代替手法 】

— RefEdit Support —

(clsRangeSelection クラス)

[AddinBox / 油断のならない RefEdit]

http://addinbox.sakura.ne.jp/Excel_Tips08.htm

http://addinbox.sakura.ne.jp/Excel_Tips08_2.htm

(旧サイト) <http://www.h3.dion.ne.jp/~sakatsu/>

(この文書は上記の解説をローカル作業時の閲覧用に PDF 化したものです)

Ver 2.00 : 2010年 4月 24日

Ver 2.10 : 2010年 5月 2日

Ver 2.20 : 2010年 5月 14日

Ver 2.30 : 2010年 12月 15日

Ver 2.31 : 2010年 12月 25日

Ver 2.40 : 2011年 7月 1日

AddinBox / K.Tsunoda

CopyRight (C) 2010 Allrights Reserved.

(A4で割付印刷 (2頁/枚) して山折りで綴じると小冊子になります)

～～～ 目 次 ～～～

(※) : ユーザーサイドでは使用しません

§ 1	RefEdit コントロールの問題点	3
1-1	セルアドレス文字から Range オブジェクトに変換	3
1-2	イベント内でシート書き出しを行なうと Excel が落ちる	6
1-3	イベント内での実行時エラーがマクロを強制終了させる	7
1-4	[Excel2003 以前]&[Excel2007 以降]のマルチインストール環境での問題	9
§ 2	RefEdit コントロール代替手法	13
2-1	基本動作	13
2-2	ソースコード	16
§ 3	RefEdit コントロール代替手法 – クラスモジュール版 –	19
3-1	基本動作	19
3-2	注 意 事 項	22
3-2-1	一 般 事 項	22
3-2-2	RefEdit 代替手法を利用する際に排他制御が必要な理由	22
3-2-3	ブック切替機能とブック参照の解放	24
3-3	利用する際の準備	27
3-4	プロパティ/メソッド 一覧	28
3-4-1	定 義	28
3-4-2	Show メソッド	29
3-4-3	InitRefEdit メソッド	30
3-4-4	SetRefEdit メソッド	31
3-4-5	Clear メソッド	32
3-4-6	Terminate メソッド	32
3-4-7	EndRefEdit メソッド (※)	33
3-4-8	BookChange メソッド (※)	33
3-4-9	ReleaseBookReference メソッド	33
3-4-10	StartUpBook プロパティ	34
3-4-11	TargetRange プロパティ	34

3-4-1-2	TargetBook プロパティ	35
3-4-1-3	TargetSheet プロパティ	36
3-4-1-4	TargetAddr プロパティ	37
3-4-1-5	ActivateInfo プロパティ	38
3-4-1-6	RefEditActive プロパティ (※)	38
3-4-1-7	IsRefEditActive 関数	39
3-4-1-8	CreateRangeSelectClass 関数	40
3-5	利 用 例	41
3-6	F A Q	48
3-7	修 正 履 歴	51

§ 1 RefEdit コントロールの問題点

1-1 セルアドレス文字から Range オブジェクトに変換

RefEdit コントロールで得られる Value プロパティは String 型 ですので、そのセルアドレス文字列から Range オブジェクトを取得する必要があります。一般に解説されているのは下記のような方法です。

```
Dim MyRng As Range
Set MyRng = Nothing
On Error Resume Next
Set MyRng = Range(RefEdit1.Value)
On Error GoTo 0
If (MyRng Is Nothing) Then
    MsgBox "セル範囲を選択して下さい"
End if
```

[A1 形式]環境ならば何も問題ないのですが、[R1C1 形式]環境では『RefEdit コントロールの Value も、R1C1 形式のアドレス文字列』となります。ところが、肝心の Range プロパティの引数には【A1 形式】しか指定できないので、R1C1 環境で上記マクロを実行した場合、RefEdit コントロールでちゃんとセルを選択してもエラー扱いになってしまいます。

同様の機能を提供する「InputBox メソッド(Type:=8)」では、そもそも返却値自体が[Range オブジェクト]で得られますので、A1 形式/R1C1 形式という環境の違いには影響されません。

対応策としては、

R1C1 形式で得られた文字列を A1 形式に変換

してから Range プロパティを通すという方法が考えられます。この変換処理には[ConvertFormula メソッド]が利用できます。

```
Dim MyRng As Range
Dim strRng As String
If (RefEdit1.Value = "") Then
    MsgBox "セル範囲を選択して下さい"
Else
    With Application
        If (.ReferenceStyle = xlA1) Then
            strRng = RefEdit1.Value
        Else
```

```

    On Error Resume Next
    strRng = .ConvertFormula(RefEdit1.Value, xlR1C1, xlA1)
    If (Err.Number <> 0) Then
        strRng = ""      ' Range プロパティでエラーにさせる
    End If
    On Error GoTo 0

    End If

    End With

    Set MyRng = Nothing
    On Error Resume Next
    Set MyRng = Range(strRng)
    On Error GoTo 0
    If (MyRng Is Nothing) Then
        MsgBox "セル範囲を選択して下さい"
    Else
        MsgBox "OK:" & MyRng.Address
    End If
End If

```

ちょっと長いですね。ConvertFormula を「使うか(R1C1 形式)/使わないか(A1 形式)」の判断と、ConvertFormula へのエラー処理が必要なので、こんなコードになります。

ところで、セルアドレス文字列を Range オブジェクトに変換する方法には、Range プロパティの他に【Evaluate メソッド】というのがあります。この Evaluate メソッドのヘルプには「A1 形式の参照～」というような事が書かれていますが、R1C1 形式でも変換してくれます。そうすると、上記のように「A1 形式なら……、R1C1 形式なら……」という風にする必要もありません。

```

Dim MyRng As Range
Set MyRng = Nothing
On Error Resume Next
Set MyRng = Application.Evaluate(RefEdit1.Value)
On Error GoTo 0
If (MyRng Is Nothing) Then
    MsgBox "セル範囲を選択して下さい"
End if

```

上記で、Evaluate メソッドは

A1 形式のアドレス文字でも RIC1 形式のアドレス文字でも変換できる

と書きましたが、

[A1 形式]環境では、RIC1 形式のアドレス文字はエラー

[RIC1 形式]環境では、A1 形式/RIC1 形式どちらのアドレス文字でも OK
という事が判りました。

ただし、上記の使用方法では、

[A1 形式]環境では、RefEdit が A1 形式アドレス文字を返す

[RIC1 形式]環境では、RefEdit が RIC1 形式アドレス文字を返す

となりますから、

[A1 形式]環境で RIC1 形式アドレス文字が Evaluate メソッドに渡る
という事は起きませんので問題はありません。

1-2 イベント内でシート書き出しを行なうと Excel が落ちる

RefEdit コントロールを使う処理では【ワークシートへの書き出し】時に Excel がダウンする場合があります。下記のタイミングで【ワークシートへの書き出し】を行なうと、UserForm が強制的に UnLoad されます。事前にブレークポイントを設定しておいて、デバッグモードで『続行』を選択すると

” 不正な処理を行なったので強制終了”

と表示され、EXCEL がダウンします。そのブックは『編集集中』となり、再立ち上げをするまでロックされます。問題となるイベントタイミングは

『RefEdit の KeyDown』から

『TextBox の MouseDown』又は『CommandButton の Enter』

の間です (TextBox/CommandButton 以外のコントロールでも同様)。イベントの発生順序を並べると下記の順になりますから、これらのイベントは要注意になります。

(RefEdit)

Enter > MouseDown > MouseUp > KeyDown > Change > KeyUp > . . . > Exit →

(TextBox)

→ Enter > MouseDown > MouseUp > KeyDown > Change > KeyUp > . . . > Exit

(CommandButton)

→ Enter > MouseDown > MouseUp > Click . . . > Exit

とはいえ、次の2点に注意していれば問題は無いでしょう。

- (1) RefEdit コントロールのイベントは一切使わない。
- (2) RefEdit コントロールを利用する場合は、他の全てのコントロールで[Enter]イベントは使わない (マウスで、何処に移るか判りませんからね)。

その問題のイベントでも「ワークシートへの書き出し」が無ければ、何も問題は無いんですが、そのイベントがマクロコードに載っていると、将来うっかりと

「あっ、丁度このイベントマクロが有るから、ここで書き出してしまえ」

というような事も無きにしも非ずで……。そういう事も有り得ますから、RefEdit コントロールを使っている場合は、UserForm モジュールの先頭に下記のような【注意書き】をコメントしておくのが良いでしょう。

```
' ///////////////////////////////////////////////////  
' /   RefEdit を使っているので、RefEdit の[イベント全て]と、  
' /   各コントロールの[Enter]で、ワークシートへの書き出しは厳禁  
' ///////////////////////////////////////////////////
```

1-3 イベント内での実行時エラーがマクロを強制終了させる

RefEdit コントロールにイベントルーチンを記述している場合には、

そのイベント内で実行時エラーが発生しない

ようにエラーチェックを厳重に行なう必要があります。

TextBox など通常のコントロールの場合、そのイベントルーチン内で実行時エラーが発生すれば

実行を止めて VBE に移る・エラーメッセージの表示・エラー発生箇所のハイライト

といった動作をしてくれます。

しかし、RefEdit コントロールの場合には同様のデバッグサポート動作は機能しないようです。

RefEdit コントロールから他のコントロールへフォーカスが移った (RefEdit コントロールの Focus Lost タイミング) 時点で UserForm が異常終了します (エラーメッセージの表示も無く UserForm が消える)。

簡単なサンプルを紹介します。UserForm に TextBox1/TextBox2/RefEdit1 を配置して以下のマクロを記述します ("Error 424" は擬似的に、[424 エラー]を発生させる命令です)。

```
Private Sub TextBox2_Change()
```

```
    Error 424
```

```
End Sub
```

```
Private Sub RefEdit1_Change()
```

```
    Error 424
```

```
End Sub
```

UserForm を表示して TextBox2 へ適当に書き込むと、1文字入力した瞬間 (Change イベント時) に実行時エラーが発生して [Error 424] のステップがハイライトされます。

RefEdit1 でセルを選択しても、その瞬間 (Change イベント時) にはエラーとはなりませんし、別セルを選択し直す事も出来ます (この時にもエラーは起きません)。

ところが、タブキーを押したり、TextBox1 をクリックして RefEdit1 からフォーカスが他コントロールへ移った瞬間に UserForm が落ちてしまいます。この後は、VBEにて『リセット (■ボタン)』を行なわないと復旧しません。

この障害は、RefEdit コントロールが参照設定で取り込む【外部のコントロール】である事に起因しているものと思われます。コントロールから Excel アプリケーションへのエラー通知がエラー発生時にリアルタイムで行なわれず、制御を離れる (Focus Lost) 際にしか通知出来ていない設計故に、エラー発生場所などの情報を十分に引き継ぐ事が出来ないのでしょうか。

