

SRIMfit 技術メモ

Update Log :

2018.10/22 ver 3.01.03 Excel97-2003(Win) 対応
2018.10/11 ver.3.01.02 Mac対応、SRIMwb切替、デバッグmemo 追記。
2018.09/01 ver.3.01.00 インストーラー memo を追加。
2017.06/01 ver.3.00 マクロとWS分離に伴う変更。関数リスト追加。
2017.03/21 ver.2.12 ver.211 から転記。関数リストなど追加。

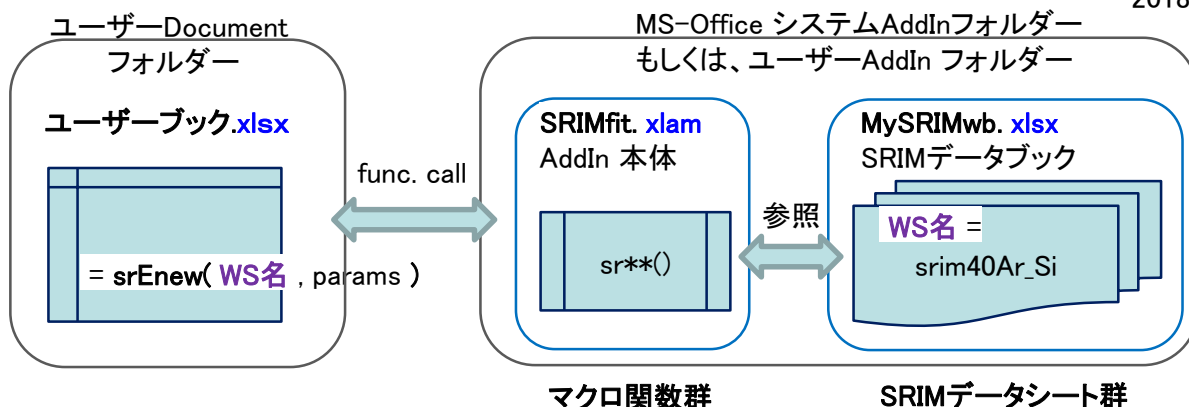
SRIMfit 技術メモ

目 次

- SRIMfit の詳しい構造
- マクロ利用のための Excel How To
- マクロの修正： 外部参照マクロとして使用
- 移行方法： SRIMfit ver.2* の WS を移行する
- その他 memo
- 参考文献

SRIMfit の 詳しい構造について

2018.10 追記



ユーザーマニュアル【SRIMfitのUserI/F】で説明したように、SRIMfitの本体は、システムAddInフォルダにインストールした

SRIMfit.xlam マクロ記述ファイル (SRIMfit.xlsm) を Excel アドイン形式に Compile したもの

MySRIMwb.xlsx ユーザーが記述する SRIM-2013 の計算結果データベース

の2つです。

尚、Excel-2003～97 の場合は、

アドインファイル形式が異なるので、SRIMfit.xlam は、SRIMfit.xla になります。

MySRIMwb.xlsx は、同じモノを使えます。

以下では主に、マクロ記述ファイル の構造について説明します。

(1) マクロ関数記述「SRIMfit.xlam」

Compile 前の元ファイル(.xlsm) は、lib¥xlsm にあります。

同フォルダに、.xlsm からエクスポートしたマクロ記述の テキストファイルも置きました。

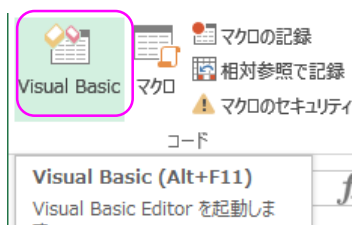
注) Excel は、AddInとして open しているモノと同じファイル名のモノは開けません。

なので、ここに置いたファイル名には、version番号を付けて、別名称にしています。

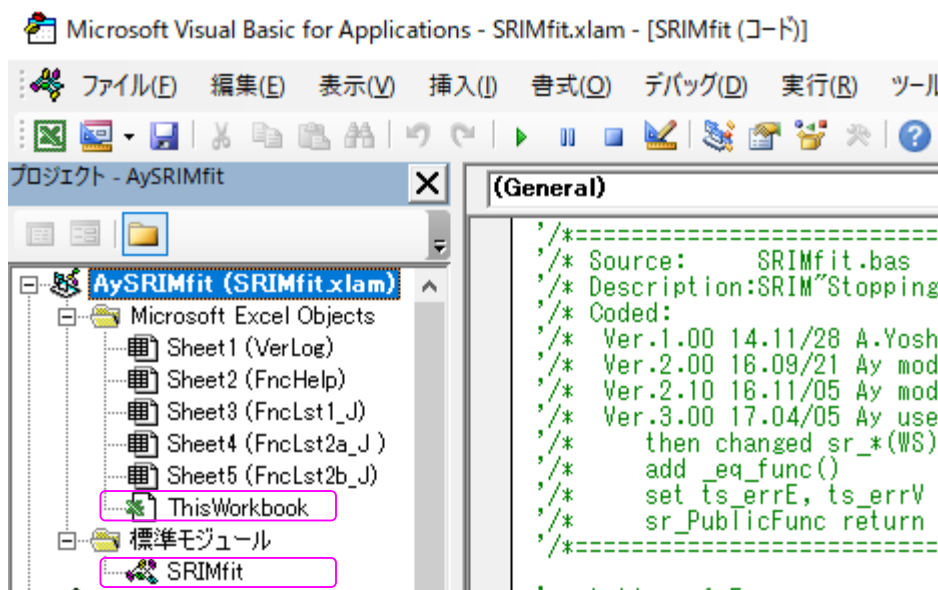
尚、SRIMfit_xxx.xlsm を開くと、そのマクロが実行されてVBAエラーが表示されることがあります。

その場合は、VBA Editor で、実行停止をして下さい。

元ファイルを開かなくても、マクロ関数記述の参照は可能です(プロテクトをかけていません)。
[開発:Visual Basic] で、VBAエディタを起動すると、AddInフォルダにインストールされている SRIMfit.xlam の マクロ記述の中身を見ることができます。



SRIMfit のマクロプログラムは、
エクセルに標準装備されている
Excel VBA (Visual Basic) で記述してあります。
このリボンボタンをクリックすると、
次頁のように VBA Editor が起動されます。



標準モジュール:SRIMfit が、マクロ記述のプロシージャ定義です。
ThisWorkbook Class モジュール は、Excel起動時の AddIn 読み込み時に実行されます。

標準モジュール:SRIMfit には、
【関数一覧】で示した、sr * () 関数 などが記述されています。

ThisWorkbook には、
AddIn モジュール 又は 外部参照マクロブック として、このブックが Open / Colse
された時に実行する処理が記述されています。

(2) マクロ起動・終了処理

マクロの起動・終了は、Excel システムによって自動的に行われます。
Excel 起動・終了時に呼び出される関数は、ThisWorkbook モジュール中の
Private Sub Workbook_Open() , Workbook_BeforeClose() です。

