**COMMENT MODIFIER LES FICHIERS MACRO MYRMIDON**

Table des matières

[Comment : afficher l'onglet Développeur sur le ruban 4](#_Toc439163175)

[Pour afficher l'onglet Développeur 4](#_Toc439163176)

[Figure 1 : Bouton Options dans Office 2010 4](#_Toc439163177)

[Figure 2 : Onglet Fichier dans Office 2013 4](#_Toc439163178)

[Figure 3 : Bouton Options dans Office 2013 5](#_Toc439163179)

[Figure 4 : Bouton Personnaliser le Ruban dans Excel 2010 5](#_Toc439163180)

[Figure 5 : Case à cocher Développeur dans Excel 2010 et Excel 2013 5](#_Toc439163181)

[Figure 1. Onglet Développeur dans Excel 2010 6](#_Toc439163182)

[Visual Basic Editor 6](#_Toc439163183)

[Figure 2. Visual Basic Editor 6](#_Toc439163184)

[COMMENT : MODIFIER OPTION SAUVEGARDE DISQUE DUR 7](#_Toc439163185)

[Etape 1 – Ouvrir le développeur Visual Basic. 7](#_Toc439163186)

[Etape 2 – Emplacement des macros : OPTIONS. 7](#_Toc439163187)

[Etape 3 – Description de certaines lignes de code. 7](#_Toc439163188)

[Etape 4 – Modification ou Ajout de chemin d’accès. 8](#_Toc439163189)

[Etape 5 – Chemin par défaut, à l’ouverture du fichier. 8](#_Toc439163190)

[Etape 6 – Vérification et validation. 8](#_Toc439163191)

[COMMENT : MODIFIER OPTION ENVOI PAR MAIL 9](#_Toc439163192)

[Etape 1 – Ouvrir le développeur Visual Basic. 9](#_Toc439163193)

[Etape 2 – Emplacement des macros : OPTIONS. 9](#_Toc439163194)

[Etape 3 – Description de certains codes préparés. 9](#_Toc439163195)

[Etape 4 – Les variables communes. 10](#_Toc439163196)

[Etape 5 – Caractère d’activation de ligne de code. 10](#_Toc439163197)

[Etape 6 – Solution d'envoi automatique de fichier Excel à partir de Windows Mail en PJ. 10](#_Toc439163198)

[Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 par P.J. avec Outlook. 11](#_Toc439163199)

[Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 en corps de texte avec Outlook. 12](#_Toc439163200)

[Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 en P.J. par serveur internet ou de messagerie. 14](#_Toc439163201)

[Etape 7 – Vérification et validation. 17](#_Toc439163202)

[Etape 8 – Modification de la réinitialisation de l’adresse mail à l’ouverture du classeur. 17](#_Toc439163203)

[Etape 9 – Vérification et validation. 18](#_Toc439163204)

[COMMENT : MODIFIER REINITIALISATION OPTION CHOIX NB COMBI 19](#_Toc439163205)

[Etape 1 – Ouvrir le développeur Visual Basic. 19](#_Toc439163206)

[Etape 2 – Emplacement des macros : OPTIONS. 19](#_Toc439163207)

[Etape 3 – Modification de la réinitialisation du choix du nb combi. à l’ouverture du classeur. 19](#_Toc439163208)

[Etape 4 – Vérification et validation. 19](#_Toc439163209)

[COMMENT : MODIFIER REINITIALISATION OPTION LIMITE PERMU MINI 20](#_Toc439163210)

[Etape 1 – Ouvrir le développeur Visual Basic. 20](#_Toc439163211)

[Etape 2 – Emplacement des macros : OPTIONS. 20](#_Toc439163212)

[Etape 3 – Modification de la réinitialisation du nb de permutation mini. à l’ouverture du classeur. 20](#_Toc439163213)

[Etape 4 – Vérification et validation. 20](#_Toc439163214)

[COMMENT : MODIFIER REINITIALISATION OPTION LIMITE PERMU MAXI 21](#_Toc439163215)

[Etape 1 – Ouvrir le développeur Visual Basic. 21](#_Toc439163216)

[Etape 2 – Emplacement des macros : OPTIONS. 21](#_Toc439163217)

[Etape 3 – Modification de la réinitialisation du nb de permutation maxi. à l’ouverture du classeur. 21](#_Toc439163218)

[Etape 4 – Vérification et validation. 21](#_Toc439163219)

[COMMENT : MODIFIER REINITIALISATION OPTION ARRET PAR MINUTEUR 22](#_Toc439163220)

[Etape 1 – Ouvrir le développeur Visual Basic. 22](#_Toc439163221)

[Etape 2 – Emplacement des macros : OPTIONS. 22](#_Toc439163222)

[Etape 3 – Modification de la réinitialisation du minuteur à l’ouverture du classeur. 22](#_Toc439163223)

[Etape 4 – Vérification et validation. 23](#_Toc439163224)

[COMMENT : SUPPRIMER L’INTERDICTION DE CREATION DE FEUILLE 24](#_Toc439163225)

[Etape 1 – Ouvrir le développeur Visual Basic. 24](#_Toc439163226)

[Etape 2 – Emplacement des macros : OPTIONS. 24](#_Toc439163227)

[Etape 3 – Suppression de l’interdiction. 24](#_Toc439163228)

[COMMENT : MODIFIER DATE D’ARRET PAR DEFAUT DU PLANIFICATEUR 25](#_Toc439163229)

[Etape 1 – Ouvrir le développeur Visual Basic. 25](#_Toc439163230)

[Etape 2 – Emplacement des macros : OPTIONS. 25](#_Toc439163231)

[Etape 3 – Modification de la valeur date d’arrêt par planification, par défaut. 25](#_Toc439163232)

[Etape 4 – Vérification et validation. 26](#_Toc439163233)

[COMMENT : INSTALLER ZAZ StopMacro 27](#_Toc439163234)

[Figure 1 : Présentation 27](#_Toc439163235)

[Etape 1 – Télécharger l’application : 28](#_Toc439163236)

[Etape 2 – Forum de suivi : 28](#_Toc439163237)

[Etape 3 – Lancement et 1er test ZAZ StopMacro : 28](#_Toc439163238)

[Etape 4 – Activation de l’option ZAZ dans les macros MYRMIDON : 28](#_Toc439163239)

[Etape 5 – Ouvrir le développeur Visual Basic. 29](#_Toc439163240)

[Etape 6 – Emplacement des macros : PRINCIPAL. 29](#_Toc439163241)

[Etape 7 – Activation du code ZAZ dans MYRMIDON Lettrage. 29](#_Toc439163242)

[Etape 8 – Activation du code ZAZ dans PLANIFICATEUR. 36](#_Toc439163243)

[Etape 9 – Facultatif – Lancer automatiquement ZAZ Stopmacro à l’ouverture fichier. 37](#_Toc439163244)

# Comment : afficher l'onglet Développeur sur le ruban

*Source :* [*https://msdn.microsoft.com/fr-fr/library/Bb608625.aspx*](https://msdn.microsoft.com/fr-fr/library/Bb608625.aspx)

*Source :* [*https://msdn.microsoft.com/fr-fr/library/office/Ee814737%28v=office.14%29.aspx*](https://msdn.microsoft.com/fr-fr/library/office/Ee814737%28v=office.14%29.aspx)

Pour accéder à l'onglet **Développeur** sur le ruban d'une application Office, vous devez configurer cette dernière pour afficher l'onglet, car il n'apparaît pas par défaut. Par exemple, vous devez afficher cet onglet pour ajouter un GroupContentControl à une personnalisation au niveau du document pour Excel.

info,information,about



**Remarque**

Ces recommandations **s'appliquent aux applications Office 2013 et Office 2010 uniquement**, consultez la version suivante de cette rubrique : Comment : afficher l'onglet Développeur sur le ruban.

**S'applique à** : les informations contenues dans cette rubrique s'appliquent aux projets de niveau document pour les applications suivantes : Excel, Outlook, PowerPoint, Word. Pour plus d'informations, voir Fonctionnalités disponibles par type d'application et de projet Office.

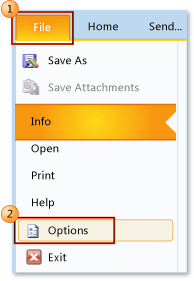
## Pour afficher l'onglet Développeur

**1** Démarrez l'une des applications Office prises en charge dans cette rubrique. Reportez-vous à la remarque **S'applique à** :, précédemment dans cette rubrique.

**2** Sous l'onglet **Fichier**, choisissez le bouton **Options**.

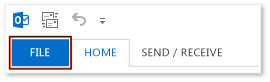
L'illustration suivante montre l'onglet **Fichier** et le bouton **Options** dans Office 2010.