これは、Excel アプリケーションから見れば、「どこで、どんなエラー」が発生したのか正確な情報が得られないのですから、このまま実行を続けるのは危険性大として、とにかく直ちに止めるしかないという状況なのでしょう (エラー情報が無いので、ユーザーへのメッセージを出す事も出来ず、コードをハイライト化する事も出来ず)。

RefEdit コントロールで、この問題に会う可能性が高いものは

【 RefEdit1.Value (String 型) から セルオブジェクト (Range 型) への変換 】

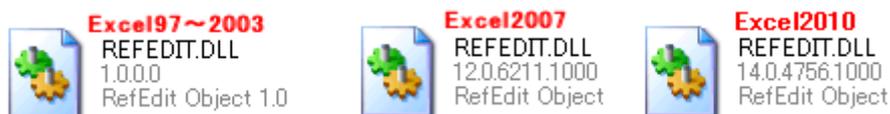
という処理でしょう。この処理には次項で解説している『 Evaluate メソッド 』を使いますが、その際にエラー判定が行なわれていないと、この障害に会う事となります。一例としては『 RefEdit1.Value が空文字 (Del キー等で入力内容のクリアなど) 』の場合が挙げられます。

ただし、この問題は、RefEdit コントロールのイベントルーチン内で処理している場合の話です。RefEdit にはイベントルーチンが無く、セルオブジェクトへの変換などを他のコマンドボタンのクリックイベント内で行なっている場合などでは、この障害は出ません (普通にエラー表示されます)。

尚、この障害は、RefEdit コントロールに限らず、【参照設定で取り込む外部のコントロール】全般で起こりうる問題だと思います。外部コントロールを使用する場合には、実行時エラーが起こる可能性のある処理を入念に洗い出して、エラーを回避できるコードを構築するようにしてください。

1-4 [Excel2003 以前]&[Excel2007 以降]のマルチインストール環境での問題

RefEdit コントロールは Excel2007 からバージョンアップされています（セル参照文字列として扱えるデータが '255文字' から '8KB' へ拡張されました）。



RefEdit コントロールの本体である [RefEdit.dll] はバージョンによって以下の場所に格納されています。マルチインストールの場合には各々のフォルダに存在します。

Excel97 : C:\WINDOWS\system32 (ここは DLL の既定フォルダ(パスの通ったフォルダ)です)
Excel2000 : C:\Program Files\Microsoft Office\Office
Excel2002 : C:\Program Files\Microsoft Office\Office10
Excel2003 : C:\Program Files\Microsoft Office\Office11
Excel2007 : C:\Program Files\Microsoft Office\Office12
Excel2010 : C:\Program Files\Microsoft Office\Office14

注) DLL のバージョン表記は Excel97 と Excel2000~2003 どちらも [1.0.0.0] ですが、Excel97 の VBAが[Ver.5]であり、RefEdit の動作も一部異なりますので、2000~2003 のものとは別物です。

RefEdit コントロールを使用すると、[VBE/ツール/参照設定]ダイアログにおいて、RefEdit コントロールの参照設定が追加されます。参照設定の内容を見ると、RefEdit.dll のフルパスを見る事ができます。

この RefEdit コントロールを使ったブック（参照設定の付いたブック）を、

- ・ 他の Excel バージョンの PC

または、

- ・ 同じバージョンの Excel であっても Excel のインストールフォルダのパスが異なっている PC

例) C:\Program Files\Microsoft Office\Office10

C:\Program Files\Microsoft Office2002\Office10

で開いた場合を試すと、バージョンによって扱いが異なる事が判りました。

- a) Excel97~2003 で作成したブックを、Excel97~2003 で開く（他バージョン or パス違い）
- b) Excel97~2003 で作成したブックを、Excel2007 で開く（互換モード）
- c) Excel2007 で作成した 97~2003 互換ブックを、Excel97~2003 で開く
- d) Excel2007 で作成したブックを、Excel2007 で開く（パス違い）

a) , b) , d) の場合には、自動的に、RefEdit の参照設定が開いた Excel のパスに変更されており、エラーも無く、動作に何の問題もありません。

c) の場合には、97 と 2000～2003 で異なります。

- Excel2000～2003 で開くと、参照設定のパスが[作成した環境時のまま]となっており、参照設定ダイアログ内において【参照不可】のフラグが立っています。
- Excel97 で開くと、参照設定が変更され何の問題もありません（これは、VBAが Excel97 は Ver. 5、Excel2000～2007 は Ver. 6 という違いによるものでしょう）。

[参照不可]についての Microsoft の回答(2009/5/7)

- Excel2007 より、RefEdit の機能が拡張された為に、Excel2003 以前のものとは「タイプライブラリ」の内容が変わりました(255文字→8KB)。
- 上位互換であるので、Excel2003 以前の RefEdit 付きブックは Excel2007 でも動作します(低機能(2003 以前の RefEdit)は高機能(2007 の RefEdit)の中に完全包含)。
- 逆に、Excel2007 の RefEdit 付きブックを Excel2003 以前で開いた場合、高機能(2007 の RefEdit)処理を低機能(2003 以前の RefEdit)ではカバー出来ない為、その部分で、予期せぬエラーが起こる可能性が出て来ます。
- これを未然に防ぐ意味で【使用不可】となるように[参照不可]状態にする仕様になっています。
- したがって、自動的に、この[参照不可]状態を回避する手段はありませんので、
 - a) マニュアル操作で Excel2003 以前の RefEdit を参照設定し直す
 - b) それで問題なく動作するか動作確認するという対応がユーザーに求められます。

この【参照不可】の付いた状態のブックですが、実はこのままでも RefEdit コントロールを載せた UserForm は問題なく動作します(これもまた謎のひとつなんですが)。試しに、UserForm に RefEdit コントロールを載せ、

```
Private Sub UserForm_Click()  
    MsgBox RefEdit1.Value  
End Sub
```

というイベントプロシジャールを書いたブックを用意してテストすると、普通にセル範囲の指定ができますし、MsgBox も表示されます。

では、ここで問題にしている「エラーを引き起こす」とは何なのでしょう？

上記のイベントマクロを

```
Private Sub UserForm_Click ()  
    MsgBox RefEdit1.Value & Format(Date, "yyyy/m/d")  
End Sub
```

と修正してみます。これを試すと、今度は

「プロジェクトまたはライブラリが見つかりません」

というコンパイルエラーが発生します。VBEでエラー箇所を見ると、**Date** の部分がハイライトしています。文法的に全く問題のないマクロです。Date の部分を数値(2000 とか)に変えてみると、今度は **Format** の部分がエラーになります。

【参照不可】が、Date などの極当たり前に使える関数をコンパイルエラーにしてしまうというのは、良くある事です（これはFAQといって良いでしょう。「Date がエラーになります」という質問に対しては、『参照不可マークの付いている項目がありませんか』という回答が常套句ですね）。

ここで、このブックをアドイン等のツール頒布で行なう「プロジェクトロック」をしてみるとどうなるでしょう。この場合には、

「非表示モジュール xxx 内でコンパイルエラーが発生しました」

というエラーメッセージが表示されます。

以上の説明から単に

「Excel2003 以前でも使うブックを、Excel2007 の互換モードで作成する場合、RefEdit コントロールは使用できない。元々、非互換だと諦めて RefEdit を使わないようにするか、セオリー通りに下位バージョンで作成する。」

と締め括って、そこで終わりにしてしまいたくなりますが、ちょっと待ってください。

確かに、複数バージョンで使うブックの作成は下位バージョンで行なうのが基本です。その場合、上位バージョンでの動作確認を容易にする為に、開発PCでは

複数のバージョンの Excel をマルチインストール

している人が多いのではないのでしょうか。

そこで、Excel2000～2007 で使う前提で、Excel2000, 2002, 2003, 2007 がマルチインストールされた P C を使い、Excel2000 にて RefEdit を使ったブックを作成するとどうなるのでしょうか？

マルチインストールされている開発 P C では、どのバージョンで開いても、【参照不可】にはなりませんし、当然コンパイルエラーにもなりません。これで「問題は無い」と判断して、Excel2000～2007 が単独でインストールされている各 P C に持って行くと……

- Excel2007 では問題なし
- Excel2000～2003 では【参照不可】 & コンパイルエラー
(開発バージョンと同じ Excel2000 でもエラーになります)

となります。

これは何故でしょう。ブックを作ったのは Excel2000 です。Excel2007 で上書き保存はしていません。Excel2002/Excel2003 でもエラーにならないのは確認しました。でも、他の P C ではエラーになってしまいます。

原因は、マルチインストール環境における、RefEdit コントロールの参照設定の有り方です。

マルチインストール環境では、各バージョンのフォルダに個々の RefEdit. dll がありますが、参照設定に載るのは『最後にインストールされたバージョン』のみです (何れのバージョンの RefEdit. dll も、その CLSID が同じである為、レジストリに残るのは最後にインストールされたものとなります)。

通常は[古→新]の順でインストールします。したがって、Excel2000～2007 のマルチインストール環境では、常に Excel2007 の RefEdit. dll しか参照設定されません。つまり、

Excel2000 でブックを作っても、この場合は参照設定には Excel2007 の RefEdit. dll が入ります。

この為、上で説明したパターン通りのエラーが発生してしまう事となります。開発 P C での Excel2002 /2003 でエラーにならないのは、元々その P C に有るものを (参照可能なパスで) 参照しているからです。

以上のことから、このトラブルを回避するには

マルチインストール環境には Excel2007 を含めない。

動作確認で必要となる Excel2007 は別 P C (または、Virtual PC) に用意する。

という配慮が必要となります。尚、開発 P C には単独バージョンしかない(2007 以外)という場合には、そもそも、このような問題は起きませんので何ら気にする必要はありません。

§ 2 RefEdit コントロール代替手法

2-1 基本動作

実際に利用する場合は【クラスモジュール版】を使用してください。

RefEdit コントロールは上記の解説のように様々な問題を抱えており、使わないで済めば、それ以上越したことは無いコントロールです。しかし、セル選択のユーザーインターフェースとして、このコントロールに代わるものが無い現状では、問題事項に引っ掛からないよう留意しつつ慎重に使うしかありませんでした。

ここでは、モードレスフォームを利用して RefEdit コントロールと同じ動作をさせる方法を解説します。まず、簡単に説明すると、

- ・ 一旦、ユーザーフォームを Hide モードで閉じて、呼び出し元の標準モジュールに戻る。
- ・ 標準モジュールからモードレスの RefEdit 代替フォームを表示。
- ・ 同時に、ブックのセル選択監視イベントをスタートさせ、セル範囲を取得。
- ・ セル選択の間 (RefEdit 代替フォームの表示中)、呼び出し元の標準モジュールではループ待機。
- ・ RefEdit 代替フォームを閉じた通知を受けたらループ待機を抜け、取得したセル範囲を設定してユーザーフォームを再表示 (Hide なので、前回内容が残っている)。