[開発:アドイン] の チェック を ON/OFF した時に呼び出される関数は、
Private Sub Workbook_AddinInstall() → Workbook_Open() の順で open処理
Private Sub Workbook_AddinUninstall() → Workbook_BeforeClose() の順で close処理
です。

Open / Close 処理の詳細は、これら関数の記述を読んでみて下さい。

関数のヘルプメッセージの定義・定義消去 処理

MySRIMwb.xlsx ファイルの Open / Close 処理

などが実行されます。

この Open処理の時に、SRIMfit モジュールの Public Sub srMySRwb_open() が呼ばれ、

Private MySRwbNow As Workbook ← Current WorkBook ポインタ

が MySRIMwb.xlsx ファイルを指すように設定されます。

(3) マクロ関数の動作

2018.10 追記

SRIMfit モジュールに記述してある関数の関数名は、

```
sr * () : Public Function
sr_ * () : Private Function    のように分類してあります。
```

Public Function の殆どは、その引数に **WS名文字列** を call by value で指定します。

そして最初の処理として、**sr_SetWS()** を呼び出して、WSの切替動作を行います。

```
例: Public Function srE2Rng(ByVal WS As String, ByVal Ei As Double) As Variant
    sr_SetWS( WS )
    ~ ~
End Function
```

sr_SetWS() では、overhead 低減の為、最小限の WS切替処理を行います。

```
Private MySRwsNow As Worksheet ← Current WS ポインタ
Dim WSnow As String ← Current WS名
Private Sub sr_SetWS(ByVal WSnew As String)
    If (WSnew <> WSnow) Then
        Set MySRwsNow = MySRwbNow.Worksheets(WSnew) ← Current WS ポインタの設定
        With MySRwsNow ← 以降、このポインタ参照で、WS中の Cell 値を読み出す
            IxEmin = .Cells(RowEmin, ClmInfo)
            ~ ~ など、WSをアクセスするための共通情報だけ、メモリ変数に読み込む
        End With
        WSnow = WSnew ← Current WS 名 変数の切替え
    End If
End Sub
```

以上の様な手順で、MySRIMwb.xlsx ブック中の WS を随時切り替えて、マクロ関数が動作します。
つまり、MySRIMwb の内容全てを配列変数などに読み込んでいるわけではありません。
MySRIMwb を、多数のWSからなる構造体データベースとしてとらえ、Current WS ポインタ を頼りに
随時 Cell 参照を行っています。

この様な SRIMfit の構造設計により、AddIn マクロ起動時の overhead を低減し、
ユーザーが MySRIMwb データベースを適宜拡張可能な I/F を提供しています。

という事は、ユーザーの .xlsx ファイルから、WS名が異なる関数呼び出しを行うと、凄く遅くなるか？
と思われそうですが、 実際はあまり気にならないようです。
今までの使用経験上、sr_SetWS() のWS切替処理の overhead より、複雑な処理をする srEnew() などの
組合せ関数の呼び出し回数による負荷の方が大きいようです。

(メモ) MySRIMwb 以外を参照するには。。

srMySRwb_open(ByVal MyFn as String) は、MyFn の引数をとれるようにしてあります。
MyFn=Null の時は、default で同じAddIn フォルダにある MySRIMwb.xlsx を開きます。
→ Ver.3.1.2 からは、sr_eg.AddIn.xslm に、SRIMwb切替え用スクリプトを用意しました。
詳しくは サンプル紹介、 及び その他memo:MySRIMwb切替 を参照ください。

(4) 関数ヘルプメッセージの定義

2018.10 追記

関数選択ボックスに「関数のヘルプメッセージ」を表示するようにしました。

登録・削除を行うマクロ関数は、ThisWorkbook class module にある

Private Sub Fhelp_def() と Fhelp_undef()

です。

登録関数は、FncHelp シート に記述した 関数とその引数一覧表に従って動作します。

- ・ 英語版、日本語版 (スイッチで切替式) で説明文を書いてあります。
- ・ ツタナイ説明ですので、このシートの内容を適当に修正して下さい結構です。
但しシートの構造はいじらないように願います。

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	1	2	3	4	5	6								15	16	17		
2	2	SRIMfit Function List					SRIMfit Function List (日本語表記)											
3	3	This sheet is used for <i>Application.MacroOptions</i> method. See the macro def.					このシートは <i>Application.MacroOptions</i> メソッドで参照されます。詳しくはヘルプメッセージを参照してください。											
4	4	You can modify the contents of this sheet except its structure.					詳しい英語・日本語ですので、ご自由に書き換えて使ってください。											
5	5																	
6	6	Lang. mode		1 = Q in English 1: in Japanese			← ヘルプメッセージ 言語切替Sw											
7	7	Num. of Func.		74														
8	8	Func. Descr. Header		This function returns			この関数は、											
9	9	Func. Descr. Footer					を返します。											
10	10	Arg. Descr. Header		indicates			は、											
11	11	Arg. Descr. Footer					です。											
12	12																	
13	13	Func.	Function	Num. of	Func. Description	Arg. Description	関数の説明					関数に渡すパラメータの説明					in	
14	14	No.	Name	Args.	In English	Arg1	In Japanese					Arg1	Arg2	Arg3	Arg4			
15	15	1	srMcVer	0	SRIMfit ver		SRIMfit version番号											
16	16	2	srMcPath	0	SRIMfit installed path		MySRMb の インストールPath											
17	17	3	srMcWbName	0	Users SRIM workbook file name		MySRMb の File Name											
18	18	4	srMcWbCount	0	Number of the Users SRIM worksheet		MySRMb に含まれる Sheet数											
19	19	5	srMcWbList	0	Name list array of the Users SRIM worksheet		MySRMb に含まれる 全Sheet名 を 1Dim文字列配列で返す											
20	20	6	srElmNm	1	Element symbol	Element(Z) number = 1..118						元素番号 Z=1..118						

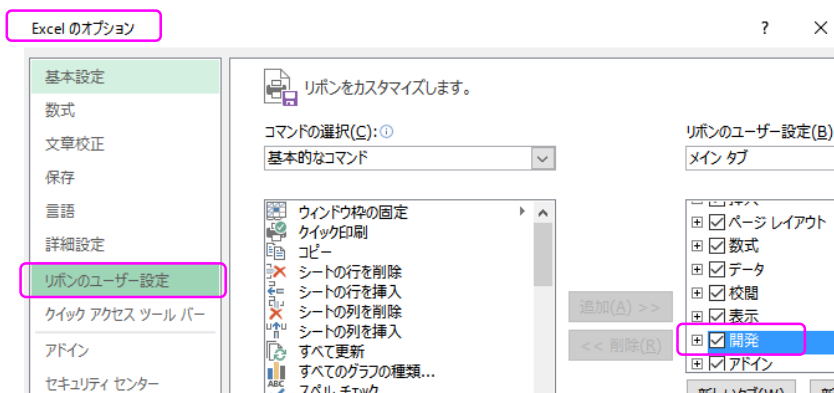
マクロ利用のための Excel How To

2017.06 記

Excel でマクロを使用する場合に必要なテクニックの How To 集です。
図は、Win10 Excel2013 の場合で説明します。他の環境では、表示が多少異なる場合があります。