## Figure 1 : Bouton Options dans Office 2010



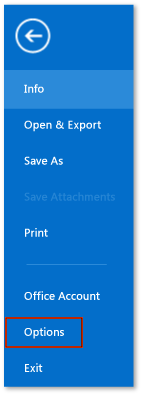
L'illustration suivante montre l'onglet **Fichier** dans Office 2013.

## Figure 2 : Onglet Fichier dans Office 2013



L'illustration suivante montre le bouton **Options** dans Office 2013.

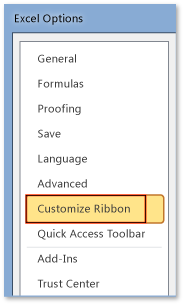
## Figure 3 : Bouton Options dans Office 2013



**3** Dans la boîte de dialogue **Options** *Nom\_application*, choisissez le bouton **Personnaliser le Ruban**.

L'illustration suivante montre la boîte de dialogue **Options** et le bouton **Personnaliser le Ruban** dans Excel 2010. L'emplacement de ce bouton est similaire dans toutes les autres applications répertoriées dans la section « S'applique à » au début de cette rubrique.

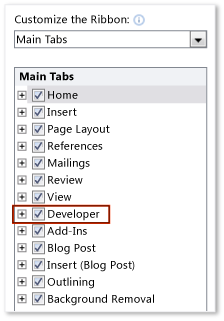
## Figure 4 : Bouton Personnaliser le Ruban dans Excel 2010



**4** Dans la liste des onglets principaux, cochez la case **Développeur**.

L'illustration suivante montre la case à cocher **Développeur** dans Excel 2010 et Excel 2013. L'emplacement de cette case à cocher est similaire dans toutes les autres applications répertoriées dans la section « S'applique à » au début de cette rubrique.

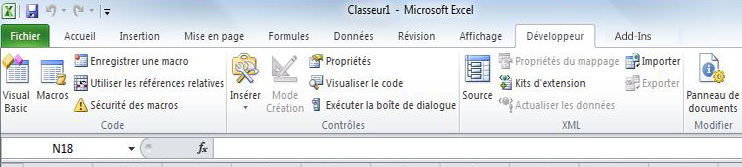
## Figure 5 : Case à cocher Développeur dans Excel 2010 et Excel 2013



**5** Choisissez le bouton **OK** pour fermer la boîte de dialogue **Options**.

Lorsque l’onglet **Développeur** est affiché, notez l’emplacement des boutons **Visual Basic**, **Macros** et **Sécurité** **des macros** sur celui-ci.

## Figure 1. Onglet Développeur dans Excel 2010



## Visual Basic Editor

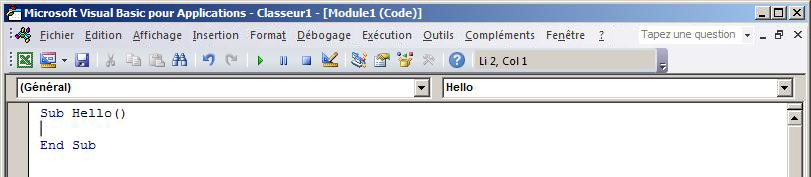
La procédure suivante vous montre comment l’éditeur Visual Basic.

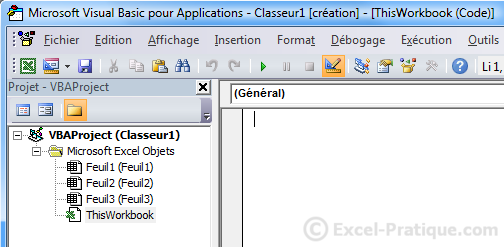
**Pour ouvrir l’éditeur Visual Basic**

**1** Cliquez sur le bouton **Visual Basic** sur l’onglet Développeur.

VBA est un langage de programmation complet doté d’un environnement de programmation complet correspondant. Cet article étudie uniquement les outils qui vous aident à démarrer la programmation, ce qui exclut la plupart des outils de Visual Basic Editor.

## Figure 2. Visual Basic Editor

**

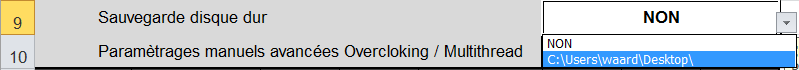


# COMMENT : MODIFIER OPTION SAUVEGARDE DISQUE DUR

Vous trouverez ci-après un exemple de procédure, pour modifier ou ajouter un lieu de sauvegarde dans le cadre de l’option de sauvegarde sur disque dur.

***Si vous modifiez cette option, vous devez le faire dans les fichiers MYRMIDON et PLANIFICATEUR.***

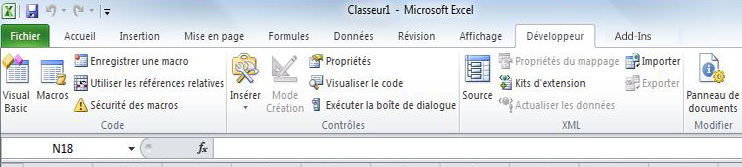




***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Sauvegarde disque dur », sont réinitialisées et prennent* ***par défaut*** *la valeur : «****Non****».*



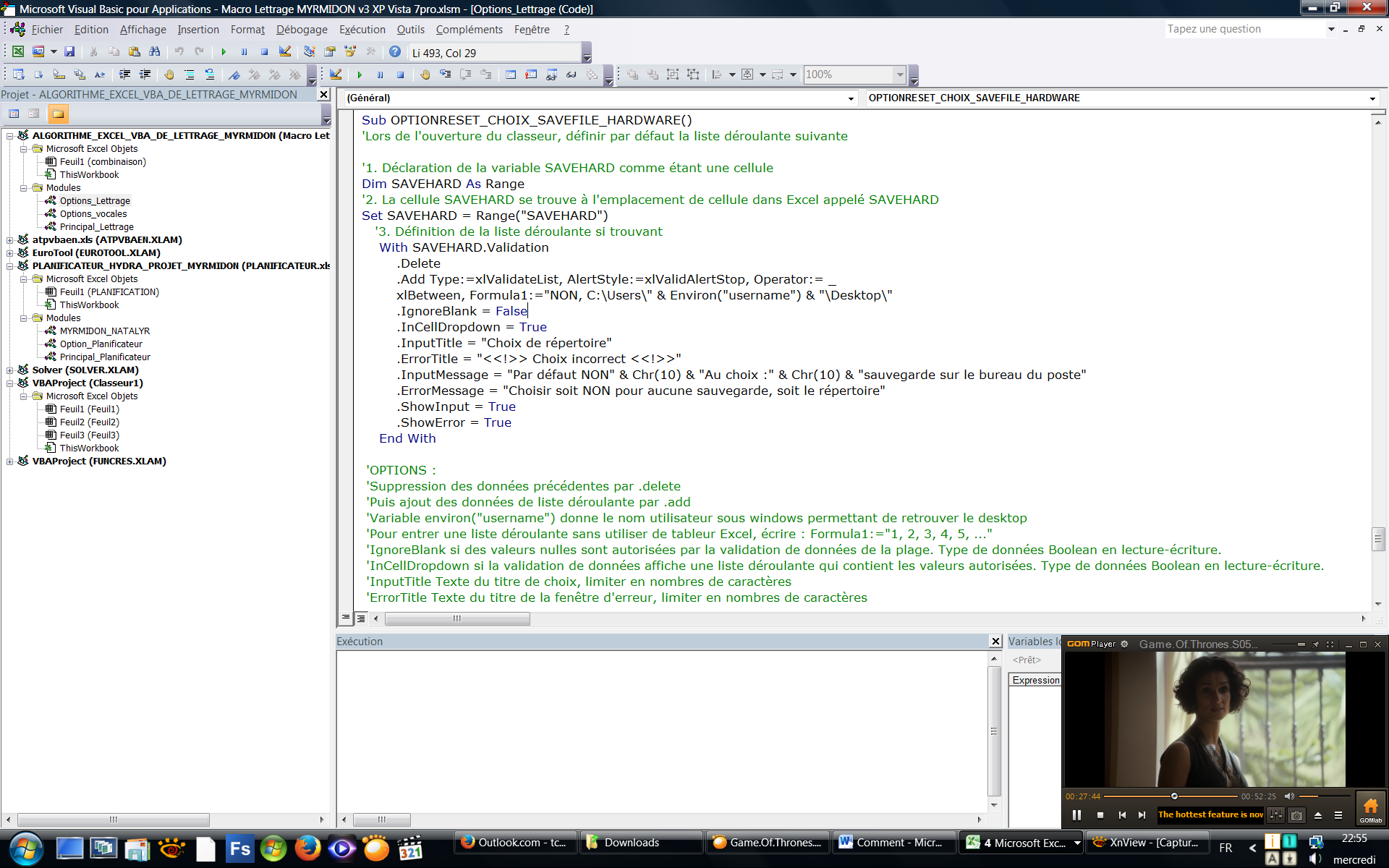
## Etape 1 – Ouvrir le développeur Visual Basic.



*Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*



## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTIONRESET\_CHOIX\_SAVEFILE\_HARDWARE

**2** **PLANIFICATEUR**

**>** Modules **>** Option\_Planificateur

**>** OPTIONRESET\_CHOIX\_SAVEHARDCOMBI

Les 2 macros sont identiques aux 2 fichiers et doivent le rester, afin de ne pas provoquer un bug dans la chaine de traitements du Planificateur, mais les noms de plage cellules ne sont pas identiques.



## Etape 3 – Description de certaines lignes de code.

Au niveau des lignes de codes suivantes, vous trouverez le type d’action réalisée par celles-ci :

Formula1:="NON, C:\Users\" & Environ("username") & "\Desktop\"

> Chemin d’accès des sauvegardes

InputTitle = "Choix de répertoire"

> Ligne de titre de la bulle commentaire

ErrorTitle = "<<!>> Choix incorrect <<!>>"

> Ligne de titre de la boite de message d’erreur

InputMessage = "Par défaut NON" & Chr(10) & "Au choix :" & Chr(10) & "sauvegarde sur le bureau du poste"

> Contenu de la bulle commentaire

ErrorMessage = "Choisir soit NON pour aucune sauvegarde, soit le répertoire"

> Contenu de la boite de message d’erreur

SAVEHARD.Value = "NON"

> Valeur par défaut, à chaque ouverture du fichier

## Etape 4 – Modification ou Ajout de chemin d’accès.

Formula1:="NON, C:\Users\" & Environ("username") & "\Desktop\"

Le code fonctionne suivant le principe **Formula1:="Chemin1, Chemin2, Chemin3, Chemin4, etc"**



**1** L’ensemble de la chaine des chemins doit rester entre les caractères : **" "**, si vous voulez ajouter une variable, il faut la fermer par **"**, ajouter **&**, mettre votre variable (ici le nom de l’utilisateur), ajouter **&**, mettre le reste du chemin et refermer la chaine avec **"**.

**2** Le chemin1, s’applique par défaut, ici le chemin1 est : « **NON** », pour donner une valeur pour garder inactive l’option lors du lancement de la recherche.

**3** Vous pouvez ajouter autant de chemin, que vous voulez.

**4** Pour disposer d’une variable nom d’utilisateur, utiliser par exemple :

**Environ("username")**, qui rapportera le nom utilisateur du PC.ou encore,

**application.username**, rapportant le nom utilisateur sous Excel.

* Ne pas oublier le caractère* ***\*** *à la fin du chemin.*

## Etape 5 – Chemin par défaut, à l’ouverture du fichier.

SAVEHARD.Value = "NON"

Si vous souhaitez qu’à l’ouverture du fichier, avoir par défaut un chemin d’accès spécifique, remplacer « NON », par le chemin que vous souhaitez, en veillant à garder le choix entre les caractères : **" "**.

 Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « Sauvegarde disque dur », en allant dans :

> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_CHOIX\_SAVEFILE\_HARDWARE **dans le fichier MYRMIDON**

OPTIONRESET\_CHOIX\_SAVEHARDCOMBI **dans le fichier Planificateur**

## Etape 6 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’éxécution en appuyant sur le bouton : « **Lecture** », pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre cellule d’option a été réinitialisée avec les nouvelles options de chemin d’accès. 

# COMMENT : MODIFIER OPTION ENVOI PAR MAIL

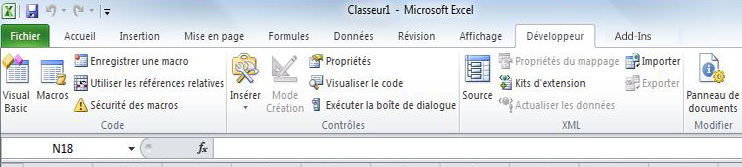
Vous trouverez ci-après 4 exemples de procédure, pour modifier la méthode d’envoi d’E-mail dans le cadre de l’option Envoi résultat(s) par E-Mail.

*** Si vous modifiez cette option, vous devez le faire dans les fichiers MYRMIDON et PLANIFICATEUR.***



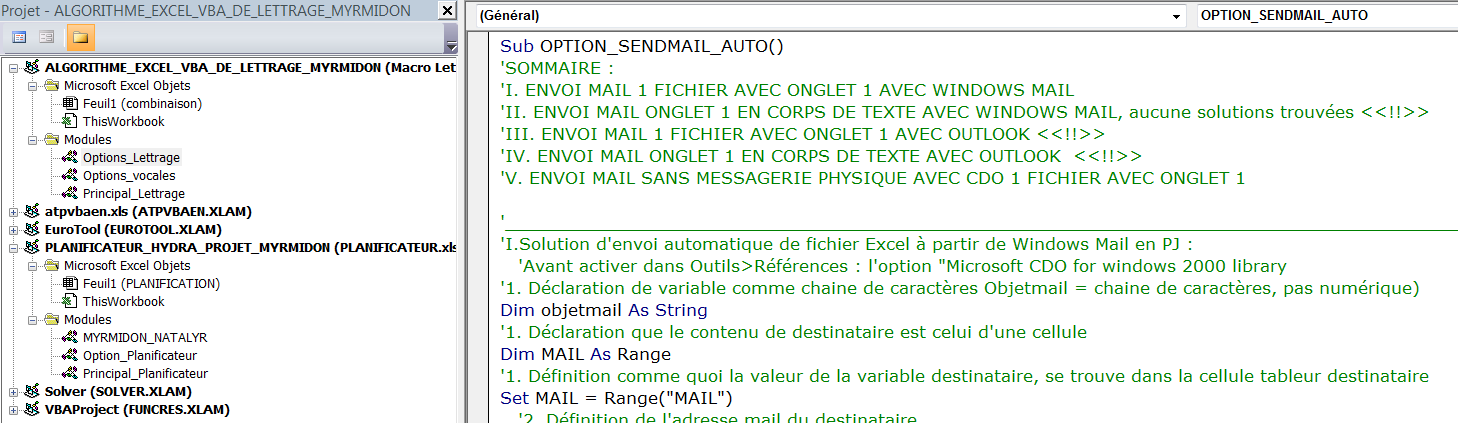
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Envoi résultat(s) par E-Mail », sont réinitialisées et prennent* ***par défaut*** *la valeur : « ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTION\_SENDMAIL\_AUTO

**2** **PLANIFICATEUR**

**>** Modules **>** Option\_Planificateur

**>** OPTION\_SENDMAIL\_PLANIFICATEUR

 Les 2 macros sont identiques aux 2 fichiers et doivent le rester, afin de ne pas provoquer un bug dans la chaine de traitements du Planificateur, mais les noms de plage cellules ne sont pas identiques.

## Etape 3 – Description de certains codes préparés.

'I. ENVOI MAIL 1 FICHIER AVEC ONGLET 1 AVEC WINDOWS MAIL

> Code permettant d’envoyer un mail avec le classeur en P.J. avec Windows Mail

'II. ENVOI MAIL ONGLET 1 EN CORPS DE TEXTE AVEC WINDOWS MAIL, aucune solutions trouvées <<!!>>

> **⦸**

'III. ENVOI MAIL 1 FICHIER AVEC ONGLET 1 AVEC OUTLOOK <<!!>>

> Code permettant d’envoyer un mail avec le classeur en P.J. à l’aide d’Outlook

'IV. ENVOI MAIL ONGLET 1 EN CORPS DE TEXTE AVEC OUTLOOK <<!!>>

> Code permettant d’envoyer un mail avec le résultat en corps de mail. à l’aide d’Outlook

'V. ENVOI MAIL SANS MESSAGERIE PHYSIQUE AVEC CDO 1 FICHIER AVEC ONGLET 1

> Code permettant d’envoyer un mail avec le classeur en P.J. à l’aide du serveur de messagerie ou d’internet

** Il ne peut y avoir qu’un choix de code d’activé, au risque d’un bug.

 Tous les codes sont préparés et dispose de leur variable de fonctionnement.