という手順になります。

モードレスフォームを使う為、Excel97 では利用できません。Excel2000 以上のバージョンが対象です。



(左記の RefEdit ボタンは通常のコマンドボタンに画像を載せて作ってあります。セルアドレス欄は ラベルコントロールです)

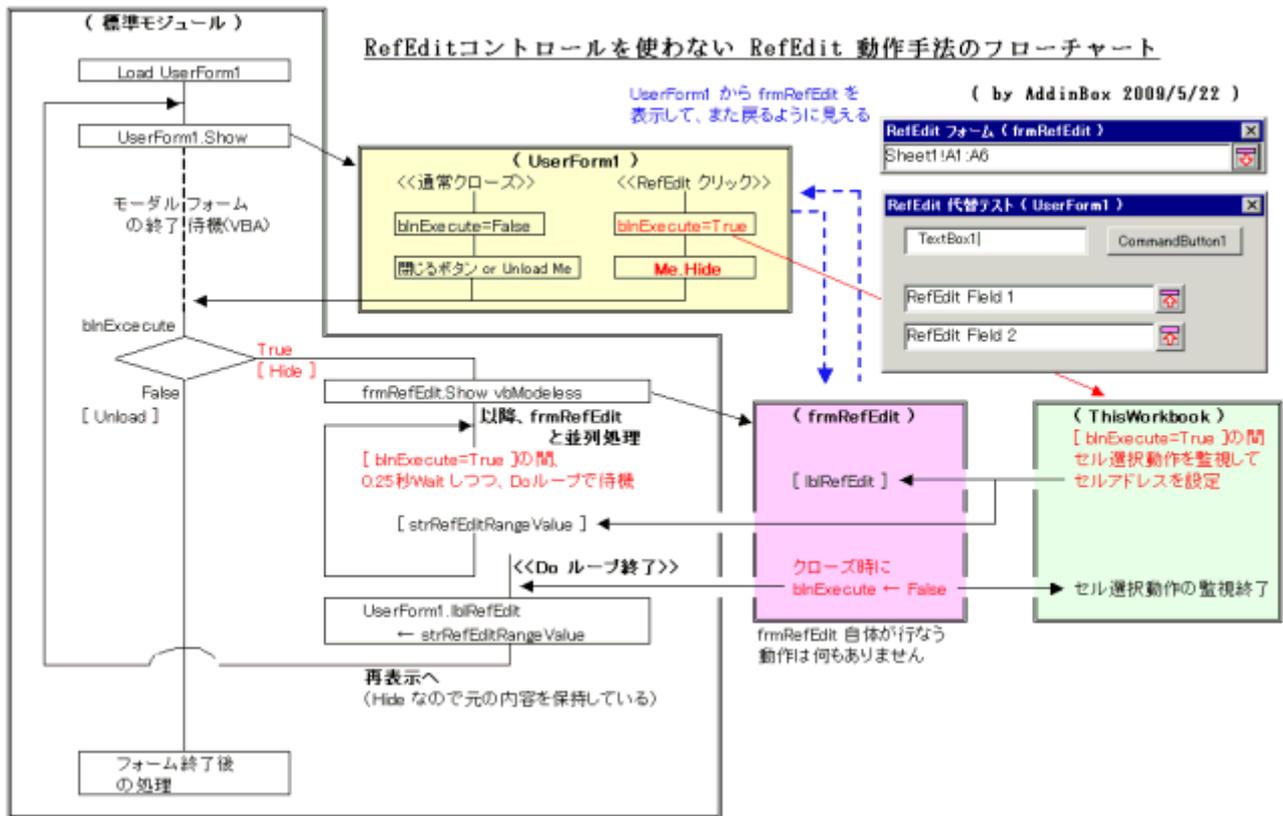


(上記は RefEdit コントロールではありません。Label & CommandButton の UserForm です。

RefEdit ボタンは、通常のコマンドボタンに画像を載せて作ってあります。)



【 詳細手順 フローチャート 】



上図のフル画像URL : http://addinbox.sakura.ne.jp/GifImg/GifDoc/Tips08_img07.gif

フローチャートに従って実行手順を順に説明します。

- (1) UserForm1にあるRefEditボタンをクリックすると、RefEdit実行フラグをオンにしてUserForm1は[Hide]で閉じます。
閉じるボタン[X] or マクロでのUnLoadの場合には、RefEdit実行フラグはオフになるようにします。
- (2) UserForm1.Showの次ステップでは、RefEdit実行フラグのオン/オフで分岐します。
- (3) RefEdit実行フラグがオフの場合は、通常クローズなので、後の処理に進みます。
- (4) RefEdit実行フラグがオンの場合は、RefEditフォームをモードレスで表示した後、実行フラグがオフになるまでループ待機します。
- (5) RefEdit実行フラグがオンになると、ThisWorkbook内でのセル範囲選択の監視が始まり、範囲選択するとRefEditフォームへそのセル範囲が渡されます。
- (6) RefEditフォームを閉じる（[X]ボタン or RefEditボタン）と実行フラグがオフとなり、ループ待機を終了します。
- (7) ループ後のステップで、セル範囲をUserForm1に設定して、UserForm1を再表示して戻ります（Hideで閉じているので、UserForm1の内容は保持されており、前回の内容で表示されます）。

～ 以上 ～

2-2 ソースコード

実際に利用する場合は【クラスモジュール版】を使用してください。

----- 標準モジュール -----

```
#If VBA7 And Win64 Then      ' 64bit 版対応
    Public Declare PtrSafe Sub Sleep Lib "kernel32" _
        (ByVal dwMilliseconds As Long)
#Else
    Public Declare Sub Sleep Lib "kernel32" _
        (ByVal dwMilliseconds As Long)
#End If

Public blnRefEditExecute As Boolean
Public strRefEditFieldName As String
Public strRefEditRangeValue As String
Public rngRefEditRangeObject As Range
'-----

Public Sub RefEditFormTest()
    blnRefEditExecute = False
    Load UserForm1
L1:
    UserForm1.Show

    If (blnRefEditExecute = True) Then
        ' [UserForm1] は Hide 状態
        strRefEditRangeValue = ""
        Load frmRefEdit
        With frmRefEdit
            .StartupPosition = 0
            .Top = UserForm1.Top
            .Left = UserForm1.Left
            .Caption = "RefEdit ( " & UserForm1.Caption & " )"
            .Show vbModeless
        End With
    End If
End Sub
```

'マウスマウスカーソルが砂時計のままとなる場合がある事への対処

ActiveSheet.Activate

DoEvents

Do While blnRefEditExecute 'RefEdit 実行中はループ継続

 Sleep 250 'Wait(0.25sec)

 DoEvents

Loop

UserForm1.Controls(strRefEditFieldName) _

 .Caption = strRefEditRangeValue

GoTo L1

End If

blnRefEditExecute = False

'~~ UserForm1 入力後の処理 ~~

End Sub

---- RefEdit を使う UserForm モジュール(UserForm1) ----

Private Sub cmdRefEdit1_Click()

 strRefEditFieldName = "lblRefEdit1"

 blnRefEditExecute = True

 Me.Hide

End Sub

'-----

Private Sub cmdRefEdit2_Click()

 strRefEditFieldName = "lblRefEdit2"

 blnRefEditExecute = True

 Me.Hide

End Sub

---- RefEdit 代替フォーム (frmRefEdit) ----

```
Private Sub cmdRefEdit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
,-----
```

```
Private Sub UserForm_QueryClose _
```

```
    (Cancel As Integer, CloseMode As Integer)
```

```
    blnRefEditExecute = False
```

```
End Sub
```

---- ThisWorkbook モジュール ----

```
Private Sub Workbook_SheetSelectionChange _
```

```
    (ByVal Sh As Object, ByVal Target As Range)
```

```
    If (blnRefEditExecute = True) Then
```

```
        If (Application.ReferenceStyle = xlA1) Then
```

```
            strRefEditRangeValue = Sh.Name & "!" _
```

```
                & Target.Address(False, False, xlA1)
```

```
        Else
```

```
            strRefEditRangeValue = Sh.Name & "!" _
```

```
                & Target.Address(True, True, xlR1C1)
```

```
        End If
```

```
        frmRefEdit.lblRefEdit.Caption = strRefEditRangeValue
```

```
    End If
```

```
End Sub
```

§ 3 RefEdit コントロール代替手法 — クラスモジュール版 —

3-1 基本動作

RefEdit コントロールを使わない上記手法の

- ・ ループ制御ロジック
- ・ RefEdit フォーム呼び出し時の RefEdit 実行フラグの設定処理

等をクラスモジュール(clsRangeSelection)にしてブラックボックス化しました (ThisWorkbook モジュールは使いません)。ループ制御などのマクロをコピーする等の手間も無くなり、非常に利用し易くなりました (実行マクロも視覚的に判り易くなりました)。

The image displays VBA code and the RefEdit control interface. The code is divided into two sections: '標準モジュール' (Standard Module) and 'UserForm1 モジュール' (UserForm1 Module).

```
'=== 標準モジュール ===
Public RefEditSupport As clsRangeSelection 'RefEdit制御クラス(Public)
Sub Test_RefEdit_1()
'RefEdit代替手法は排他運用で使用する
If IsRefEditActive(RefEditSupport, False) Then
    Beep
    Exit Sub
End If
'クラスインスタンス作成時の ActiveWorkbook が RefEdit 選択対象となる。
Set RefEditSupport = New clsRangeSelection
Load UserForm1 '[UserForm1.Show]は使用しない
'UserForm1表示とRefEdit処理の制御
RefEditSupport.Show UserForm1
Set RefEditSupport = Nothing
End Sub
```

```
'=== UserForm1 モジュール ===
Private Sub cmdRefEdit1_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit1, 0 '複数セル範囲可
    Me.Hide
End Sub
Private Sub cmdRefEdit2_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit2, 1 '単一セル範囲可
    Me.Hide
End Sub
Private Sub cmdRefEdit3_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit3, 2 '単一セルのみ
    Me.Hide
End Sub
Private Sub cmdMsgBox1_Click()
    Dim rngSelect As Range
    Set rngSelect = RefEditSupport.TargetRange(Me.lblRefEdit1)
    If (rngSelect Is Nothing) Then
        MsgBox "Nothing"
    Else
        MsgBox rngSelect.Parent.Parent.Name & vbCrLf & _
            rngSelect.Parent.Name & vbCrLf & _
            rngSelect.Address(False, False)
    End If
End Sub
```

The RefEdit control interface is shown in two states. The top state shows the control with a selected range (Sheet2!A1:A4,C1:C2,E1) and a message box. The bottom state shows the control with a selected range (Sheet2!A1:A4,C1:C2,E1) and a message box. Red and blue arrows indicate the flow of data between the code and the interface.