(HwTo-01) [開発] タブ を表示する

[ファイル:オプション:リボンのユーザー設定]で、メインタブ の [開発] にチェックを入れます。



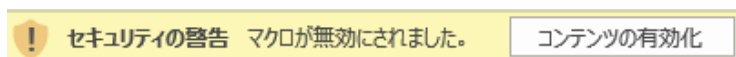
(HwTo-02) マクロを実行できる環境にしておく

[開発]タブ が表示されるようになったことを確認します。 次に、
[開発:マクロのセキュリティ]で、「警告を表示しすべてのマクロを無効にする」をチェックします。



セキュリティ上の観点から、この様な設定にしておいた方が安全です。
「警告を表示せずにすべてのマクロを無効にする」という設定では、マクロが動作しません。

マクロファイルのフォルダ位置やファイル名等を変更した後に、



と表示される時があります。その場合は「コンテンツの有効化」をクリックしてください。

マクロ呼び出しを膨大な回数行っているブックを開いた場合、
エクセル画面がずっと真っ白のままで、画面最下行に「再計算」表示
が現れます。

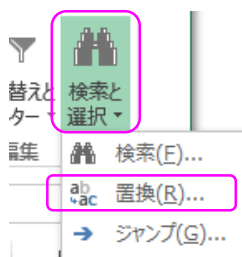
再計算: (プロセス 8 個): 25%

お使いのCPUの性能により、待ち時間は異なりますが、
あまりにも 遅い！ 重い！ 場合には、1つのブック内のWS数を減らしてみてください。

(HwTo-03) 数式の一括置換

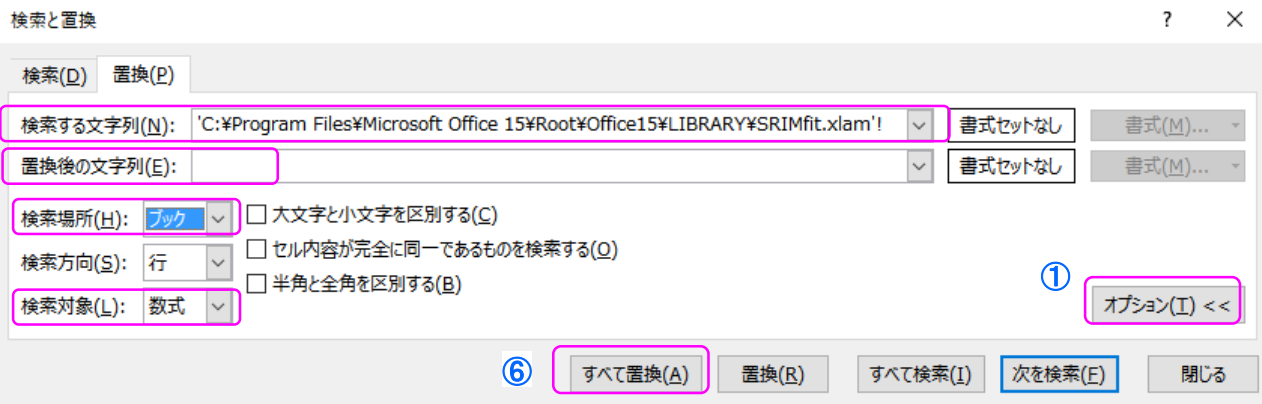
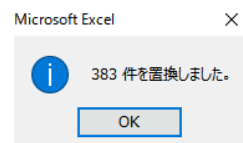
ユーザーブック (*.xlsx) において、

- リンクエラー発生の際に、操作を間違えて、全ての数式がフルパス表示に書き換わってしまった…
 - AddInマクロ ↔ 外部参照マクロ の関数呼出しを変更する
- といった場合に必要となるテクニックです。



[HOME] リボンの [編集:検索と置換] を用います。

- ① オプション メニューを表示させます。
- ② 検索場所 は、ブック全体 を指定します。
- ③ 検索対象 は、数式 についてです。
- ④、⑤ で、検索と置換 の文字列を指定し、
- ⑥ すべて置換 をクリックして暫く待ちます。



フルパス表示を修正する

修正前 = 'c:¥Program Files¥~¥SRIMfit.xlam'! srFunc() → 修正後 = srFunc()
 とした場合は、

- ④ 検索する文字列 は、「フルパス」部分の文字を指定します。即ち、'C:¥~ ¥~' ! で、
 シングルクォート から シングルクォート と ! マークまで です。
- ⑤ 置換後の文字列 は、空白のまま です。

外部参照マクロ ↔ AddInマクロ の修正

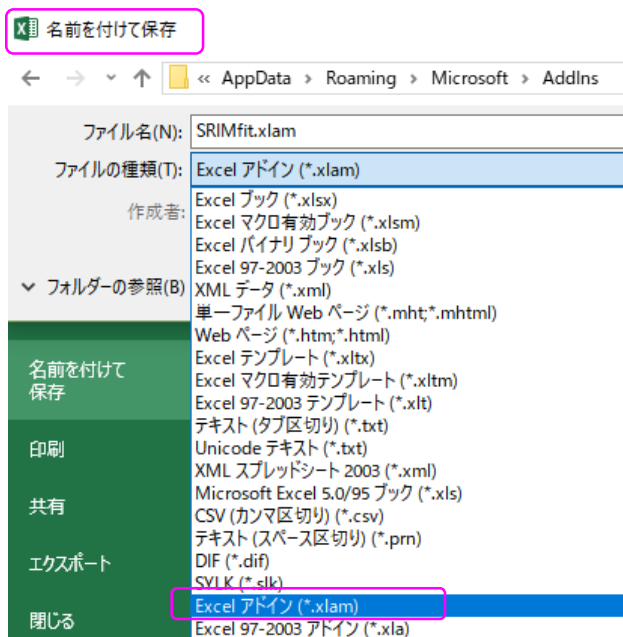
外部参照マクロ 「= SRIMfit.xlam ! srFunc() 」 ↔ AddInマクロ 「= srFunc() 」
 ですので、

- ④、⑤ は、「= SRIMfit.xlam !sr 」 か 「= sr 」 で指定します。

(HwTo-04) ファイル形式を変換して保存

例えば、マクロ記述がある SRIMfit.xlsm など、AddIn 形式 (*.xlam) で保存する時に必要となるテクニックです。

1) 形式を変換して保存



[形式を変換して保存する “前に”
変換元のファイルを、
CTRL+S で保存 しておきます。

[ファイル：名前を付けて保存] で、
ファイルの種類 をクリック すると、
ファイル形式の一覧表が表示されます。

AddIn マクロとして用いる場合は、
Excelアドイン (*.xlam) を選択します。

*.xlam ファイルの保存場所は、デフォルトでは ユーザーAddIn フォルダ
c:\¥～ ¥ユーザー名¥AppData¥Roaming¥Microsoft¥AddIns
になっています。必要に応じて保存先を変更してください。