## Etape 4 – Les variables communes.

Dim objetmail As String

Objetmail > Sert de raccourci afin de fixer le formatage du titre du mail

Dim MAIL As Range

MAIL > Sert de raccourci, fait référence à la cellule : « Envoi résultat(s) par E-Mail ».

Portant le nom de cellule **MAIL** dans le fichier Myrmidon et **MAILPLANIFICATEUR** dans le Planificateur.

## Etape 5 – Caractère d’activation de ligne de code.

Le caractère **‘** permet de désactiver le code, celui sera considéré par VBA, comme étant du texte.

* Les lignes de texte, ne doivent surtout pas être activées, sous peine d’erreurs.*

## Etape 6 – Solution d'envoi automatique de fichier Excel à partir de Windows Mail en PJ.

Code avec lignes de code activé, pour exemple dans le fichier Myrmidon.



'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'I.Solution d'envoi automatique de fichier Excel à partir de Windows Mail en PJ :

'Avant activer dans Outils>Références : l'option "Microsoft CDO for windows 2000 library

'1. Déclaration de variable comme chaine de caractères Objetmail = chaine de caractères, pas numérique)

Dim objetmail As String

'1. Déclaration que le contenu de destinataire est celui d'une cellule

Dim MAIL As Range

'1. Définition comme quoi la valeur de la variable destinataire, se trouve dans la cellule tableur destinataire

Set MAIL = Range("MAIL")

'2. Définition de l'adresse mail du destinataire

MAIL = MAIL.Value

'3. Définition de l'intitulé de l'objet du mail, (dans WindowsMail le symbole € n'est pas pris en charge, remplacer par E)

objetmail = "Résultats des Combinaisons de" & " " & Format(Range("E1").Value, "# ##0.00") & "E" & " - " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'4. Controle pour envoi mail si adresse mail écrite dans la cellule destinataire, >> sinon FIN MACRO

If Not IsEmpty(MAIL) Then

'5. Création d'une copie excel temporaire pour l'envoyer par mail (seulement l'onglet 1)

ActiveWorkbook.worksheets(1).Copy

'6. Envoi auto du fichier en P.J. "Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38

ActiveWorkbook.SendMail MAIL, objetmail

'7. Fermeture du fichier excel temporaire sans le sauvegarder

ActiveWorkbook.Close SaveChanges:=False

End If

'----------------------------------------------------------------------------------------------------------------------------------------------------

 Code avec lignes de code activé, pour exemple dans le fichier Planificateur.

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'I.Solution d'envoi automatique de fichier Excel à partir de Windows Mail en PJ :

'Avant <!> activer dans Outils>Références : l'option "Microsoft CDO for windows 2000 library

'1. Déclaration de variable comme chaine de caractères Objetmail = chaine de caractères, pas numérique)

Dim objetmail As String

'1. Déclaration que le contenu de destinataire est celui d'une cellule

Dim MAIL As Range

'1. Définition comme quoi la valeur de la variable destinataire, se trouve dans la cellule tableur destinataire

Set MAIL = Range("MAILPLANIFICATEUR")

'2. Définition de l'adresse mail du destinataire

MAIL = MAIL.Value

'3. Définition de l'intitulé de l'objet du mail, (dans WindowsMail le symbole € n'est pas pris en charge, remplacer par E)

objetmail = "Résultats de la planification du " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'4. Controle pour envoi mail si adresse mail écrite dans la cellule destinataire, >> sinon FIN MACRO

If Not IsEmpty(MAIL) Then

'5. Création d'une copie excel temporaire pour l'envoyer par mail (seulement l'onglet 1)

ActiveWorkbook.Worksheets(1).Copy

'6. Envoi auto du fichier en P.J. "Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38

ActiveWorkbook.SendMail MAIL, objetmail

'7. Fermeture du fichier excel temporaire sans le sauvegarder

ActiveWorkbook.Close savechanges:=False

End If

'----------------------------------------------------------------------------------------------------------------------------------------------------

**1** **Objetmail** est formaté pour donner ce titre : *Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38*

> Chaque partie de texte doit être en caractére : **«  »**

> A chaque variable, pour récupérer une date, nom utilisateur, valeur cellule, vous devez fermer la chaine de texte à l’aide **«**  et ajouté entre la chaine de texte et la variable, le caractère **&**

**2** Le morceau de code worksheets(1), fait référence à une action sur la feuille 1 du classeur actif, si vous vouliez la feuille 2, il faudrait écrire worksheets(2), pour faire référence à toutes les feuilles du classeur, il faudrait écrire worksheets.

**3** L’action SaveChanges:=False, False pour interdire la sauvegarde avant la fermeture du fichier temporaire envoyé par mail ou True pour sauvegarder.

**4** Pour l’activation, aller dans **Outils > Références**, cocher l'option : "Microsoft CDO for windows 2000 library"

**5** Dans le planificateur, l’objet range(« **Mail** »), doit faire référence à la place "**MAILPLANIFICATEUR**".

** Il ne peut y avoir qu’un choix de code d’activé, au risque d’un bug.

## Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 par P.J. avec Outlook.

 Code avec lignes de code activé, pour exemple dans le fichier Myrmidon.

'----------------------------------------------------------------------------------------------------------------------------------------------------

'III. ENVOI MAIL 1 FICHIER AVEC ONGLET 1 AVEC OUTLOOK - << !! NON TESTE, PAS OUTLOOK !! >>

'Working in Excel 2000-2013

'For Tips see: http://www.rondebruin.nl/win/winmail/Outlook/tips.htm

'1. Déclaration de variable comme chaine de caractères (adresse mail + Objet = chaine de caractères, pas numérique) et object d'application

Dim OutApp As Object

Dim OutMail As Object

Dim MAIL As range

Dim objetmail As String

'2. Définition des variables particulières

Set OutApp = CreateObject("Outlook.Application")

Set OutMail = OutApp.CreateItem(0)

set MAIL = Range("MAIL").Value

objetmail = "Résultats des Combinaisons de" & " " & Format(Range("E1").Value, "# ##0.00") & "E" & " - " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'3. Création d'une copie excel temporaire pour l'envoyer par mail (seulement l'onglet 1)

If Not IsEmpty(MAIL) Then

ActiveWorkbook.worksheets(1).Copy

'5. Envoi auto du fichier en P.J. "Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38

With OutMail

.to = MAIL

.Subject = objetmail

.Body = "Résultat en P.J. du traitement combinatoire finalisé"

.Attachments.Add (ActiveWorkbook.Fullname) 'ou essayer : ActiveWorkbook.Path & "/" & activeworkbook.name

.Send

End With

'6. Fermeture du fichier excel temporaire sans le sauvegarder

ActiveWorkbook.Close SaveChanges:=False

End if 'Fin If not isempty(mail)

'7. Fermeture des objets d'application créés

Set OutMail = Nothing

Set OutApp = Nothing

'----------------------------------------------------------------------------------------------------------------------------------------------------

 Code avec lignes de code activé, pour exemple dans le fichier Planificateur.

'-----------------------------------------------------------------------------------------------------------------------------------------------------

'III. ENVOI MAIL 1 FICHIER AVEC ONGLET 1 AVEC OUTLOOK - << !! NON TESTE, PAS OUTLOOK CHEZ MOI !! >>

'Working in Excel 2000-2013

'For Tips see: http://www.rondebruin.nl/win/winmail/Outlook/tips.htm

'1. Déclaration de variable comme chaine de caractères (adresse mail + Objet = chaine de caractères, pas numérique) et object d'application

Dim OutApp As Object

Dim OutMail As Object

Dim MAIL As range

Dim objetmail As String

'2. Définition des variables particulières

Set OutApp = CreateObject("Outlook.Application")

Set OutMail = OutApp.CreateItem(0)

set MAIL = Range("MAILPLANIFICATEUR")

objetmail = "Résultats de la planification du " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'3. Création d'une copie excel temporaire pour l'envoyer par mail (seulement l'onglet 1), Si Option activé par user

If Not IsEmpty(MAIL) Then

ActiveWorkbook.worksheets(1).Copy

'5. Envoi auto du fichier en P.J. "Résultats de la planification du 02/11/2014 - 16H38

With OutMail

.to = MAIL.value

.Subject = objetmail

.Body = "Planificateur en P.J. de vos traitements combinatoires réalisés"

.Attachments.Add (ActiveWorkbook.Fullname) 'ou essayer : ActiveWorkbook.Path & "/" & activeworkbook.name

.Send

End With

'6. Fermeture du fichier excel temporaire sans le sauvegarder

ActiveWorkbook.Close SaveChanges:=False

End if 'Fin If not isempty(mail)

'7. Fermeture des objets d'application créés

Set OutMail = Nothing

Set OutApp = Nothing

'-----------------------------------------------------------------------------------------------------------------------------------------------------

**1** **Objetmail** est formaté pour donner ce titre : *Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38*

> Chaque partie de texte doit être en caractère : **«  »**

> A chaque variable, pour récupérer une date, nom utilisateur, valeur cellule, vous devez fermer la chaine de texte à l’aide **«**  et ajouté entre la chaine de texte et la variable, le caractère **&**

**2** Le morceau de code worksheets(1), fait référence à une action sur la feuille 1 du classeur actif, si vous vouliez la feuille 2, il faudrait écrire worksheets(2), pour faire référence à toutes les feuilles du classeur, il faudrait écrire worksheets.