上図のフル画像URL : http://addinbox.sakura.ne.jp/GifImg/GifDoc/Tips08_img08.gif

このクラスには【Range オブジェクト】を取得するプロパティを用意してありますので、RefEdit コントロールでは必要だった「セルアドレス文字→Range オブジェクト」変換処理を行なう必要がありません。

また、RefEdit コントロールの場合、TextBox 部が修正可能である為、選択したセル範囲のアドレス文字列が手入力により壊される（変更される）可能性もあります（Evaluate による Range オブジェクト変換の際にエラーチェックが必須）。しかし、RefEdit 代替手法では、セルアドレス表示部は表示専用である為、セル選択した内容が保障されます。

さらに、RefEdit コントロールでは出来なかった「他ブックのセルの選択」が可能であり、加えて「単一セルのみ選択可」といった制限を設けることも可能です。

～ RefEdit 代替手法を利用する UserForm の表示手順（上図の解説） ～

- (1) clsRangeSelection クラスオブジェクトについて排他制御の確認を行なう(後述 [§3-2-2](#) 参照)。
- (2) clsRangeSelection クラスのインスタンスを作成します。この時点の ActiveWorkbook が RefEdit による選択対象ブック (Startupブック) となります ([Show メソッド](#)で[BookChange =True]を指定した場合は他のブックのセルも選択可能です)。
- (3) Load UserForm1 でユーザーフォームをロードする (ユーザーフォームの初期処理が必要な場合は、この後に記述するか、もしくは UserForm_Initialize に記述する)。
- (4) clsRangeSelection クラスの [Show メソッド](#)を実行して RefEdit 代替手法の制御ルーチンを起動する。ユーザーフォームは、この Show メソッド内から表示されます。
(ユーザーサイドで UserForm1.Show は使用しません)
- (5) RefEdit ボタンクリック時に、clsRangeSelection クラスの [SetRefEdit メソッド](#)を実行します。その後に【 Me.Hide 】 でユーザーフォームを閉じると RefEdit 代替フォームが表示されます。

[SetRefEdit メソッド](#)には、選択セルアドレスを表示する【ラベルコントロール】と、選択セルの制限条件 (単一セルのみ可 or 複数セル範囲が可 等) を指定します。

- (注) この条件設定機能は、Ver2.30 より [InitRefEdit メソッド](#) (UserForm_Initialize 内で実行) にて行なうようになりました。尚、[SetRefEdit メソッド](#)でも引き続き指定可能ですので、[InitRefEdit メソッド](#)を使わなくても問題なく動作します。

- (6) RefEdit 代替フォームを閉じると、ユーザーフォームを再表示して、指定のラベルコントロールに選択セルアドレスが設定されます。
- (7) Activate イベントと [ActivateInfo プロパティ](#) を使って【RefEdit からの復帰イベント】を構築できます (AfterUpdate イベントに相当)。
- (8) 通常クローズ (閉じるボタン または マクロの Unload) すると、[Show メソッド](#) を抜けるので、クラスオブジェクトを解放して終了してください。
- (9) 選択セルの Range オブジェクトは [TargetRange プロパティ](#) で取得できます。セルアドレス文字は [TargetAddr プロパティ](#) から取得します。
- (10) セル選択内容をクリアする場合は [Clear メソッド](#) を実行します (ラベル上のセルアドレス文字を消去し、[TargetRange プロパティ](#) の値を Nothing にします)。クリアボタン またはラベルのマウスイベント等で実行してください。

3-2 注意事項

3-2-1 一般事項

- (1) モードレスフォームを使用していますので、ユーザーフォームから「子フォーム」を起動する事には利用できません（モーダルフォーム内からモードレスフォームは表示不可）。標準モジュールからの起動に限ります。また、Excel97 では利用できません。
- (2) RefEdit 代替フォーム入力中にはシート切り替えが可能です。但し、シート切り替え直後には、「シート」が正しく取り込まれない場合があります。その場合には、RefEdit 代替フォーム上をクリックしてから再度セルを選択して下さい。

3-2-2 RefEdit 代替手法を利用する際に排他制御が必要な理由

RefEdit 代替手法を使用する場合には、`clsRangeSelection` クラスオブジェクトが稼働中か否かを判断してユーザーフォーム表示マクロの排他制御を行なってください ([サンプルマクロ参照](#))。

RefEdit 代替フォームは通常モードレスフォームです。したがって、セル選択操作だけではなく、

- ・メニュー（コマンドバー/リボン、右クリックメニュー）
- ・シート上に配置したボタン等

の操作も可能です。それ故に、ユーザーフォームの処理の最中であるにも係わらず、

- ・メニュー/ボタン等に登録されているマクロが呼出し可能な状態
- ・[メニュー > マクロの実行] からマクロを呼出す事が可能な状態
- ・上記の呼出し可能なマクロに、現在実行中の自分自身のマクロが含まれていれば、それも呼出し可能（この場合は[再帰呼出し]になります）

という問題が出て来ます（RefEdit コントロールの場合には、メニュー/ボタン等の操作はできません）。

この問題を回避する為に、RefEdit 代替手法を利用するプロジェクト（ブック）では、各プロシジャの先頭で排他チェックを行なう必要があります。

この排他チェックを行なう為に [IsRefEditActive 関数](#) を用意してあります。Type 引数を以下のように使い分けて判定します。

- (1) RefEdit 代替手法を利用する プロシジャーの場合

[Type = False] として RefEdit 代替手法の重複実行（`clsRangeSelection` クラスオブジェクトのバッティング）を回避します。

(2) 上記 (1) 以外の Sub プロシジャー で、メニュー登録されているプロシジャー、または、[マクロの実行]から呼出し可能なプロシジャーの場合

[Type = True] として、セル選択操作中 (RefEdit フォーム 表示中) にメニューから実行されないようにします。

(3) 上記 (1) (2) 以外の プロシジャーの場合

IsRefEditActive 関数による排他制御処理の組み込みは必要ありません。

この排他制御管理を行なう為に、

**clsRangeSelection クラスオブジェクト変数は
プロジェクト (ブック) 内に ただ1つだけ**

としてください。複数の ユーザーフォーム で RefEdit 代替手法を利用する場合、その1つのオブジェクト変数を共用して使用します。

(注) Excel2000～2003 では、

- ・『待機ループ (DoEvents あり)』の最中に
- ・[メニュー/マクロの実行] からマクロを実行する

と、【そのマクロが ” 2度 ” 連続して実行されてしまう】という不具合があります。メニュー or ボタン等に登録して実行されるものは大丈夫です。また、Excel2007/Excel2010 では、この不具合はありません (リボンシステムとして新たにメニュー系が作り直された為と思われます)。

したがって、排他制御により実行抑止する際には Beep による警告に留め、MsgBox 表示は行なわないで下さい。

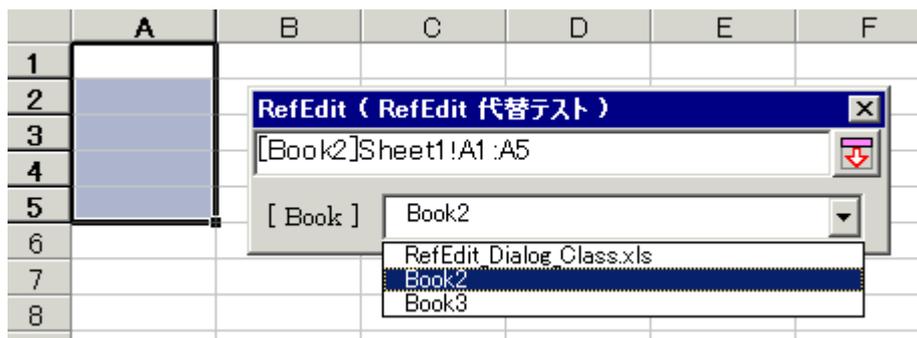
3-2-3 ブック切替機能とブック参照の解放

通常 ([Show メソッド](#)で BookChange を省略 or False) は、

Startup ブック : クラスオブジェクト生成時点の ActiveWorkbook

に対してのみセル選択操作が可能です。

[Show メソッド](#)で [BookChange = True] とすると、RefEdit フォーム上でブックを切り替えるコンボボックスが有効となり、オープン中のブックがリストに載ります。そのコンボボックスでブックを切り替えると、切替後のブックに対してセル選択が可能となります。なお、ブックのリストにはアドインブック・個人用マクロブック・テンプレートブックは含まれません。



このコンボボックスでブックを指定すると、そのブックが Active になります。RefEdit フォームを閉じてユーザーフォームへ戻る際には StartUp ブック（クラスオブジェクト生成時点の Active-Workbook）へ自動的に切り替わります。また、ブック切り替え機能を有効にした場合には、セルアドレス文字に「ブック名」が付きます（[TargetAddr プロパティ](#)、ユーザーフォームの RefEdit ラベルも同様です）。

ここで、ブック切替機能を有効にしている場合に注意を要する点がひとつあります。それは、

RefEdit でセル選択をしたブックをユーザーフォーム側で Close する

という場合です。例としては、

「データブック」が別個にあるとして、ユーザーフォームマクロから「データブック」を開き、RefEdit で取り込みセル範囲を指定し、その範囲のデータを取り込んだ後に「データブック」を閉じる。

というものが考えられます。

[Clear メソッド](#)、または [TargetRange プロパティ](#) に Nothing を代入しない限り、クラスオブジェクト内には、選択したセルのオブジェクト参照が残っています。このオブジェクト参照を残したままそのセルが属するブックを Close すると、俗に言う『幽霊プロジェクト（エクセル上からも、タスクバーからもブックが消えているのに、VBE 上には残っている）』の状態になります。

これを回避する為に、ブックを Close する前に、[ReleaseBookReference メソッド](#)を実行してください。

ReleaseBookReference メソッドは、指定したブックのセルを選択している RefEdit を探して一括して Clear（オブジェクト参照を解放します）します。ユーザーサイドで「どの RefEdit が、そのブックのセルを選択しているか」といった事を気にする必要はありません。一度のメソッドで対象となる RefEdit を全て Clear します。

(例) `Workbooks.Open “ABC.xls”`
:
～ `ABC.xls` に関する処理、`ABC.xls` のセルを `RefEdit` 選択 等 ～
:
`RefEditSupport.ReleaseBookReference Workbooks(“ABC.xls”)`
`Workbooks(“ABC.xls”).Close`