尚、～ ¥ユーザー名¥AppData は、通常は非表示フォルダです。
表示させるには、エクスプローラの[表示:隠しファイル]にチェックを入れます。

2) 保存 しないで、エクセルを終了

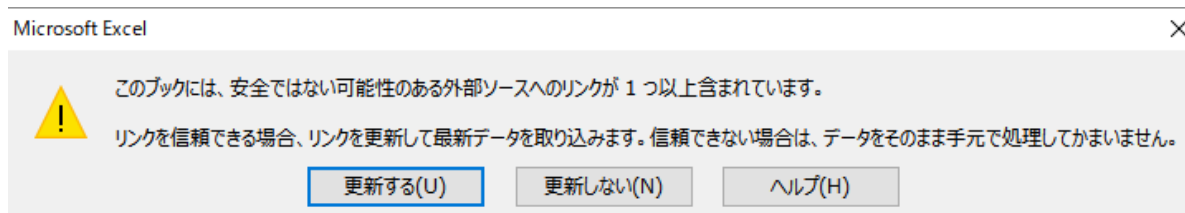
変換して保存後に、変換元のマクロファイルの終了方法ですが、
通常のように 保存してから終了すると、リンク先が上記の保存先 (ローミング用AddInフォルダ等)に
「自動的」に置き換わってしまうことがあるので、注意が必要です。
「保存しますか？」と表示されたら「キャンセル」して、エクセルを終了します。
尚、「MySRIMwb.xlsx を 保存？」と聞かれることもあります。
このファイルは SRIMfit.xlsm から ReadOnlyOpen しているので、保存は「キャンセル」します。

3) 保存された SRIMfit.xlam を、手動で移動

保存された .xlam ファイルを、
インストールガイド で述べた様に、指定されたAddInフォルダへ 手動で移動して下さい。
そして、有効なAddIn モジュールのチェックを入れてください。

(HwTo-05) リンクエラー対策

ユーザーが作成した sr*() 関数呼び出ししている *.xlsx ファイルを開いた直後に、次のようなメッセージが表示されることがあります。



どのような状況で、このメッセージが表示されるかというと、(作者の経験では…)

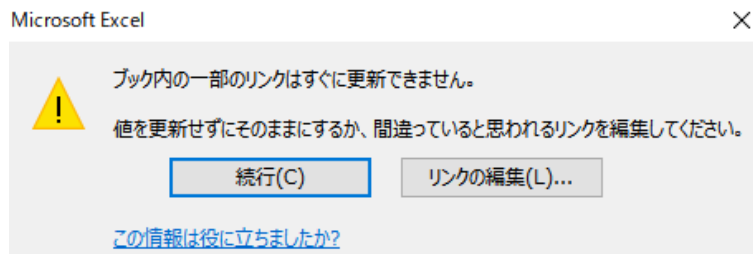
- Case 1) AddIn 選択の チェック を ON/OFF した後
- Case 2) AddIn フォルダの .xlam を更新した場合
- Case 3) AddIn フォルダの場所を変更した場合
- Case 4) sr*() 関数のマクロ呼出し方法を AddIn ↔ 外部参照 に切替えた場合
- Case 5) 何か操作を間違っ、sr*() 関数の呼出しが フルパス表示 になった場合
- Case e.t.c) マクロでは無い他のリンクエラーの場合

このように、いろいろな場合があるので、一概にその対処方法を決めることはできないのですが、以下にケーススタディを述べます。

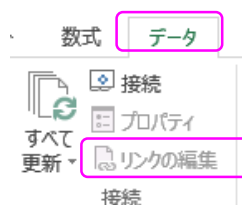
- Excel の リンクエラー について 詳しい説明 は、
下記文献をご覧ください。大変詳しく明快に開設されております。
http://addinbox.sakura.ne.jp/Excel_Tips21.htm AddinBox 角田 桂一氏
Excelのリンクの管理と格納(ワークブックの移動で起きるリンクエラーの仕組み)

Case 1), 2) は、→ AddIn の リンク元を確認 だけで済むことがある

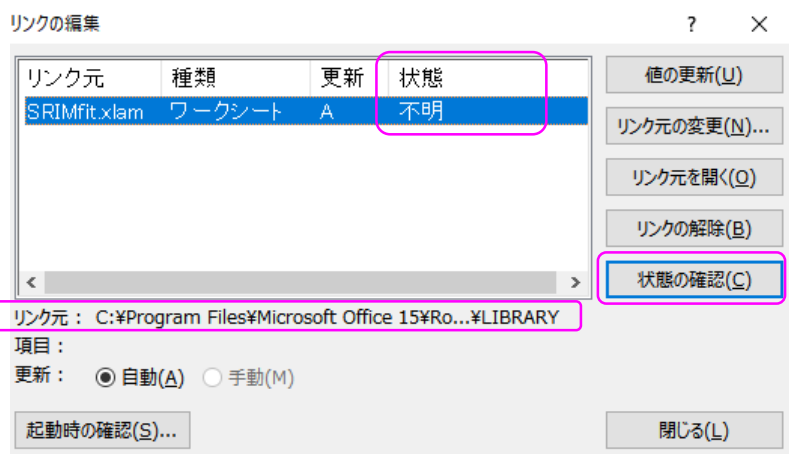
前のメッセージで「更新する」を選択してください。
すると、続けざまに、次のメッセージが表示されます。



また迷うのですが、「リンクの編集」を選択してください。すると、次のウィンドウが表示されます。

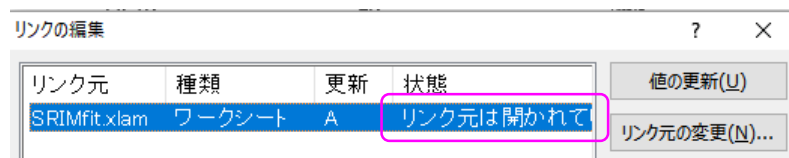


尚「リンクの編集」は、リボンメニューから
[データ：リンクの編集]を選択しても
随時確認することができます。



SRIMfit.xlam の「状態＝不明」と表示されている場合があります。

リンク元:c:¥～
は正しそうなのに、変だなあ～と思われる場合は、「状態の確認」をクリックしてください。



AddIn のインストールが正しければ、「状態＝リンク元は開かれています」と表示されます。

「閉じる」をクリックして終了。

そして、“念のため” エクセルを再起動 します。
その時、今エラー表示になっているブックは、「保存しない」で、再起動 したほうが安全です。
保存すると、エラー表示になっているセルの内容も そのまま保存されてしまう事があるからです。
再度、同じブックファイルを開いて エクセルを再起動してください。
エクセルシステムが、ブック中のセル全てについて、正しく修正されたリンクを用いて、「再計算」を実行してくれます。

Case 3) は、→ AddIn の リンク元を変更 を試みる

上と同じウインドウで、「リンク元を開く」や「リンク元の変更」をクリックしてください。

インストールガイドで説明した、AddIn フォルダ

MS-OfficeシステムのAddIn フォルダ : [Application Library Path](#)

ユーザー AddIn フォルダ : [Application UserLibrary Path](#)