**3** **.Body** Permet d’indiquer un message en corps de mail, fonctionne comme **Objetmail**.

**4** L’action SaveChanges:=False, False pour interdire la sauvegarde avant la fermeture du fichier temporaire envoyé par mail ou True pour sauvegarder.

**5** Pour l’activation, aller dans **Outils > Références**, cocher l'option : " Microsoft Outlook Library "

**6** Dans le planificateur, l’objet range(« **Mail** »), doit faire référence à la place "**MAILPLANIFICATEUR**".

** Il ne peut y avoir qu’un choix de code d’activé, au risque d’un bug.

** Code non testé, car Outlook non possédé pour les tests, mais en théorie, fonctionnel.

## Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 en corps de texte avec Outlook.

 Code avec lignes de code activé, pour exemple dans le fichier Myrmidon.

'----------------------------------------------------------------------------------------------------------------------------------------------------

'IV. ENVOI MAIL ONGLET 1 EN CORPS DE TEXTE AVEC OUTLOOK

'<<!!>> NON TESTE, PAS OUTLOOK CHEZ MOI, FONCTIONNE EN 2 ROUTINES

'http://www.meilleurereponse.net/excel/contenu-dune-feuille-dans-le-corps-dun-mail-excel-vba/

'+++++++++++++<<!!>>+FONCTIONNE+AVEC+1+SUB+ET+1+FONCTION+<<!!>>++++++++++++++++

Sub OPTION\_SENDMAIL\_AUTO()

'1. Déclaration des variables destinataire/objetmail comme chaine de caractères, rng comme tableur, caractérisation des variables objets : Outapp et outmail

Dim MAIL As range

Dim objetmail As String

Dim rng As Range

Dim OutApp As Outlook.Application

Dim OutMail As Outlook.MailItem

Application.DisplayAlerts = False

'2. Caractérisation des variables contenu dans des lots/application

Set rng = Nothing

Set rng = ActiveSheet ‘selection des cellules à envoyer dans le message ou Activesheet.range("A1:E20") au choix

Set OutApp = CreateObject(« Outlook.Application »)

Set OutMail = OutApp.CreateItem(olMailItem)

'3. Définition de la variable destinataire et objetmail

MAIL = Range("MAIL").Value

objetmail = "Résultats des Combinaisons de" & " " & Format(Range("E1").Value, "# ##0.00") & "E" & " - " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'4. Envoi du mail avec en corps de message le tableur de la feuille 1

If Not IsEmpty(MAIL) Then

With OutMail

.to = MAIL

.Subject = objetmail

.HTMLBody = RangetoHTML(rng) ‘on appelle la fonction rangetohtml pour les cellules sélectionnées

.send

End With

End if 'Fin If not isempty(mail)

End Sub

'+++++++++++++AVEC+PROCEDURE+LIEE+++++++++++++++++

Function RangetoHTML(rng As Range)

'1. Déclaration des variables

Dim fso As Object

Dim ts As Object

Dim TempFile As String

'2. creation d’un fichier htm vide qui servira à coller les données

TempFile = "C:\meilleurereponse\Temp.htm" 'PENSER A CHANGER L'EMPLACEMENT TEMP

'3. copier les cellules que l’on veut envoyer dans une nouvelle feuille

Sheets.Add

ActiveSheet.Name = "temp"

rng.Copy

Sheets("temp").Select

ActiveSheet.Cells(1).Select

ActiveSheet.Paste

'4. copier les données collées dans la feuille vide dans le fichier htm

With ActiveWorkbook.PublishObjects.Add( \_

SourceType:=xlSourceRange, \_

Filename:=TempFile, \_

Sheet:= "temp", \_

Source:=ActiveWorkbook.Sheets("temp").UsedRange.Address, \_

HtmlType:=xlHtmlStatic)

.Publish (True)

End With

'5. rangetohtml = données du fichier temporaire créé

Set fso = CreateObject("Scripting.FileSystemObject")

Set ts = fso.GetFile(TempFile).OpenAsTextStream(1, -2)

RangetoHTML = ts.ReadAll

ts.Close

RangetoHTML = Replace(RangetoHTML, "align=center x:publishsource=", \_

"align=left x:publishsource=")

'6. 'on supprime le fichier temporaire

Kill TempFile

'7. Fermeture des objets

Set ts = Nothing

Set fso = Nothing

End Function

'----------------------------------------------------------------------------------------------------------------------------------------------------

 Code avec lignes de code activé, pour exemple dans le fichier Planificateur.

'-----------------------------------------------------------------------------------------------------------------------------------------------------

'IV. ENVOI MAIL ONGLET 1 EN CORPS DE TEXTE AVEC OUTLOOK

'<<!!>> NON TESTE, PAS OUTLOOK CHEZ MOI, FONCTIONNE EN 2 ROUTINES

'http://www.meilleurereponse.net/excel/contenu-dune-feuille-dans-le-corps-dun-mail-excel-vba/

'+++++++++++++<<!!>>+FONCTIONNE+AVEC+1+SUB+ET+1+FONCTION+<<!!>>++++++++++++++++

Sub OPTION\_SENDMAIL\_AUTO()

'1. Déclaration des variables destinataire/objetmail comme chaine de caractères, rng comme tableur, caractérisation des variables objets : Outapp et outmail

Dim MAIL As range

Dim objetmail As String

Dim rng As Range

Dim OutApp As Outlook.Application

Dim OutMail As Outlook.MailItem

Application.DisplayAlerts = False

'2. Caractérisation des variables contenu dans des lots/application

Set rng = Nothing

Set rng = ActiveSheet ‘selection des cellules à envoyer dans le message ou Activesheet.range("A1:E20") au choix

Set OutApp = CreateObject(« Outlook.Application »)

Set OutMail = OutApp.CreateItem(olMailItem)

'3. Définition de la variable destinataire et objetmail

MAIL = Range("MAILPLANIFICATEUR ").Value

objetmail = "Résultats de la planification du " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'4. Envoi du mail avec en corps de message le tableur de la feuille 1, si option activé par user

If Not IsEmpty(MAIL) Then

With OutMail

.to = MAIL

.Subject = objetmail

.HTMLBody = RangetoHTML(rng) ‘on appelle la fonction rangetohtml pour les cellules sélectionnées

.send

End With

End if

End Sub

'+++++++++++++AVEC+PROCEDURE+LIEE+++++++++++++++++

Function RangetoHTML(rng As Range)

'1. Déclaration des variables

Dim fso As Object

Dim ts As Object

Dim TempFile As String

'2. creation d’un fichier htm vide qui servira à coller les données

TempFile = "C:\meilleurereponse\Temp.htm" 'PENSER A CHANGER L'EMPLACEMENT TEMP

'3. copier les cellules que l’on veut envoyer dans une nouvelle feuille

Sheets.Add

ActiveSheet.Name = "temp"

rng.Copy

Sheets("temp").Select

ActiveSheet.Cells(1).Select

ActiveSheet.Paste

'4. copier les données collées dans la feuille vide dans le fichier htm

With ActiveWorkbook.PublishObjects.Add( \_

SourceType:=xlSourceRange, \_

Filename:=TempFile, \_

Sheet:= "temp", \_

Source:=ActiveWorkbook.Sheets("temp").UsedRange.Address, \_

HtmlType:=xlHtmlStatic)

.Publish (True)

End With

'5. rangetohtml = données du fichier temporaire créé

Set fso = CreateObject("Scripting.FileSystemObject")

Set ts = fso.GetFile(TempFile).OpenAsTextStream(1, -2)

RangetoHTML = ts.ReadAll

ts.Close

RangetoHTML = Replace(RangetoHTML, "align=center x:publishsource=", \_

"align=left x:publishsource=")

'6. 'on supprime le fichier temporaire

Kill TempFile

'7. Fermeture des objets

Set ts = Nothing

Set fso = Nothing

End Function

'-----------------------------------------------------------------------------------------------------------------------------------------------------

**1** **Objetmail** est formaté pour donner ce titre : *Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38*

> Chaque partie de texte doit être en caractère : **«  »**

> A chaque variable, pour récupérer une date, nom utilisateur, valeur cellule, vous devez fermer la chaine de texte à l’aide **«**  et ajouté entre la chaine de texte et la variable, le caractère **&**

**2** Le morceau de code **Set rng = ActiveSheet**, fait référence à une action sur la feuille active du classeur, si vous vouliez la feuille 2, il faudrait écrire worksheets(2), pour faire référence à une plage de cellules, il faudrait écrire : Activesheet.range("A1:E20").