なお、ブック切替機能を使わない場合にはこの処理は必要ありません。また、ユーザーフォーム終了後に行なう「`クラスオブジェクト = Nothing`」により、クラス内の全てのオブジェクト参照は解放されます。

3-3 利用する際の準備

～ 利用ブックに コピー（インポート）するもの ～

- (1) 下記リンク先の Excel ブックをダウンロードします。
(クラスモジュールとサンプルコードが収録されています)
【 クラスモジュール & サンプルブック 】 (Ver. 2.40 2011/7/1)
http://addinbox.sakura.ne.jp/Download/RefEdit_Support_Class.xls
- (2) サンプルブックから、クラスモジュール : clsRangeSelection をインポートする。
- (3) サンプルブックから、標準モジュール : mdlRangeSelection をインポートする。
- (4) サンプルブックから、ユーザーフォームモジュール : frmRefEdit をインポートする。
(※ frmRefEdit のフォーム名は変更できません)
- (5) サンプルブックの UserForm1 上の「RefEdit ボタンと RefEdit ラベル」を、利用するユーザーフォームにコピー&ペーストする。ボタンをコピーすればアイコン画像も一緒にコピーされます。

Excel の RefEdit アイコン画像に変更したい場合には、下記リンク先のファイルをダウンロードし、CommandButton の Picture プロパティでアイコン画像を差し替えてください。

【 アイコン画像ファイル 】 

http://addinbox.sakura.ne.jp/Download/RefEdit_Icon.zip

～ アドイン運用など、他ブックから参照設定して利用する場合 ～

- (1) プロパティウィンドウで、clsRangeSelection クラスモジュールの [Instancing] プロパティを Private → **PublicNotCreatable** に変更する。
- (2) クラスを利用するブックでは、クラスオブジェクト作成の際、
Set RefEditSupport = **New clsRangeSelection**
の代わりにインスタンス作成関数を使います。
Set RefEditSupport = [CreateRangeSelectClass](#) ()

3-4 プロパティ/メソッド 一覧

下記で[object]の部分には、**clsRangeSelection** クラスを定義したオブジェクト変数を記述します。

3-4-1 定義

標準 モジュールの宣言セクション (モジュールの先頭) で、下記のように定義します。

```
Public object As clsRangeSelection
```

このクラスオブジェクトは、必ず **Public** で宣言してください。

なお、宣言時に **New** キーワード を付けないでください。

```
OK Public object As clsRangeSelection
```

```
NG Public object As New clsRangeSelection
```

(object)

任意の名前を指定して下さい。

RefEdit 代替手法を利用する ユーザーフォーム を表示する前に、下記のようにしてクラスのインスタンスを作成します。

```
Set object = New clsRangeSelection
```

※ 複数のユーザーフォーム でこの機能を使う場合、個々のユーザーフォーム ごとに クラスオブジェクトを用意せず、1プロジェクト (ワークブック) にクラスオブジェクトは1つのみとし、複数のユーザーフォーム で共有するようにしてください (排他制御の判定の為)。

※ 通常は、**StartUp** ブック (クラスオブジェクト作成時の **ActiveWorkbook**) が RefEdit 代替フォームで選択可能なブックとなります。もし、ユーザーフォームのマクロ で アクティブブック を他のブック に切り替えたとしても、その ブックに対しては機能しません (選択セルは取り込まれません)。なお、シート は切り替え可能です。

※ 上記 StartUp ブック 以外のブックのセル選択を行ないたい場合には、[Show](#) [メソッド](#)の[BookChange]に **True** を指定します。

※ アドイン運用など、参照設定でクラスを利用している場合は、**New** キーワードを使わずに下記のようにします。

```
Set object = CreateRangeSelectClass ( )
```

(CreateRangeSelectClass 関数は、mdlRangeSelection モジュール に用意してあります)

3-4-2 Show メソッド

object .**Show** MsForms ユーザーフォーム オブジェクト , [BookChange]

RefEdit 代替手法の制御ルーチンを起動するメソッドです。ユーザーフォームは、このメソッド内から表示されます（ユーザーサイドで UserForm1.Show 等とする必要はありません）。尚、この Show メソッドを実行する前に ユーザーフォームを Load してください。

(例)

```
Set RefEditSupport = New clsRangeSelection
Load UserForm1
' ~ UserForm1 の初期処理が必要な場合は、この場所、
'     ~
'     もしくは UserForm_Initialize に記述する ~
RefEditSupport.Show UserForm1 , False
Set RefEditSupport = Nothing
```

(MsForms ユーザーフォーム オブジェクト)

RefEdit 代替手法を利用する ユーザーフォームオブジェクトを指定します。

(BookChange)

Startup ブック（クラスオブジェクト生成時点の ActiveWorkbook ）以外のブックへ切り替えを行なってセル選択を行なうか否かを指定します。

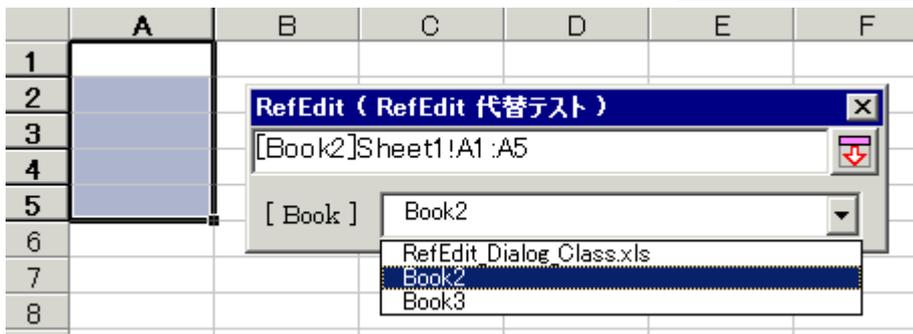
省略が可能です。

False : Startup ブックに対してのみセル選択可能です。（既定値）

True : Startup ブック以外のブックに切り替えてセル選択が可能です。

ブック切替機能を使用する場合、以下の点が変わります。

- RefEdit フォームに『ブック切替コンボボックス』が表示されます。このコンボボックスでブックを切り替えてからセルを選択します。
- セルアドレス表示にブック名が加わります（[TargetAddr プロパティ](#)も同様）。



3-4-3 InitRefEdit メソッド

object . **InitRefEdit** RefEditLabel , RangeType

RefEdit 対象コントロール（選択したセルアドレスを表示するラベル）の初期設定を行なうメソッドです。UserForm_Initialize 等で実行します。

[単一セル]のみ許すのか、[セル範囲]も可とするのかという制限を指定できます。

(例)

```
Private Sub UserForm_Initialize ()
    With RefEditSupport
        .InitRefEdit Me.lblRefEdit1 , 0        ' 複数セル範囲可
        .InitRefEdit Me.lblRefEdit2 , 2        ' 単一セルのみ可
    End With
End Sub
```

(RefEditLabel)

RefEdit 対象コントロールオブジェクト（選択したセルアドレスを表示するラベル）を指定します。ラベルのみ指定できます。テキストボックスなどは指定できません。

(RangeType)

セルを選択する際の制限を指定します。

- 0 : 制限なし (複数セル範囲 or 単一セル範囲 or 単一セル)
- 1 : 単一セル範囲 or 単一セル が可能
- 2 : 単一セル のみ可能
- 3 : 複数の単一セル が可能

なお、結合セルは1セル（結合セルの先頭セル）として扱います。

InitRefEdit メソッドで指定した RangeType 属性は [SetRefEdit メソッド](#) で変更する事が可能です。InitRefEdit/SetRefEdit メソッドにて設定した RangeType 属性は [TargetRange プロパティ](#) でセル範囲を設定する際の妥当性チェックにも使用されます（[単一セルのみ可]としているコントロールに対してセル範囲を設定しようとした場合など。この場合にはセル範囲の先頭セルのみが設定されます）。

3-4-4 SetRefEdit メソッド

object .SetRefEdit RefEditLabel , [RangeType]

RefEdit 対象コントロール（選択したセルアドレスを表示するラベル）を指定するメソッドです。このメソッドの後で [Me.Hide] でフォームを閉じると、RefEdit 代替フォームが表示されます。

[単一セル] のみ許すのか、[セル範囲] も可とするのかという制限を指定できます。

(例)

```
Private Sub cmdRefEdit3_Click ()
    RefEditSupport.SetRefEdit Me.lblRefEdit3 , 2 ' 単一セルのみ
    Me.Hide
End Sub
```

セルアドレスは Excel のオプション設定にしたがって

"Sheet2!A1:B5" (A1 形式) or "Sheet2!R1C1:R5C2" (R1C1 形式)

の形式で表示されます。なお、ブック切替機能を有効にしている場合にはセルアドレスにブック名が付きます。

"[Book1.xls]Sheet2!A1:B5" (A1 形式) or "[Book1.xls]Sheet2!R1C1:R5C2" (R1C1 形式)

選択したセルが結合セルの場合には「結合セルの先頭セル」が取得されます（結合セルを含むセル範囲の場合にはセル範囲全体が取得されます）。

(RefEditLabel)

RefEdit 対象コントロールオブジェクト（選択したセルアドレスを表示するラベル）を指定します。ラベルのみ指定できます。テキストボックスなどは指定できません。

(RangeType)

セルを選択する際の制限を指定します（省略が可能です）。実行の都度、変更しても構いません。

省略 : 前回の設定が継続 (Ver2.30)。初回設定時は [0] と解釈します。

- 0 : 制限なし (複数セル範囲 or 単一セル範囲 or 単一セル)
- 1 : 単一セル範囲 or 単一セル が可能
- 2 : 単一セル のみ可能
- 3 : 複数の単一セル が可能

なお、結合セルは 1 セル（結合セルの先頭セル）として扱います。

[InitRefEdit/SetRefEdit](#) を一度も実行していない RefEdit 対象コントロールに対して、RangeType を省略した場合（初回設定）には [0 : 制限なし] となります。

3-4-5 Clear メソッド

object .**Clear** RefEditLabel

セル選択情報をクリアするメソッドです。クリアボタン押下 または RefEdit 対象コントロール (ラベル) のマウスイベント等で実行してください。

指定した RefEdit 対象コントロール (選択したセルアドレスを表示するラベル) 上に表示されているセルアドレスを消去します。また、このメソッド実行後、指定した RefEdit 対象コントロールに対する [TargetRange プロパティ](#) は Nothing を返します。

なお、TargetRange プロパティに Nothing を代入しても同じ効果が得られます。

(例)

```
Private Sub lblRefEdit1_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, _
                                   ByVal X As Single, ByVal Y As Single)
    If ( Button = 2 ) And ( Shift = 4 ) Then
        ' [ Alt + 右 Click ] 操作でクリアする
        RefEditSupport.Clear Me.lblRefEdit1
    End If
End Sub
```