のどちらかに、リンク元の .xlam があることを確認・変更 してください。

その後、「状態の確認」をクリックして、「状態＝リンク元は開かれています」と表示されれば、OK “でしょう”。

そして、case 1),2) と同じく、“念のため” エクセルを再起動 します。

Case 4), 5) は、→ #NAME? とエラー表示されている セル内の数式 を編集する

最初のメッセージで「更新しない」を選択、次のメッセージで「続行」を選択すれば、メッセージは消えて、通常の画面に戻ります。

ブック内のセルで #NAME? や #REF! エラーが表示されている箇所を探し出します。

セル内の、関数呼出し式 の確認をしてみてください

例えば、次の様に 長～い 記述式 になっていませんか？

例1) = 'C:¥Users¥ユーザー名¥AppData¥Roaming¥Microsoft¥AddIns' ! srInfoIonZ(C\$3)

例2) = 'C:¥Program Files¥Microsoft Office¥～¥～¥LIBRARY¥SRIMfit.xlam' ! srInfoIonZ(C\$3)

参照先のワークブックやアドインブックが開かれている状態では、

= srInfoIonZ(C\$3) のように、短く「**関数名のみで表示**」されます。

参照先のブックが閉じている状態では、

= '参照先のフルパス' ! srInfoIonZ(C\$3) のように「**フルパスが付いた表示**」となります。

この「**フルパス表示**」は、

自分で意図的に記述していなくても、ブックを保存する時に自動的に付加されます。

これが最近バージョンのエクセル仕様のようで、便利なようですが、時に非常に厄介な機能です。

参照先のリンクが切れた状態で 保存 してしまったファイル や、

保存する時はリンクがつながっていたけれど、再度開いた時にリンク切れが発生した場合 は、

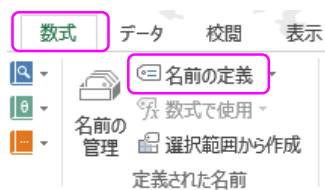
今まで短い「**関数名のみ表示**」で記述していたにもかかわらず、

保存された時の環境で参照していた「フルパス」が付加された長～い表示に自動修正されてしまうようです。

このような、関数呼び出し式のエラーが見つかった場合は、

【(HwTo-03) 数式の 一括 置換】に述べた方法で、変更を試みて下さい。

Case e.t.c.) は、→ 名前の管理 を確認してみる



マクロ関数参照のエラーではなく、

名前参照のエラー の濡れ衣 という可能性もあります。

リボンメニューから **数式：名前の定義** を選択し

ブック内の名前参照にエラーが発生していないか確認して下さい。

SRIMfit ver.2* の WS を移行する

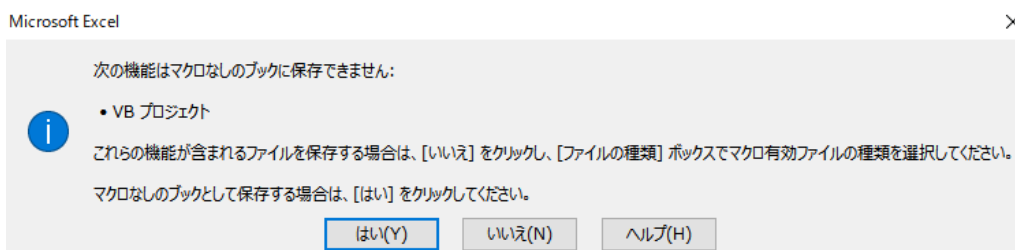
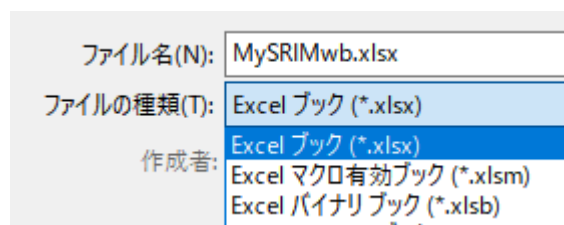
2017.06 記

「SRIMfit のUser I/F」で説明したように、
 ver 3 からは、SRIMfit.xlam(ver3) と MySRIMwb.xlsx を分離しました。
 ver 2 では、SRIMfit.xlam(ver2) 内部の WS として、MySRIMwb のWSを入れてありました。

よって、作業は簡単で、
 SRIMfit.xlam(ver2) を アドイン形式で保存した時の、ご自分の元ファイル
 SRIMfit.xlsm(ver2) から、マクロ部分を消去 して、MySRIMwb.xlsm を作る
 だけです。

マクロ部分を消去して MySRIMwb.xlsx で保存

今まで使っていた SRIMfit.xlsm (ver2版) を開きます。
 [ファイル:名前を付けて保存]で、
 ファイル名 = MySRIMwb
 ファイルの種類=Excelブック (*.xlsx)
 で保存します。



当然の様に、上のような警告メッセージが表示されますが、「はい」でOKです。

作った MySRIMwb.xlsx を、AddIn フォルダに保存

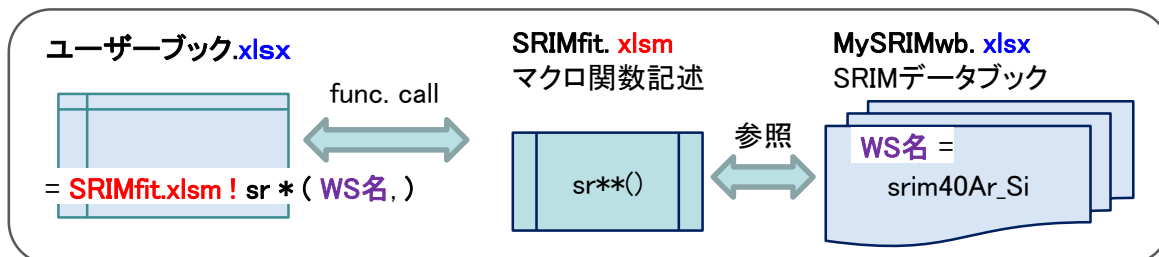
あとは、インストールマニュアルに従って、
 SRIMfit.xlam をインストールしたのと同じ AddInフォルダー へ
 MySRIMwb.xlsx を 上書きコピー して下さい。

マクロの修正： 外部参照マクロとして使用

2017.06 記

今まで説明してきた AddInマクロ関数 としての使い方以外に、外部参照マクロ関数 として使う場合について説明します。AddIn方式の方が便利なので、あまり使う機会は無いのですが、例えば「マクロ関数部分の修正(デバッグ)」など、SRIMfit.xlsm を .xlsm 形式にコンパイルせずに扱う必要がある時には便利です。