**3** **TempFile** Pour faire le transvasement du contenu des cellules avec le formatage cellule, une copie de ce qui doit être traité est créé, dans un dossier temporaire, pour l’exemple, celui utilise ce chemin, si vous le souhaitez, vous pouvez personnaliser ce chemin, mais en veillant à bien garder Temp.htm à la fin du chemin, ce fichier temporaire est détruit à la fin du traitement de manière automatique.

**4** Pour l’activation, aller dans **Outils > Références**, cocher l'option : " Microsoft Outlook Library "

**5** Dans le planificateur, l’objet range(« **Mail** »), doit faire référence à la place "**MAILPLANIFICATEUR**".

** Il ne peut y avoir qu’un choix de code d’activé, au risque d’un bug.

** Code non testé, car Outlook non possédé pour les tests, mais en théorie, fonctionnel.

## Etape 6 – Envoi mail en 1 fichier avec l’onglet 1 en P.J. par serveur internet ou de messagerie.

 Code avec lignes de code activé, pour exemple dans le fichier Myrmidon.

'-----------------------------------------------------------------------------------------------------------------------------------------------------

'V. ENVOI MAIL SANS MESSAGERIE PHYSIQUE AVEC CDO 1 FICHIER AVEC ONGLET 1

'1. Déclaration de variable comme chaine de caractères Objetmail = chaine de caractères, pas numérique)

Dim objetmail As String

'1. Déclaration que le contenu de destinataire est celui d'une cellule

Dim MAIL As Range

'1. Déclaration pour envoi mail avec methode CDO

Dim mMessage As Object

Dim mConfig As Object

Dim mChps

'1. Définition comme quoi la valeur de la variable destinataire, se trouve dans la cellule tableur destinataire

Set MAIL = Range("MAIL")

'1. Définition création d'objet CDO

Set mConfig = CreateObject("CDO.Configuration")

'2. Définition de l'adresse mail du destinataire

MAIL = MAIL.Value

'2. Définition de l'intitulé de l'objet du mail, (dans WindowsMail le symbole € n'est pas pris en charge, remplacer par E)

objetmail = "Résultats des Combinaisons de" & " " & Format(Range("E1").Value, "# ##0.00") & "E" & " - " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

‘4. Controle pour envoi mail si adresse mail écrite dans la cellule destinataire, >> sinon FIN MACRO

if Not IsEmpty(MAIL) Then

'5. Désactivation des messages de sécurité, pour permettre l'automatisation sans passer par l'utilisateur

Application.DisplayAlerts = False

'5. Définition du serveur pour l'envoi

mConfig.Load -1

Set mChps = mConfig.Fields

With mChps

.Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2

‘Adapter suivant votre serveur de messagerie mail physique ou serveur internet. HoTmail => "smtp.live.com" si connexion numéricable => "smtp.numericable.fr" Orange => "smtp.orange.fr"

.Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = "smtp.numericable.fr"

'En principe, 25 fonctionne avec tout les serveurs.

.Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 25

'!>> Vous pouvez essayer sans ces trois lignes <<!

'Mais si votre serveur demande une authentification,

'.Item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate") = "1"

'.Item("http://schemas.microsoft.com/cdo/configuration/sendusername") = "MyMail@gmail.com" 'votre adresse mail entre parenthèses

'.Item("http://schemas.microsoft.com/cdo/configuration/sendpassword") = "MyCodePass" 'le mot de passe de votre adresse mail entre parenthèses

'Si votre serveur demande une connexion sûre (SSL), sinon laisser en option risque de bug

'.Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") = "true"

.Update

End With

'6. Procédure d'envoi du mail

'6. Création d'une copie excel temporaire pour l'envoyer par mail

'6.a Déclaration de variable, le fichier de base s'appel fichier source

Dim FichierSource As Workbook

'6.b Déclaration de variabale, nom du fichier à envoyer

Dim FichierPourMail As Workbook

'6.c Déclaration de variable comme texte titre

Dim Temporaire As String

'6.d Déclaration de variable comme texte titre

Dim NOMFichierPourMail As String

'6.e définit l'emplacement du fichier source

Set FichierSource = ActiveWorkbook

'6.f copie du fichier source pour créer une copie du fichier source à envoyer

ActiveSheet.Copy

'6.g emplacement du fichier a envoyer, correspondant au fichier copier ouvert

Set FichierPourMail = ActiveWorkbook

'6.h Définition de la variable "temporaire" : emplacement nom et format fichier

Temporaire = ThisWorkbook.Path & Application.PathSeparator & "Résultat recherche" & ThisWorkbook.FileFormat

'6.i sauvegarde du fichier a envoyer avec le titre temporaire

FichierPourMail.SaveAs Temporaire

'6.j Définition de la variable "nomfichierpourmail" pour l'envoi dans vba

NOMFichierPourMail = FichierPourMail.Path & Application.PathSeparator & FichierPourMail.Name

'6.k Fermeture du fichier à envoyer pour pouvoir le supprimer plus tard

FichierPourMail.Close

'6.L. Création d'un objet mail mmessage

Set mMessage = CreateObject("CDO.Message")

With mMessage 'Définition critères pour l'envoi du message

Set .Configuration = mConfig 'avec la configuration précédemment déclarée dans 5

.To = MAIL 'Envoi mail à destinataire adresse dans les options

.From = "Combinaison@MYRMIDON.com" 'Adresse expéditeur imaginaire ou non pour envoi du mail

.Subject = objetmail

.TextBody = "Message automatique suite à votre demande, contenant le résultat de votre recherche"

‘''Chemin et nom complet du fichier à joindre

.AddAttachment NOMFichierPourMail

.Send

End With

'7. suppression du fichier temporaire a envoyer par mail

Kill NOMFichierPourMail

'7. Libèration des ressources mémoires

Set mMessage = Nothing

Set mConfig = Nothing

Set mChps = Nothing

'7. Réactivation des messages de sécurité

Application.DisplayAlerts = True

End If ‘FIN if Not IsEmpty(MAIL) Then

'-----------------------------------------------------------------------------------------------------------------------------------------------------

 Code avec lignes de code activé, pour exemple dans le fichier Planificateur.

'-----------------------------------------------------------------------------------------------------------------------------------------------------

'V. ENVOI MAIL SANS MESSAGERIE PHYSIQUE AVEC CDO 1 FICHIER AVEC ONGLET 1

'1. Déclaration de variable comme chaine de caractères Objetmail = chaine de caractères, pas numérique)

Dim objetmail As String

'1. Déclaration que le contenu de destinataire est celui d'une cellule

Dim MAIL As Range

'1. Déclaration pour envoi mail avec methode CDO

Dim mMessage As Object

Dim mConfig As Object

Dim mChps

'1. Définition comme quoi la valeur de la variable destinataire, se trouve dans la cellule tableur destinataire

Set MAIL = Range("MAILPLANIFICATEUR")

'1. Définition création d'objet CDO

Set mConfig = CreateObject("CDO.Configuration")

'2. Définition de l'adresse mail du destinataire

MAIL = MAIL.Value

'2. Définition de l'intitulé de l'objet du mail, (dans WindowsMail le symbole € n'est pas pris en charge, remplacer par E)

objetmail = "Résultats de la planification du " & Format(Date, "dd/mm/yyyy") & " - " & Format(Time, "h\Hmm")

'4. Controle pour envoi mail si adresse mail écrite dans la cellule destinataire, >> sinon FIN MACRO

If Not IsEmpty(MAIL) Then

'5. Désactivation des messages de sécurité, pour permettre l'automatisation sans passer par l'utilisateur

Application.DisplayAlerts = False

'5. Définition du serveur pour l'envoi

mConfig.Load -1

Set mChps = mConfig.Fields

With mChps

.Item("http://schemas.microsoft.com/cdo/configuration/sendusing") = 2

'Adapter suivant votre serveur de messagerie mail physique ou serveur internet. HoTmail => "smtp.live.com" si connexion numéricable => "smtp.numericable.fr" Orange => "smtp.orange.fr"

.Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = "smtp.numericable.fr"

'En principe, 25 fonctionne avec tout les serveurs.