(RefEditLabel)

RefEdit 対象コントロールオブジェクト (選択したセルアドレスを表示する ラベル) を指定します。ラベル のみ指定できます。テキストボックスなどは指定できません。

3-4-6 Terminate メソッド

object .**Terminate**

RefEdit 利用フォーム の Terminate イベントで必ず実行してください。

クラスオブジェクト内から UserForm (コントロール) 参照を完全に削除して、Unload を確実にこなえるようにする為に必要です。

3-4-7 EndRefEdit メソッド

object .**EndRefEdit**

[frmRefEdit : QueryClose] で RefEdit 終了通知(ループ停止指示) に使用しています。
【ユーザーサイド】では使用しません。

3-4-8 BookChange メソッド

object .**BookChange** Index

[frmRefEdit : cmbBook_Change] で 対象ブック切替に使用しています。
【ユーザーサイド】では使用しません。

(Index)

ブック切替コンボボックスの ListIndex 値を指定します。

3-4-9 ReleaseBookReference メソッド

object .**ReleaseBookReference** Book

ブック切替機能を使用している場合に使用します (ブック切替機能を使用していない場合は無効です)。これから Close しようとしているブックのセルを選択している RefEdit 情報を一括してクリアして、そのブックへのオブジェクト参照を解放する為に使用します。

詳細は「[§ 3-2-3 ブック切替機能とブック参照の解放](#)」を参照してください。

(Book)

Workbook オブジェクトを指定します。

Startup ブック (クラスオブジェクト生成時点の ActiveWorkbook) を指定した場合は無効です。

3-4-10 StartUpBook プロパティ

object .**StartUpBook** () [= Workbook]

StartUp ブック（クラスオブジェクト生成時の ActiveWorkbook）のオブジェクトを取得します。
値の取得のみ可能です。

（ 値の取得 ）

StartUp ブック（クラスオブジェクト生成時の ActiveWorkbook）のオブジェクトを返します。

3-4-11 TargetRange プロパティ

object .**TargetRange** (RefEditLabel) [= Range]

RefEdit ボタンから選択したセルの Range オブジェクトを取得します。逆に RefEdit ボタンへ Range オブジェクト を設定する事もできます（RefEdit ラベル上への セルアドレス文字の設定も同時に行なわれます）。

値の取得/設定が可能です。

このプロパティを使うと、RefEdit コントロールの場合に必要な

「セルアドレス文字 → Range オブジェクト」の変換処理（Evaluate 等による変換）
が必要ありません。

Nothing を代入する操作は [Clear メソッド](#) と同じ効果です（RefEdit ラベルのセルアドレス文字は消去されます）。

（ RefEditLabel ）

RefEdit 対象コントロールオブジェクト（選択したセルアドレスを表示する ラベル） を指定
します。ラベル のみ指定できます。テキストボックスなどは指定できません。

（ 値の取得 ）

指定の RefEdit 対象コントロールについて、セル選択済みの場合は、その Range オブジェクト
を返します。セル未選択の場合は Nothing を返します。

初期値（セル選択 or TargetRange（値の設定）を一度も行なっていない状態）は Nothing です。

選択したセルが結合セルの場合には「結合セルの先頭セル」が取得されます（結合セルを含むセル範囲の場合にはセル範囲全体が取得されます）。

（ 値の設定 ）

指定の RefEdit 対象コントロールに対して、Range オブジェクト を設定できます（その Range オブジェクト は、この TargetRange プロパティで再び取得できます）。Range オブジェクトのセルアドレス文字が RefEdit ラベル に設定されます（Nothing を設定した場合は、消去になります）。

[InitRefEdit/SetRefEdit](#)にて設定した RangeType に合わないセル範囲を渡した場合には、RangeType に沿うように渡されたセル範囲を整形してから設定されます。

例： RangeType=2（単一セルのみ可）と指定されている RefEdit 対象コントロールに「セル範囲」を渡した場合には、そのセル範囲の先頭セルだけが設定されます。

InitRefEdit/SetRefEdit を一度も実行していない RefEdit 対象コントロールに対して、[TargetRange プロパティ](#)でセル範囲（or 単一セル）を設定する場合には、その RefEdit 対象コントロールの Range Type は[0：制限なし]と見做されます。

ブック切替機能を使わない場合、その Range オブジェクトが Startup ブック（クラスオブジェクト生成時の ActiveWorkbook）と異なる時は無効です。

セルアドレス文字のフォーマット（A1 形式，R1C1 形式）は Excel のオプションの設定に従います。

A1 形式 : "Sheet2!B2:C5"
R1C1 形式 : "Sheet2!R2C2:R5C3"

ブック切替機能を有効にしている場合には、セルアドレスにブック名が付きます。

A1 形式 : "[Book1.xls]Sheet2!B2:C5"
R1C1 形式 : "[Book1.xls]Sheet2!R2C2:R5C3"

3 - 4 - 1 2 **TargetBook** プロパティ

object .**TargetBook** ([RefEditLabel](#)) [= [Workbook](#)]

RefEdit ボタンから選択したセルの Workbook オブジェクトを取得します。
値の取得のみ可能です。

(RefEditLabel)

RefEdit 対象コントロールオブジェクト (選択したセルアドレスを表示するラベル) を指定します。
ラベルのみ指定できます。テキストボックスなどは指定できません。

(値の取得)

指定の RefEdit 対象コントロールについて、セル選択済みの場合は、そのセルの Workbook
オブジェクト を返します。セル未選択の場合は Nothing を返します。

初期値 (セル選択 or [TargetRange](#) (値の設定) を一度も行なっていない状態) は Nothing です。

3-4-13 TargetSheet プロパティ

```
object .TargetSheet ( RefEditLabel ) [ = Worksheet ]
```

RefEdit ボタンから選択したセルの Worksheet オブジェクトを取得します。
値の取得のみ可能です。

(RefEditLabel)

RefEdit 対象コントロールオブジェクト (選択したセルアドレスを表示するラベル) を指定します。
ラベルのみ指定できます。テキストボックスなどは指定できません。

(値の取得)

指定の RefEdit 対象コントロールについて、セル選択済みの場合は、そのセルの Worksheet
オブジェクト を返します。セル未選択の場合は Nothing を返します。

初期値 (セル選択 or [TargetRange](#) (値の設定) を一度も行なっていない状態) は Nothing です。

3-4-14 TargetAddr プロパティ

object .**TargetAddr** (RefEditLabel , [ReferenceStyle] , [CellOnly]) [= String]

RefEdit ボタンから選択したセルの セルアドレス文字 を取得します。
値の取得のみ可能です。

選択したセルが結合セルの場合には「結合セルの先頭セル」が取得されます（結合セルを含むセル範囲の場合にはセル範囲全体が取得されます）。

(RefEditLabel)

RefEdit 対象コントロールオブジェクト（選択したセルアドレスを表示するラベル）を指定します。
ラベル のみ指定できます。テキストボックスなどは指定できません。

(ReferenceStyle)

セルアドレスの編集形式（A1 形式/R1C1 形式）を **x1A1** または **x1R1C1** で指定します。
省略可能です。省略した場合は Excel のオプション設定に従います。

(CellOnly)

セルアドレスにブック名/シート名を含めるか否かを指定します。
省略可能です。

False : ブック名/シート名を含めます。既定値。

True : ブック名/シート名を含めずに、セルアドレスのみの編集。

(値の取得)

指定の RefEdit 対象コントロールについて、セル選択済みの場合は、そのセルのアドレスを文字列で返します。セル未選択の場合は 空文字("") を返します。

A1 形式 : "Sheet2!B2:C5"

R1C1 形式 : "Sheet2!R2C2:R5C3"

ブック切替機能を有効にしている場合には、セルアドレスにブック名が付きます。

A1 形式 : "[Book1.xls]Sheet2!B2:C5"

R1C1 形式 : "[Book1.xls]Sheet2!R2C2:R5C3"

CellOnly=True とした場合には、ブック名/シート名は付きません（セルアドレスのみ）。

初期値（セル選択 or [TargetRange](#)（値の設定）を一度も行なっていない状態）は 空文字 です。

3-4-15 ActivateInfo プロパティ

```
object .ActivateInfo [ = String ]
```

UserForm_Activate イベント内で使用します。

このプロパティには Activate イベントの発生理由が文字列で返ります。

値の取得のみ可能です。

※ この ActivateInfo プロパティは、値を一度取得すると、値が空文字("")にリセットされます。したがって、**Activate イベントの先頭で変数に取り込み、その変数で処理を行ってください。**

- a) UserForm_Initialize イベント直後の Activate イベントでは **"Initialize"** の文字列が返ります。
- b) RefEdit 入力から復帰する際にも、ユーザーフォームの再表示によって Activate イベントが発生します。この場合には、**RefEdit ラベルのオブジェクト名**が返ります。
- c) 上記以外の Activate イベントでは **空文字("")** が返ります。
 - ・ 子フォームを表示し、その子フォームを閉じて戻って来た時 (Excel2000(SR 無)のみ)
 - ・ RefEdit 処理以外で Hide した後に再表示した時

"Initialize" / 空文字("") 以外の場合が、**RefEdit 入力の AfterUpdate イベント** に相当します。RefEdit 項目が複数ある場合は、その項目名で分岐させてください。

(値の取得)

"Initialize" or RefEdit ラベル項目名 or 空文字("") が返ります。

3-4-16 RefEditActive プロパティ

```
object .RefEditActive ( ) [ = Boolean ]
```

RefEdit フォーム表示中 (セル選択操作中) か否かの状態を取得します。

値の取得のみ可能です。

[IsRefEditActive] 関数内で使用しています。

【ユーザーサイド】 では使用しません。

(値の取得)

True : RefEdit フォーム表示中 (セル選択操作中)

False : RefEdit フォームは表示されていない (利用元の ユーザーフォーム を表示中)

3-4-17 IsRefEditActive 関数 (mdlRangeSelection に収録)

IsRefEditActive (object , [Type]) [= Boolean]

RefEdit 代替手法によるユーザーフォーム処理が実行中か否かの状態を返します。

(clsRangeSelection クラスオブジェクトのインスタンス作成済か否か)

[Type = True] とすると、範囲を狭めて『RefEdit フォーム表示中(セル選択操作中)か否か』の状態を返します。

この関数の使用目的については前述 『§3-2-2 RefEdit 代替手法を利用する際に排他制御が必要な理由』 を参照。

Type 引数の使い分けは以下のようにしてください。

(1) RefEdit 代替手法を利用する プロシジャー

[Type = False] として RefEdit 代替手法の重複実行 (clsRangeSelection クラスオブジェクトのバッティング) を回避します。