同じフォルダー

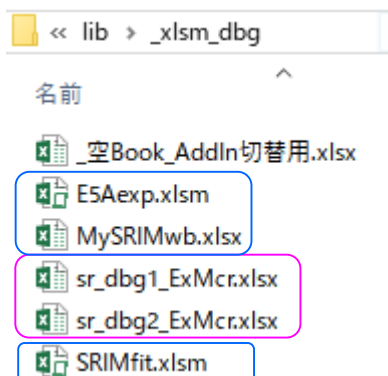


外部参照マクロ として使う場合、今までのAddInマクロと異なる点があります。

- マクロ記述ファイルは、SRIMfit.xlsm です。SRIMfit.xlam ではありません。
- 3つのファイル： ユーザーブック.xlsx と SRIMfit.xlsm と MySRIMwb.xlsx は、なるべく、**同じフォルダー** に置いたほうが便利でしょう。
- ユーザーブックからの関数呼び出しは、
 同じフォルダーの場合： = **SRIMfit.xlsm** ! sr** (WS名, param)
 違うフォルダーの場合： = 'c:¥ディレクトリ¥ツリー¥SRIMfit.xlsm' ! sr** (WS名, param)
 のように、フルパスを シングルクォート " でくくれます。
- 3つのファイルの起動順番は、
必ず最初に、SRIMfit.xlsm を開いておく。
 この中の Workbook_Open() で MySRIMwb.xlsx は自動的に開かれます。
 それから、 ユーザーブック.xlsx を開く。

(1) SRIMfit.xlsm のデバッグ環境

参考として、.lib¥_xlsm_dbg 以下に、デバッグ環境をおいてあります。



Original .xlsm ファイル

マクロ記述ファイル **SRIMfit** と **E5Aexp** の元ファイルです。
プロテクトはかけてありません。

Debug 用 .xlsx ファイル

SRIMfit 関数のデバッグ用ファイルです。
関数呼出しは、外部参照マクロ 形式にしてあります。

空Book_AddIn切替用.xlsx は、
AddIn切替手順ガイドが書いてあるだけの空ブックです。
内部には、関数呼出しも、マクロも書いてありませんので、
外部参照 ↔ AddIn の切替時に、安全に使えます。

(2) SRIMfit.xlsm の デバッグ手順

1) AddIn を解除する

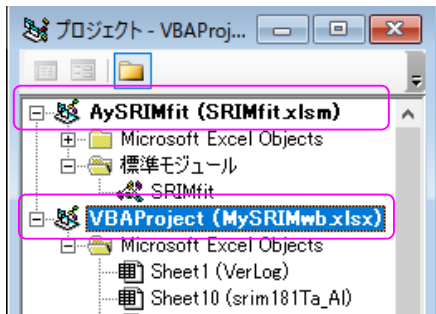
_空Book_AddIn切替用.xlsx を開いて、
「外部参照マクロ」として用いる場合 のガイドに従って下さい。

2) マクロが実行可能な環境であることを確認

【Excel HowTo: HwTo-02】に従って、
マクロが実行可能になっているか再確認しておいて下さい。

3) ファイルを開く順番に注意

必ず最初に、SRIMfit.xlsm を開いておく。次にユーザーブック(例: sr_dbg1_ExMcr.xlsx) を開く。
SRIMfit.xlsm 起動時の Workbook_Open() から MySRIMwb.xlsx がOpenされているかを確認する。



SRIMfit.xlsm 起動直後に、
VBAエディタ:プロジェクトを見て
SRIMfit.xlsm と一緒に
MySRIMwb.xlsx も読み込まれているかを確認する。

MySRIMwb.xlsx が読み込まれていないと、
ユーザーブックの sr*() 関数参照セルの表示が
エラー表示 (#NAME? など) となります。

4) ファイルの保存

3つのファイルのうち、変更を加えたファイルは随時 CTRL+S で保存しておきます。安全の為。
また、SRIMfit.xlsm のマクロ記述を 大幅に変更した場合(例:関数追加・削除・名前変更など) は、



VBAエディタの[デバッグ:コンパイル]を実行し
エラーが無いことを確認の上、
SRIMfit.xlsm を CTRL+Sで保存し、
一度Excelを終了してください。
Excelが再起動された時に、変更結果が有効になります。

5) AddIn モジュールとして保存

外部参照マクロ から AddInマクロ として使う場合は、
【Excel HowTo (HwTo-04) ファイル形式を変換して保存】を参照してください。

(3) AddInマクロをデバッグしたい方へ

2018.10 記

SRIMfit にはプロテクトをかけてませんので、ご自分の責任でご自由に修正・デバッグをして下さっても結構です。使用しているマクロは基本的なスクリプトですので、Excel-2003等の古~いExcelでも動作しているようです。
「AddInマクロのデバッグ」は、前述の「外部参照マクロのデバッグ」とほぼ同じ作業なのですが、幾つか注意点があります。参考として、以下にメモを記します。

(デバッグ手順)

- D01) Install_手動補助.xlsm で Application.LibraryPath = システムAddInフォルダー の場所を確認しておく。
D02) SRIMfit の AddInリスト=OFF して Excel終了。AddInフォルダーから SRIMfit.xlam を削除。
念の為、空bookを起動し、AddInリストでチェック無しになっている事を確認。
D03) 修正する SRIMfit_xxx.xlsm と テスト用の MySRIMwb.xlsx を、自分用の同じフォルダーに置いておく。
SRIMfit.xlsm のスクリプトで、MsgBox や Stop がコメントアウトされている部分を適当に復活させる。
Stopで VBA Editorに制御が移るので、チキチキとデバッグする。
Const 定義されている部分はなるべくいじらない。MySRIMwbの構造定義などなので。
D04) 修正した SRIMfit_xxx.xlsm は、先ず CTRL-S で保存。次に .xlam と .xla 形式で保存。
その後、Excel を終了する時に「.xlsm を保存しますか？」には、「しない」とすること。
操作を間違えると、.xlsmのファイル形式が返られてしまう恐れがあるので。
D05) 生成した .xlam や .xla は、バージョン番号を取って、SRIMfit.xlam のように名前変更してから、
AddInフォルダーにコピーする事。Winの場合、バージョン番号付きでもAddInリストには
VBA Object name= SRIMfitが使われるので良さそうだが、こうすると既存の *.xlsx ファイルに記述されている
関数参照式の .xlamファイル名が変更となり、全てリンク切れになり悲惨。
D06) AddInリストに復活させるときは、先ず空book起動し、AddInリストを表示させ、
残っている SRIMfit にチェックを入れるだけではダメな場合が多い。UserAddInフォルダーを参照されてしまう。
SRIMfitの行を選択し、(チェックを入れる前に)「参照」を押し、システムAddInフォルダーに入れた
修正版の SRIMfit.xlam の場所を指定。「既にあります。置き換えますか? =Yes」で追加すること。
D07) エラー無しで AddIn登録ができたなら、空book のセルに =srMcrPath() を書いて動作確認。

尚、慣れてくると、

- D02) AddInリスト=OFF をせずに、AddInフォルダーから SRIMfit.xlam を削除、
D06) 修正版 SRIMfit.xlam をAddInフォルダーへコピー だけでもよさそうです。Winの場合は。。