.Item("http://schemas.microsoft.com/cdo/configuration/smtpserverport") = 25

'!>> Vous pouvez essayer sans ces trois lignes <<!

'Mais si votre serveur demande une authentification,

'.Item("http://schemas.microsoft.com/cdo/configuration/smtpauthenticate") = "1"

'.Item("http://schemas.microsoft.com/cdo/configuration/sendusername") = "MyMail@gmail.com" 'votre adresse mail entre parenthèses

'.Item("http://schemas.microsoft.com/cdo/configuration/sendpassword") = "MyCodePass" 'le mot de passe de votre adresse mail entre parenthèses

'Si votre serveur demande une connexion sûre (SSL), sinon laisser en option risque de bug

'.Item("http://schemas.microsoft.com/cdo/configuration/smtpusessl") = "true"

.Update

End With

'6. Procédure d'envoi du mail

'6. Création d'une copie excel temporaire pour l'envoyer par mail

'6.a Déclaration de variable, le fichier de base s'appel fichier source

Dim FichierSource As Workbook

'6.b Déclaration de variabale, nom du fichier à envoyer

Dim FichierPourMail As Workbook

'6.c Déclaration de variable comme texte titre

Dim Temporaire As String

'6.d Déclaration de variable comme texte titre

Dim NOMFichierPourMail As String

'6.e définit l'emplacement du fichier source

Set FichierSource = ActiveWorkbook

'6.f copie du fichier source pour créer une copie du fichier source à envoyer

ActiveSheet.Copy

'6.g emplacement du fichier a envoyer, correspondant au fichier copier ouvert

Set FichierPourMail = ActiveWorkbook

'6.h Définition de la variable "temporaire" : emplacement nom et format fichier

Temporaire = ThisWorkbook.Path & Application.PathSeparator & "Résultat recherche" & ThisWorkbook.FileFormat

'6.i sauvegarde du fichier a envoyer avec le titre temporaire

FichierPourMail.SaveAs Temporaire

'6.j Définition de la variable "nomfichierpourmail" pour l'envoi dans vba

NOMFichierPourMail = FichierPourMail.Path & Application.PathSeparator & FichierPourMail.Name

'6.k Fermeture du fichier à envoyer pour pouvoir le supprimer plus tard

FichierPourMail.Close

'6.L. Création d'un objet mail mmessage

Set mMessage = CreateObject("CDO.Message")

With mMessage 'Définition critères pour l'envoi du message

Set .Configuration = mConfig 'avec la configuration précédemment déclarée dans 5

.To = MAIL 'Envoi mail à destinataire adresse dans les options

.From = "Combinaison@MYRMIDON.com" 'Adresse imaginaire ou non pour envoi du mail

.Subject = objetmail

.TextBody = "Message automatique suite à votre demande, contenant les résultats de votre planification"

'''Chemin et nom complet du fichier à joindre

.AddAttachment NOMFichierPourMail

.Send

End With

'7. suppression du fichier temporaire a envoyer par mail

Kill NOMFichierPourMail

'7. Libèration des ressources mémoires

Set mMessage = Nothing

Set mConfig = Nothing

Set mChps = Nothing

'7. Réactivation des messages de sécurité

Application.DisplayAlerts = True

End If ‘FIN if Not IsEmpty(MAIL) Then

'-----------------------------------------------------------------------------------------------------------------------------------------------------

**1** **Objetmail** est formaté pour donner ce titre : *Résultats des Combinaisons de 222 000,22€ - 02/11/2014 - 16H38*

> Chaque partie de texte doit être en caractère : **«  »**

> A chaque variable, pour récupérer une date, nom utilisateur, valeur cellule, vous devez fermer la chaine de texte à l’aide **«**  et ajouté entre la chaine de texte et la variable, le caractère **&**

**2** .**Item("http://schemas.microsoft.com/cdo/configuration/smtpserver") = "smtp.numericable.fr"**, permet d’indiquer la source serveur utilisée pour l’envoi du mail, si vous passez par votre serveur internet, il faudrait écrire par exemple :

smtp.numericable.fr

smtp.orange.fr

smtp.sfr.fr

smtp.bbox.fr

Ou par votre serveur de messagerie, par exemple :

smtp.live.com

smtp.gmail.com

*Pour plus d’adresse de serveur, voir ce lien :*



<http://www.commentcamarche.net/faq/893-parametres-de-serveurs-pop-imap-et-smtp-des-principaux-fai>

**3** Le morceau de code **ActiveSheet**.**copie**, fait référence à une action de création d’une copie de la feuille active du classeur (soit la feuill1), si vous vouliez la feuille 2, il faudrait écrire worksheets(2).copy, pour faire référence à toutes les feuilles du classeur, il faudrait écrire worksheets.copy.

**4** .**From** Adresse de l’expéditeur, par simplification et pour rendre flexible en cas d’utilisateur multiple, vous pouvez mettre une adresse imaginaire, mais en ayant tout de même un sens pour pouvoir repérer l’expéditeur facilement dans votre boite mail, l’adresse n’a pas besoin d’être valide, cela n’empêche pas l’envoi.

**5** **.TextBody** Permet d’indiquer un message en corps de mail, fonctionne comme **Objetmail**.

**6** Fonctionne sans, mais si erreur référence, alors activé la référence CDO, pour l’activation, aller dans **Outils > Références**, cocher l'option : " Microsoft CDO for Windows 2000 Library "

**6** Dans le planificateur, l’objet range(« **Mail** »), doit faire référence à la place "**MAILPLANIFICATEUR**".

** Il ne peut y avoir qu’un choix de code d’activé, au risque d’un bug.

## Etape 7 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option d’envoi a été modifiée. 

## Etape 8 – Modification de la réinitialisation de l’adresse mail à l’ouverture du classeur.

Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « d’envoi Mail », en allant dans :



> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_MAIL **dans le fichier MYRMIDON**

OPTIONRESET\_MAILPLANIFICATEUR **dans le fichier Planificateur**

 Pour modifier la valeur **d’adresse mail, à l’ouverture** du classeur, aller dans :

**Dans le fichier MYRMIDON** > Module Option\_lettrage > OPTIONRESET\_MAIL

Range("F7:H7").ClearContents > Pour effacer

Range("MAIL").value = « *votreadressemail@mail.com* » > Pour indiquer une adresse mail par défaut

**Dans le fichier Planificateur** > Module Options\_planificateur > OPTIONRESET\_MAILPLANIFICATEUR

Range("MAILPLANIFICATEUR").ClearContents > Pour effacer

Range("MAILPLANIFICATEUR").value = « *votreadressemail@mail.com* » > Pour indiquer l’adresse mail par défaut

## Etape 9 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option de valeur par défaut d’option d’envoi par mail a été modifiée. 

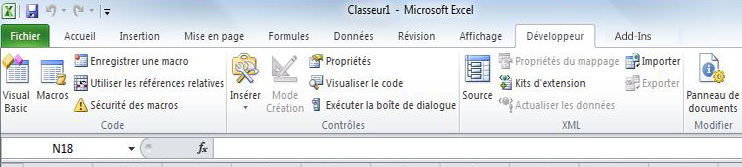
# COMMENT : MODIFIER REINITIALISATION OPTION CHOIX NB COMBI

Vous trouverez ci-après un exemple de procédure, pour modifier la valeur par défaut lors de l’ouverture du classeur, dans le cadre de l’option Choix du nombre de combinaison à rechercher.