(2) 上記 (1) 以外の Sub プロシジャー で、メニュー登録されているもの

／[マクロの実行] から呼出し可能なもの

[Type = True] として、セル選択操作中 (RefEdit フォーム 表示中) にメニュー から実行されないようにします。

(3) 上記 (1) (2) 以外の プロシジャー

IsRefEditActive 関数による排他制御処理の組み込みは必要ありません。

(object)

clsRangeSelection クラスのオブジェクト変数 を指定します。

この排他制御管理を行なう為に、clsRangeSelection クラスオブジェクト変数はプロジェクト (ブック) 内にただ1つだけとしてください。複数の ユーザーフォーム で RefEdit 代替手法を利用する場合、その1つの オブジェクト変数を共用して使用するようにしてください。

(Type)

省略 or False : RefEdit 代替手法を利用する プロシジャール が実行中か否かの状態
True : RefEdit フォーム表示中(セル選択操作中) か否かの状態

(返却値)

Type = False の場合

True : RefEdit 代替手法を利用する プロシジャール が実行中
False : 上記以外 (未実行状態)

Type = True の場合

True : RefEdit フォーム表示中 (セル選択操作中)
False : 上記以外

- RefEdit 代替手法を利用するプロシジャール が実行中ではない
- RefEdit フォームは表示されていない (利用元のユーザーフォームを表示中)

3-4-18 CreateRangeSelectClass 関数 (mdlRangeSelection に収録)

CreateRangeSelectClass () [= clsRangeSelection オブジェクト]

VBAでは外部のクラスのインスタンス (オブジェクト) を New キーワードによって生成する事ができませんので、clsRangeSelection クラスをアドイン運用する場合には、New キーワードの代わりに、この関数によって、インスタンス (オブジェクト) を生成します。

```
Set object = CreateRangeselectClass ( )
```

RefEdit 代替手法を利用するワークブック内に clsRangeSelection クラスモジュールを収録して使用する場合には、この関数は必要ありません。

補: クラスをアドイン運用する場合には、clsRangeSelection クラスモジュールの [Instancing] プロパティを【 PublicNotCreatable 】とする必要があります。

(返却値)

clsRangeSelection クラスの新規インスタンス (オブジェクト) を作成して返します。

3-5 利 用 例

'==== 標準 モジュール ====

' RefEdit 制御クラスオブジェクト(Public),プロジェクト内で共有して使用する

Public RefEditSupport As clsRangeSelection

'-----

Sub Test_RefEdit_1()

'=====

' RefEdit 入力中(モードレス)はメニュー操作も可能になります。その為、

' • [メニュー/マクロの実行]から[Public]マクロの実行

' • コマンドメニュー/右クリックメニューに登録されたマクロの実行

' が可能な状態になっています。そこで、安全性の面から上記のマクロ等

' が起動しないように、以下のように排他制御を行なってください。

If IsRefEditActive (RefEditSupport , **False**) Then

 Beep

 Exit Sub

End If

'=====

' StartUp ブック : [クラスオブジェクト作成時の ActiveWorkbook]

Set RefEditSupport = New clsRangeSelection

Load UserForm1 ' [UserForm1.Show]は使用しない

' ~ UserForm1 の初期処理が必要な場合は、この場所か、UserForm_Initialize に記述する ~

With UserForm1

 .Caption = "RefEdit 代替テスト (ブック切替 無) "

 .cmdBook1Open.Enabled = **False**

 .cmdBook20Open.Enabled = **False**

End With

' UserForm 表示と RefEdit 処理の制御

' [BookChange=False] : StartUp ブックのみ RefEdit でセル選択可

RefEditSupport.Show UserForm1, **False**

Set RefEditSupport = **Nothing**

End Sub

'-----

Sub Test_RefEdit_1_BookChange()

If IsRefEditActive(RefEditSupport, **False**) Then

 Beep

 Exit Sub

End If

Set RefEditSupport = New clsRangeSelection

```

Load UserForm1
With UserForm1
    .Caption = "RefEdit 代替テスト (ブック切替 有) "
    .cmdBook10pen.Enabled = True
    .cmdBook20pen.Enabled = True
End With

' [BookChange=True] : StartUp ブック以外も RefEdit でセル選択可
RefEditSupport.Show UserForm1, True

Set RefEditSupport = Nothing
End Sub

,-----
' RefEditSupport を使用するマクロは RefEditSupport の有無で判断
,
' Sub Test_RefEdit_2()
'   If IsRefEditActive(RefEditSupport, False) Then
'       Beep
'       Exit Sub
'   End If
,
'   Set RefEditSupport = New clsRangeSelection
'   Load UserForm2
'   RefEditSupport.Show UserForm2, False
'   Set RefEditSupport = Nothing
'End Sub

,-----
' RefEditSupport を使用しないマクロは RefEdit フォームの入力中か否かで判断
,
' Sub Test_Other ()
'   If IsRefEditActive(RefEditSupport, True) Then
'       Beep
'       Exit Sub
'   End If
,
'   UserForm3.Show
'End Sub

,-----
' ※ アドイン運用など、別ブックから参照設定して利用する場合 ※
'   (1) clsRangeSelection の Instancing プロパティを"PublicNotCreatable"にする。
'   (2) [ Set RefEditSupport = New clsRangeSelection ] の代わりに
'       インスタンス作成関数を使って
'       Set RefEditSupport = CreateRangeSelectClass
'       としてインスタンスを作成する。

```

```

'==== UserForm1 モジュール ====

Private Wkbk1 As Workbook 'New Book[1]
Private Wkbk2 As Workbook 'New Book[2]

Private Sub UserForm_Initialize()
' [InitRefEdit] メソッド
' arg1: 呼出元 Label (選択セル範囲のアドレス文字を受取ります)
' arg2: 0(複数セル範囲可, 既定値), 1(単一セル範囲可), 2(単一セルのみ), 3(1つ以上の単一セル)

With RefEditSupport
. InitRefEdit Me.lblRefEdit1, 0 '複数セル範囲可
. InitRefEdit Me.lblRefEdit2, 1 '単一セル範囲可
. InitRefEdit Me.lblRefEdit3, 2 '単一セルのみ
. InitRefEdit Me.lblRefEdit4, 3 '1つ以上の単一セル
End With
End Sub
'-----

Private Sub UserForm_Activate()
Dim strActivateInfo As String
' ActivateInfo プロパティの取得は1回のみ有効です (取得後は空文字にリセットされる)。
' 直接参照せずに変数に取り出して利用します。
strActivateInfo = RefEditSupport.ActivateInfo

Select Case strActivateInfo
Case ""
' [RefEdit 呼出]以外で Hide した後に再表示した場合は空文字です。
' その際の Activate イベント処理は、ここに記述します。

Case "Initialize"
' Initialize イベント直後の Activate イベント処理は、ここに記述します。

Case Else
' RefEdit からの復帰イベント処理は、ここに記述します。
' (AfterUpdate イベントの代替になります)
' [ActiveInfo]には RefEdit 代替 Label 名が入っているので、
' 複数の RefEdit 使用時にはラベル名で分岐させます。

lblInfo.Caption = strActivateInfo & " から復帰"
End Select
End Sub
'-----

Private Sub UserForm_Terminate()
RefEditSupport.Terminate '※Terminate イベントで必ず実行する
End Sub
'-----

```

```

' ===== RefEdit 呼び出し =====
' 呼出元の RefEdit 代替 Label を設定して Hide すると、RefEdit フォームが呼び出されます。
' [SetRefEdit] メソッド
'   arg1: 呼出元 Label (選択セル範囲のアドレス文字を受取ります)
'   arg2: 0 (複数セル範囲可), 1 (単一セル範囲可), 2 (単一セルのみ), 3 (1 つ以上の単一セル)
'   省略時は [InitRefEdit] メソッドでの設定値で実行される。
'   [SetRefEdit] メソッドにて [RangeType] の変更は可能。

```

```

Private Sub cmdRefEdit1_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit1 '省略により [0] 複数セル範囲可
    Me.Hide
End Sub

```

```

' -----
Private Sub cmdRefEdit2_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit2 '省略により [1] 単一セル範囲可
    Me.Hide
End Sub

```

```

' -----
Private Sub cmdRefEdit3_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit3 '省略により [2] 単一セルのみ
    Me.Hide
End Sub

```

```

' -----
Private Sub cmdRefEdit4_Click()
    RefEditSupport.SetRefEdit Me.lblRefEdit4 '省略により [3] 複数の単一セル可
    Me.Hide
End Sub

```

```

' 入力内容をクリアする場合は、クリアボタン もしくは ラベルの
' マウスイベント等で Clear メソッドを実行します。

```

```

Private Sub lblRefEdit1_MouseDown(ByVal Button As Integer, _
                                   ByVal Shift As Integer, _
                                   ByVal X As Single, ByVal Y As Single)
    If (Button = 2) And (Shift = 4) Then
        '[Alt+右 Click] 操作でクリアする
        RefEditSupport.Clear Me.lblRefEdit1

        ' 下記でも同じクリア操作になります
        '[ RefEditSupport.TargetRange(Me.lblRefEdit1) = Nothing ]
    End If
End Sub

```

```

' -----
Private Sub lblRefEdit2_MouseDown(ByVal Button As Integer, _
                                   ByVal Shift As Integer, _
                                   ByVal X As Single, ByVal Y As Single)
    If (Button = 2) And (Shift = 4) Then
        RefEditSupport.Clear Me.lblRefEdit2
    End If
End Sub

```

```

', -----
Private Sub lblRefEdit3_MouseDown(ByVal Button As Integer, _
                                   ByVal Shift As Integer, _
                                   ByVal X As Single, ByVal Y As Single)
    If (Button = 2) And (Shift = 4) Then
        RefEditSupport.Clear Me.lblRefEdit3
    End If
End Sub
', -----

Private Sub lblRefEdit4_MouseDown(ByVal Button As Integer, _
                                   ByVal Shift As Integer, _
                                   ByVal X As Single, ByVal Y As Single)
    If (Button = 2) And (Shift = 4) Then
        RefEditSupport.Clear Me.lblRefEdit4
    End If
End Sub
', -----

' RefEdit 入力後には[TargetRange]プロパティで Range オブジェクトを取得できます。
' [TagetRange] プロパティ
'   arg1: 呼出元 Label を指定します

Private Sub cmdMsgBox1_Click()
Dim rngSelect As Range
    Set rngSelect = RefEditSupport.TargetRange(Me.lblRefEdit1)
    If (rngSelect Is Nothing) Then
        MsgBox "Nothing"
    Else
        MsgBox rngSelect.Parent.Parent.Name & vbCrLf & _
               rngSelect.Parent.Name & vbCrLf & _
               rngSelect.Address(False, False)

'   TargetBook/TargetSheet プロパティでも Book/Sheet オブジェクトを取得できます。
'   TargetAddr プロパティでも セルアドレス文字を取得できます。
'   MsgBox RefEditSupport.TargetBook(Me.lblRefEdit1).Name & vbCrLf & _
'           RefEditSupport.TargetSheet(Me.lblRefEdit1).Name & vbCrLf & _
'           RefEditSupport.TargetAddr(Me.lblRefEdit1, x1A1, True)

    End If
End Sub