(MacOSでデバッグ時の注意点) 私はMacユーザーではないので、間違っているかもしれませんが。。

- ・ MacOSの場合、Excelプロセスが残るので、画面下タスクバーで 右クリック+終了 させてからすること。
- ・ MacOS 10.14 から sandbox化され、実体ファイルの場所が不明なので注意！
例えば、Install_手動補助.xlsm で Application.LibraryPath で表示されるフォルダー位置は
ファインダーで見える実際位置と異なる？らしい。
- ・ 特に、AddIn =OFF/ON する時には注意。sandbox化で何処かに古いファイルが残っている可能性あるかも？
D06) のように、毎回 システムAddInフォルダを指定し、「置き換えますか? =Yes」で追加すること。
- ・ Win版の .xlsm を Mac上に持ってくると、きっと必ず全てリンク切れになります。
.xlam のファイル名が同じでも、Win と Mac のファイル構造が大きく異なる為でしょうか？
ブック内の数式一括変換【HowTo 03】で、リンク切れを修正してください。

● インストールスクリプト

● 自動インストールスクリプトは、VBScript + DOS BAT にしてあります。

SRIMfit のインストールは単純で、ファイル2～3個を copy するだけなので、Visual Studio などでインストーラーを作成する程の必要はない。
ネット検索して、ちょうどよい例題 [ref Inst01](#)) があったので、これを参考にさせて頂いた。



このスクリプトは、VBScript で記述されており、Excel の AddIn フォルダの場所を取得、AddIn.Add でアドイン追加までしてくれる優れものであった。
但し、インストール先フォルダが UserLibraryPath であったので、これを Excelシステムの LibraryPath に変更することにした。

[問題発生1]: システムフォルダへcopy するので、管理者権限で VBScript を実行する必要あり。

でも、残念ながら VBS を右クリックしても、「管理者として実行」のメニューが無い。
いろいろ探した結果、VBSを管理者として実行する方法 [ref Inst02](#)) を見つけた。
これは正しく動作した。が、スクリプトが(私には)難しいので、採用しないことにして、
同 ref) に記されていた次の方法にした。

[解決策1]: VBSを呼ぶ DOS BATファイルを作成し、これを管理者モードで実行にした

DOS BATを管理者モードで実行すると、C:¥Windows¥system32 がカレントパスになってしまうので、
[ref Inst03](#)) を参照して、BATのカレントディレクトリを、BATファイルのディレクトリにした。

参考文献

- [Inst01](#)) <https://qiita.com/fuku2014/items/9c72fc04265bfc7f7f40>
【Excel】アドインをインストールするスクリプト ; Qitta @fuku2014
- [Inst02](#)) http://masahiror.hatenadiary.jp/entry/20111201/vbs_admin_run
VBSを毎回「管理者として実行」する方法 ; masahirorの気まま記録簿
- [Inst03](#)) <http://d.hatena.ne.jp/nakamura001/20090203/1233652705>
[Windows][バッチファイル]カレントディレクトリをバッチファイルのディレクトリにする

● Excelバージョン・互換性など

2018.10 記 ver.3.1.2

● Excel version 番号取得 [ref 01\)](#)

Application.Version : String型 で取得できる。

● 条件付きコンパイル [ref 02\)](#)

#IF VBA6, VBA7, Win64 などの compile option あり。
VBA Object model の変更点は、[ref03\)](#), [04\)](#)

● VBA互換性 [ref 05\)](#)

● ワークシート互換性 [ref 06\)](#), [07\)](#)

Excel	Application. Version	#IF VBA	#IF Win64	AddIn 形式	ワークシート ファイル形式
2016	“16.0”	Vba7	有	.xlam	.xlsx
2013	“15.0”	Vba7	有	”	”
2010	“14.0”	Vba7	有	”	”
2007	“12.0”	Vba6	未定義	”	”
2003	“11.0”	Vba6	未定義	.xla	.xls
2002	“10.0”	”	”	”	”
2000	“9.0”	”	”	”	”
97	“8.0”	”	”	”	”
95	“7.0”	?			
5.0	“5.0”	?			

参考文献

- 01) <http://officetanaka.net/excel/vba/tips/tips77.htm>
マクロとVBAのテクニック ; Office TANAKA
- 02) http://www.asahi-net.or.jp/~ef2o-inue/vba_o/sub05_800_500.html
条件付きコンパイルについて ; Excelでお仕事
- 03) <https://docs.microsoft.com/ja-jp/previous-versions/office/ff512734%28v%3doffice.14%29>
VBA オブジェクト モデルの変更点 ; Microsoft Docs
- 04) [https://docs.microsoft.com/ja-jp/previous-versions/office/ms474284\(v=office.14\)](https://docs.microsoft.com/ja-jp/previous-versions/office/ms474284(v=office.14))
VBA オブジェクト モデルの変更点の一覧表 ; Microsoft Docs
- 05) <https://techinfoofmicrosofttech.osscons.jp/index.php?VBA%E7%A7%BB%E8%A1%8C>
VBA移行;マイクロソフト系技術情報 Wiki
- 06) <https://support.office.com/ja-jp/article/%27ワークシートの互換性に関する問題-f9c80c5b-5afc-40da-a841-b888746abd40>
ワークシートの互換性に関する問題 ; Microsoft
- 07) <https://www.microsoft.com/ja-jp/download/details.aspx?id=27836>
Microsoft Office 互換機能パック Service Pack 3 (SP3) ; Microsoft

● Excel-2003,2000 インストmemo

2018.10 記 ver.3.1.2

SRIMfit ver3.1.13 から、Excel 97～2003 でも動作するようになりました。
一応、WinXP+Excel2000 まではテストしました。Excel97は持っていないので未テストです。

● VBA6 では Application.MacroOptions ArgumentDescriptions が使えない

SRIMfit部 には、VBA6 互換上の問題はありませんでした。
ThisWorkbook部 の、関数ヘルプ定義の部分に問題ありました。以下の様に対処。

```
Private Sub Fhelp_def()      より、要点部のみ
#If VBA7 Then ' for Excel 2010 or later
    Application.MacroOptions Macro:=Fnm, _
        Description:=Fdsc, Category:=Fcat, _
        ArgumentDescriptions:=Farg
#Else ' for Excel2007 or under does NOT have ArgumentDescriptions
    Application.MacroOptions Macro:=Fnm, _
        Description:=Fdsc, Category:=Fcat
#End If

Private Sub Fhelp_undef()    も同様に対処した。
```

● MS-Office互換性パック のインストールが必要

- Excel2003以前では .xlsx が読めないのです。
- Excel起動時、AddInを読み込むときに、MS-Office互換性パックが起動して MySRIMfit.xlsx が変換されるため、かなり遅いです。
- また、Excelの互換性については、下記文献をご参照ください。
シート数制限などがあるらしい。

<https://support.office.com/ja-jp/article/ワークシートの互換性に関する問題-f9c80c5b-5afc-40da-a841-b888746abd40>
ワークシートの互換性に関する問題 ; Microsoft

● MS-Office互換性パック 無し でも動くようにした。

- SRIMfit ver 3.1.3 より。 本体マクロ 書き換え、_Install_OldExcell を整備。

要点部のみ

```
Private Const MySRwbFnDfa = "MySRIMwb.xlsx" ' Default SRIM work book file name
Private Const MySRwbFnOld = "MySRIMwb.xls" ' for Excel 97-2003
Public Sub srMySRwb_open(ByVal MyFn As String)
Dim ExcelVer As Integer

DirSep = Application.PathSeparator ' get Path Separator Char
ExcelVer = CInt(Application.Version) ' get Excel ver. as integer
If (MyFn = "") Then ' open default WB
    MySRwbDirNow = ThisWorkbook.Path ' assume placed in same dir
    If (ExcelVer <= 11) Then ' for Excel 97-2003
        fn = MySRwbDirNow & DirSep & MySRwbFnOld
    Else
        fn = MySRwbDirNow & DirSep & MySRwbFnDfa
    End If
```

● Excel for Mac 版 インストmemo

2018.10 記 ver.3.1.2

MacOS 10.14 Mojave とExcel for Mac 2016 は、今までのバージョンから随分変わったそうで、Microsoft社やユーザーフォーラムの方々も混乱気味の様です。詳しくは以下の記事が参考になると思います。

- Microsoft Docs/Office VBA Reference/Office for Mac
<https://docs.microsoft.com/en-us/office/vba/api/overview/office-mac>
- Installing User Content in Office 2016 for Mac
<https://macadmins.software/docs/UserContentIn2016.pdf>

● File Path Separator が Windows と異なる問題

Public Sub srMySRwb_open(ByVal MyFn As String) を書き換えました。要点部のみ記します。

Dim DirSep As String

DirSep = Application.PathSeparator ← コレで、Win/Mac Excel共通になりました。

If (MyFn = "") Then

MySRwbDirNow = ThisWorkbook.Path ' assume placed in same dir

fn = MySRwbDirNow & DirSep & MySRwbFnDfa

● MS-Office システム AddIn フォルダーが異なる？ 問題

手動インスト用に用意されている '2_Install_手動補助.xlsm' を起動して、Application.LibraryPath を表示。この場所がFinderで見つからない。
MacOS 10.14 から Sandbox化されたのが原因か？



Install_SRIMfit_case1.vbs

尚、コレは、Excelがインストールされている MacOS なら動くかもしれない。
誰か試したら教えてください。

● Excel の Object Module が少々異なる？ 問題

ThisWorkbook の、関数ヘルプ定義の部分 Sub Fhelp_def() において、
Mac版 Excel for Mac 2016 + MacOS 10.14 でテストした限りでは、Application.MacroOptions の部分は
実行エラーになっていないのですが、関数ヘルプに カテゴリ=SRIMfit が表示されません。

Category= 数字で 1~20 を試してもダメ。

外人さんの書き込みでは出来ているというのもありましたが。。私は諦めました。

ref) KWD = excel for mac UDF(User Def. Func) MacroOption

- https://answers.microsoft.com/en-us/msoffice/forum/msoffice_excel-mso_mac-mso_mac2016/user-defined-functions-excel-2016-mac/74e13c40-13ee-4fca-b174-7ed8f64ba3a6
- MS-Japan Office Developer Support Blog ; Excel ユーザー定義関数に機能説明や独自ヘルプをリンクする方法
https://blogs.msdn.microsoft.com/office_client_development_support_blog/2017/11/21/udf-help/
- Application.MacroOptions プロパティ
<https://docs.microsoft.com/ja-jp/office/vba/api/Excel.Application.MacroOptions>

● MySRIMwb.xlsx の切替え

サンプル紹介も参照の事

2018.10 記 ver.3.1.2

File名: **sr_eg_AddIn.xlsm** = 3_Install後_動作確認.xlsm と同じモノ Sheet名: **eg11**マクロ情報

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	

ボタン を押すと実行されるスクリプトを以下に記します。

- ・ 必要に応じて、ご自分のワークシートにも同様なボタンを作っておくと便利かもしれません。
- ・ SRIMfit.xlam にある **srMySRwb_open()** 関数 を呼んでいます。

```

'/*-----
'/* button1: 現在使用中の MySRIMwb を 切替える
'/*-----
Sub btn1_Click()
    Dim fnSRIMwb As String
    MsgBox "変更する MySRIMwb を指定してください。"
    fnSRIMwb = Application.GetOpenFilename(
        FileFilter:="SRIMwbファイル, *.xlsx, 全てのファイル, *.*")
    MsgBox "選択された MySRIMwb=" & vbCrLf & fnSRIMwb & vbCrLf & _
        "に切り替えます。"
    Call Application.Run("srMySRwb_open", fnSRIMwb)
    Application.CalculateFull
End Sub
'/*-----
'/* button2: 既定の MySRIMwb に 切替える
'/*-----
Sub btn2_Click()
    MsgBox "既定の MySRIMwb に戻します。"
    Call Application.Run("srMySRwb_open", "")
    Application.CalculateFull
End Sub

```

- ・ これらのボタンは、SRIMwb を修正してテストする時などに便利です。
- ・ 例えば、ご自分のワークシートを開いて、SRIMfit関数の動作試験をしている最中に、この sr_eg_AddIn.xlsm も開いて SRIMwbを切り替えてテストする場合など。
- ・ この切替は一時的なものです。Excelプロセスを全て終了し、Excelを再起動すると、デフォルトの MySRIMwb.xlsx が読み込まれます。

※ 次の version では、これらのボタンを、リボンメニューにする事を検討してみます。

参考文献

SRIMfit の開発に当たっては、以下の文献・HomePageなどを参考にさせていただきました。
参照させて頂いた箇所については、詳しく記載できておりませんが、ご了承ください。
以下の諸先生方に感謝申し上げます。大変勉強になりました。

SRIM コード

SR01) <http://srim.org/index.htm> SRIMコード James F. Ziegler氏

Excel VBA 関連全般

以下順不同です

VB00) <https://msdn.microsoft.com/ja-jp/library/office/ee861528.aspx> Microsoft Office Dev Center

VB01) <http://www.asahi-net.or.jp/~ef2o-inue/top01.html> Excelでお仕事！ 井上 治氏

VB02) <http://home.att.ne.jp/zeta/gen/excel/> エクセル大辞典 武藤 玄氏

VB03) <https://sites.google.com/site/compositiosystemae/home> VBAシステム開発 氏

VB04) <http://excelshogikan.com/vba/vba.html> エクセル将棋館 氏

VB05) <http://www.moug.net/tech/exvba/> mougモーグ モーグ編集部殿

VB06) https://kokodane.com/macro_kouza.htm なんだ！カンタン！Excel塾 管理人殿

VB07) <http://www.clayhouse.jp/index.htm> Excel研究室 土屋和人氏

AddIn モジュール化 関連

AI01) <http://addinbox.sakura.ne.jp/index.htm> AddinBox 角田 桂一氏

AI02) <http://excel-ubara.com/excelvba4/> エクセルの神髄 鵜原パソコンソフト研究所殿

AI03) <http://officetanaka.net/excel/vba/tips/> OfficeTANAKA 田中 亨氏

以上です。

SRIMfit が、どなたかの実験のお役に立てれば幸甚です。
理研・仁科加速器センター・産業チーム あ吉田