```

```

', -----
Private Sub cmdMsgBox2_Click()
Dim rngSelect As Range
Set rngSelect = RefEditSupport.TargetRange(Me.lblRefEdit2)
If (rngSelect Is Nothing) Then
MsgBox "Nothing"
Else
MsgBox rngSelect.Parent.Parent.Name & vbCrLf & _
rngSelect.Parent.Name & vbCrLf & _
rngSelect.Address(False, False)
End If
End Sub
', -----

```

```

Private Sub cmdMsgBox3_Click()
Dim rngSelect As Range
Set rngSelect = RefEditSupport.TargetRange(Me.lblRefEdit3)
If (rngSelect Is Nothing) Then
MsgBox "Nothing"
Else
MsgBox rngSelect.Parent.Parent.Name & vbCrLf & _
rngSelect.Parent.Name & vbCrLf & _
rngSelect.Address(False, False)
End If
End Sub
', -----

```

```

Private Sub cmdMsgBox4_Click()
Dim rngSelect As Range
Set rngSelect = RefEditSupport.TargetRange(Me.lblRefEdit4)
If (rngSelect Is Nothing) Then
MsgBox "Nothing"
Else
MsgBox rngSelect.Parent.Parent.Name & vbCrLf & _
rngSelect.Parent.Name & vbCrLf & _
rngSelect.Address(False, False)
End If
End Sub

```

```

-----
Private Sub cmdBook1Open_Click()
    If (Wkbk1 Is Nothing) Then
        'Book[1] Open
        Set Wkbk1 = Workbooks.Add

        cmdBook1Open.Caption = "Close"
        cmdBook1Open.ForeColor = vbRed
    Else
        'Book[1] Close , Close する前に Book[1]のセル選択を全て解放する
        RefEditSupport.ReleaseBookReference Wkbk1
        Wkbk1.Close SaveChanges:=False
        Set Wkbk1 = Nothing

        cmdBook1Open.Caption = "Open"
        cmdBook1Open.ForeColor = vbBlack
    End If
End Sub

```

```

-----
Private Sub cmdBook2Open_Click()
    If (Wkbk2 Is Nothing) Then
        'Book[2] Open
        Set Wkbk2 = Workbooks.Add

        cmdBook2Open.Caption = "Close"
        cmdBook2Open.ForeColor = vbRed
    Else
        'Book[2] Close , Close する前に Book[2]のセル選択を全て解放する
        RefEditSupport.ReleaseBookReference Wkbk2
        Wkbk2.Close SaveChanges:=False
        Set Wkbk2 = Nothing

        cmdBook2Open.Caption = "Open"
        cmdBook2Open.ForeColor = vbBlack
    End If
End Sub

```

3-6 F A Q

Q 1. VBEの実行ボタンで UserForm を表示するとエラーになります。

A 1. clsRangeSelection を使用している UserForm を、VBEの実行ボタンで直接表示させる事はできません。また、UserForm を Show メソッドで表示させた場合も同様にエラーになります。標準モジュールのプロシジャから clsRangeSelection の [Show メソッド](#) で表示させてください。

Q 2. RefEdit ボタン(CommandButton)のアイコン画像は、どのように設定しますか？

A 2. サンプルブックの UserForm1 にある RefEdit ボタンをコピー&ペーストすれば、アイコン画像付きで作成できます。Excel のアイコン画像に変更したい場合は、アイコン画像ファイルをダウンロードして、その中にある画像ファイルをVBEのプロパティウィンドウから CommandButton の Picture プロパティに読み込めば差し替えられます。



アイコン画像ファイル：http://addinbox.sakura.ne.jp/Download/RefEdit_Icon.zip

Q 3. RefEdit を UserForm に複数配置したいのですが、クラスオブジェクトは RefEdit の数だけ定義するのですか？

A 3. clsRangeSelection のクラスオブジェクトは、1プロジェクト（ワークブック）に1つだけ定義して、その1つのクラスオブジェクトを全ての RefEdit（プロジェクト内に RefEdit を使う UserForm が複数ある/UserForm 上に RefEdit が複数ある）で共有して使用します。

Q 4. RefEdit を使う UserForm がブックに複数あるのですが、クラスオブジェクトは UserForm ごとに定義するのですか？

A 4. A 3. を参照

Q 5. シートを切り替えてからセルを選択しても、セルアドレスのシート表示が変わりません。

A 5. その場合には、一旦、RefEdit フォームをクリックしてからセル選択を行ってください。

Q 6. UserForm Activate イベントを使っていますが、RefEdit フォームを閉じる度に Activate イベントの処理が実行されてしまいます。

A 6. UserForm.Hide → RefEdit フォーム表示 → UserForm.Show の流れで作られていますので RefEdit フォームを閉じる度に Activate イベントが発生します。[ActivateInfo プロパティ](#)の値で Initialize 後の Activate か、RefEdit 入力後の Activate かが判断できます。判定方法の詳細は [ActivateInfo プロパティ](#)を参照してください。

Q 7. RefEdit 入力後のイベント処理を行いたい。

A 7. RefEdit フォームを閉じる度に UserForm_Activate イベントが実行されます。Activate イベント内で [ActivateInfo プロパティ](#) の値を判断して、Initialize 後の Activate か、RefEdit 入力後の Activate か (どの RefEdit か) が判断できます。判定方法の詳細は [ActivateInfo プロパティ](#) を参照してください。

Q 8. 他のブックのセルを選択したい。

A 8. [Show メソッド](#) の BookChange 引数に True を指定すると他のブックのセルを選択できます。なお、ブックの切り替えは RefEdit フォーム上から行います。

Q 9. UserForm 表示の際に RefEdit に初期選択セルを設定しておきたい。

A 9. Activate イベントで [ActivateInfo プロパティ](#)="Initialize" の際に、[TargetRange プロパティ](#) へ初期選択セルの Range オブジェクトを設定(Set)します。セルアドレス文字は自動的に編集されて RefEdit ラベルに表示されます。

Q 10. RefEdit で選択したセル範囲の Range オブジェクトを取得したい。

A 10. [TargetRange プロパティ](#) で取得します。

Q 11. RefEdit で選択したセル範囲のセルアドレス文字を取得したい。

A 11. [TargetAddr プロパティ](#) で取得します。

Q 12. RefEdit で選択したセル範囲のセルアドレス文字のシートとセルを分割したい。

A 12. [TargetBook プロパティ](#)/[TargetSheet プロパティ](#) でブックとシートオブジェクトが取得できますので、その Name プロパティから取得できます。セルアドレス部分は [TargetAddr プロパティ](#) で [CellOnly = True] とします。

Q 13. RefEdit で選択した内容をクリアしたい。

A 13. RefEdit 入力でも何もセルを選択しないで戻ればクリアされます。マクロで行うには、[Clear メソッド](#) を実行するか、[TargetRange プロパティ](#) に Nothing を設定するとクリアされます。

Q 14. RefEdit での選択を単一セルに制限したい。

A 14. [SetRefEdit メソッド](#) で [RangeType = 2] とするとセル範囲を選択しても取得されるのは、その先頭セルのみになります。

Q 1 5. ブック切替機能を使用していて、セル選択していたブックをマクロからクローズしても大丈夫ですか？

A 1 5. そのブックを選択している RefEdit を残したままでブックをクローズすると、エクセル上では消えても、VBE 上では VBA プロジェクトが残ってしまいます。ブックをクローズする前に、そのブックを選択している RefEdit を全てクリアする必要があります。[ReleaseBookReference メソッド](#)に、そのクローズするブックオブジェクトを指定すれば、そのブックのセルを選択している RefEdit を一括してクリアします。ユーザーサイドで「どの RefEdit が、そのブックを選択しているか」を調べる必要はありません。

Q 1 6. 排他制御が何故必要なんですか？

A 1 6. 「[§ 3 - 2 - 2 RefEdit 代替手法を利用する際に排他制御が必要な理由](#)」を参照。

3-7 修正履歴

- Ver 2.00 2010/ 4/24
- Ver 2.10 2010/ 5/ 2
 - a) 標準モジュール名を変更しました。 [mdlRangeSelect ⇒ mdlRangeSelection]
 - b) 代替手法利用時の排他制御に関する修正を行ないました。
これに伴い、標準モジュール(mdlRangeSelection)が 必須 となりましたので、一緒に インポート してください。
 - c) 排他制御の判断用に RefEditActive プロパティ と IsRefEditActive 関数を追加しました。
 - d) [TargetRange]プロパティ(値の設定)で、指定の Range オブジェクトが処理対象ブックか否かのチェックが抜けていたのを修正しました。
- Ver 2.20 2010/ 5/14
 - a) SelectionChange イベントが常時稼働 (RefEdit 入力中か否かをフラグで判別してスキップ) していたのを、RefEdit フォーム表示中のみに変更 (監視ワークブックオブジェクトを、RefEdit フォーム表示直前に設定し、RefEdit フォーム消去直後に破棄(Nothing)します) しました。
 - b) RefEdit フォームにブック切替機能を追加しました。これに伴い、Show メソッドにブック切替機能用の引数を追加し、ReleaseBookReference メソッドを追加しました。
 - c) TargetAddr プロパティに CellOnly 引数を追加しました。
 - d) StartUpBook プロパティを追加しました。
 - e) 結合セルを選択した場合に「セル範囲」となっていたのを「結合セルの先頭セル」が取得セルとなるように修正しました。
 - f) 『 F A Q 』を記載しました。
- Ver 2.30 2010/12/15
 - a) SetRefEdit メソッドの RangeType 引数に[3: 1つ以上の単一セルを選択]を追加しました。
 - b) TargetRange(設定プロパティ)にて、指定の RangeType と異なる Range オブジェクトが設定できてしまう不具合を修正しました。
 - c) InitRefEdit メソッドを追加しました。
 - d) A P I 呼び出しの 64bit 対応を行ないました。
- Ver 2.31 2010/12/25
 - a) 隣接するセルを CTRL で選択した場合に「セル範囲」に纏められてしまう不具合を修正しました。
 - b) RangeType 引数に[3: 1つ以上の単一セル]で隣接するセルを指定できない不具合を修正しました。

• Ver 2.40 2011/7/1

- a) RefEdit ボタンのアイコンをオリジナルアイコンに変更しました。
(機能の変更はありません)。

Addin Box

Addin Box

Addin Box

Addin Box