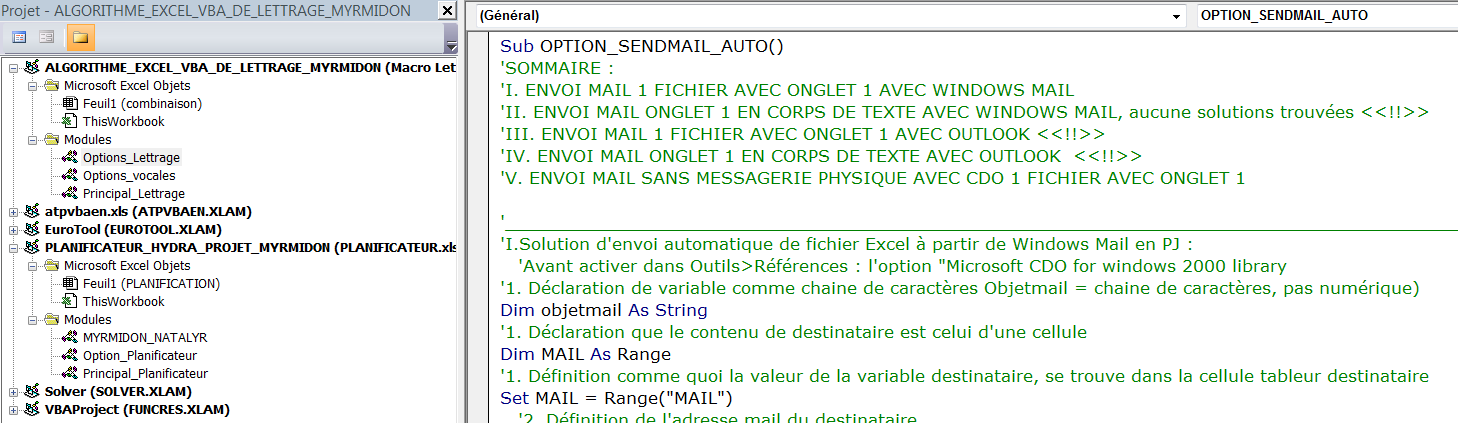
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Choix du nombre de combinaison à rechercher », sont réinitialisées et prennent* ***par défaut*** *la valeur : « 1 ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTIONRESET\_CHOIX\_NB\_COMBI

## Etape 3 – Modification de la réinitialisation du choix du nb combi. à l’ouverture du classeur.

 Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « Choix du nombre de combinaison à rechercher», en allant dans :

> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_CHOIX\_NB\_COMBI **dans le fichier MYRMIDON**

 Pour modifier la valeur **du nombre de choix de combinaison à l’ouverture** du classeur, aller dans :

**Dans le fichier MYRMIDON** > Module Option\_lettrage > OPTIONRESET\_CHOIX\_NB\_COMBI

A la ligne : **choix.Value = 1** > Pour 1 combinaison, peut prendre une valeur comprise entre 1 et 3.

## Etape 4 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option de valeur par défaut de choix combi. a été modifiée. 

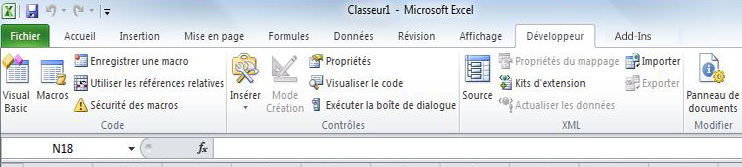
# COMMENT : MODIFIER REINITIALISATION OPTION LIMITE PERMU MINI

Vous trouverez ci-après un exemple de procédure, pour modifier la valeur par défaut lors de l’ouverture du classeur, dans le cadre de l’option : « Limiter nombre de cellules minimum à permuter ».



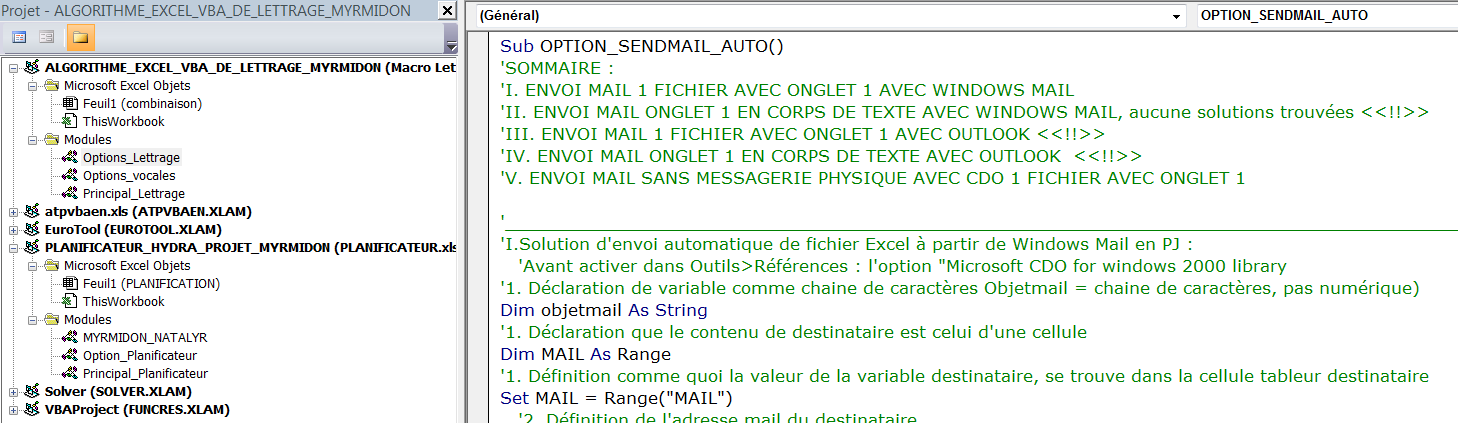
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Limiter nombre de cellules minimum à permuter », sont réinitialisées et prennent* ***par défaut*** *la valeur : « 1 ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTIONRESET\_NBPERMUMINI

## Etape 3 – Modification de la réinitialisation du nb de permutation mini. à l’ouverture du classeur.

 Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « « Limiter nombre de cellules minimum à permuter », en allant dans :

> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_NBPERMUMINI **dans le fichier MYRMIDON**

 Pour modifier la valeur **du nombre de permutation minimum à l’ouverture** du classeur, aller dans :

**Dans le fichier MYRMIDON** > Module Option\_lettrage > OPTIONRESET\_NBPERMUMINI

A la ligne : **NBPERMUMINI.Value = 1** > Pour 1 cellule permutée, pas de limite à ce niveau, sauf lors du calcul.

## Etape 4 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option de limite de permutation minimum. a été modifiée. 

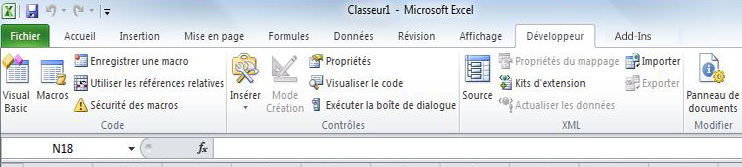
# COMMENT : MODIFIER REINITIALISATION OPTION LIMITE PERMU MAXI

Vous trouverez ci-après un exemple de procédure, pour modifier la valeur par défaut lors de l’ouverture du classeur, dans le cadre de l’option : « Limiter nombre de cellules maximum à permuter ».



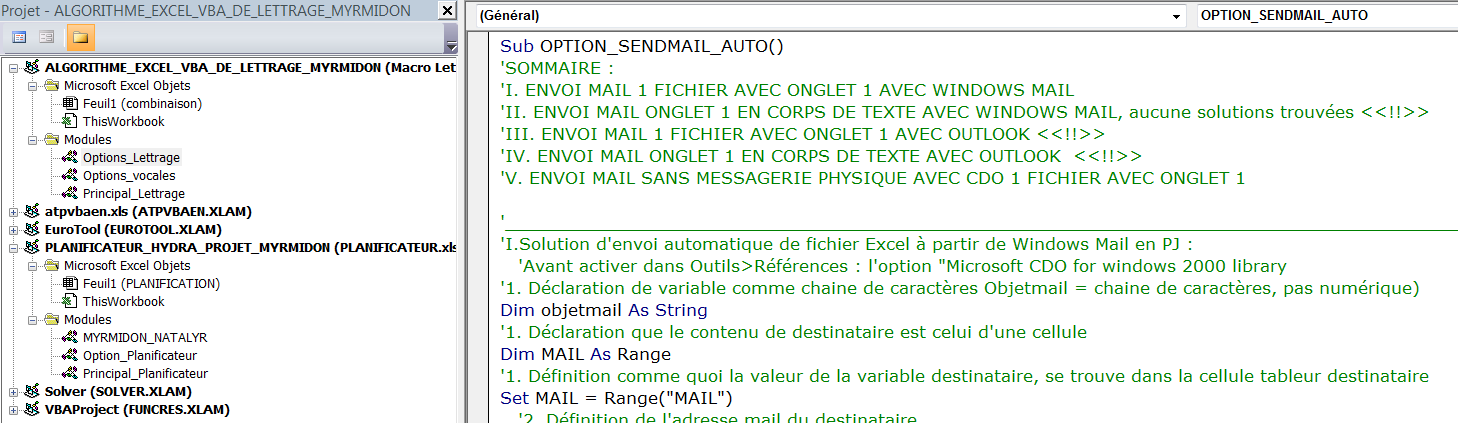
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Limiter nombre de cellules maximum à permuter », sont réinitialisées et prennent* ***par défaut*** *la valeur : « 3 ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTIONRESET\_NBPERMU

## Etape 3 – Modification de la réinitialisation du nb de permutation maxi. à l’ouverture du classeur.

 Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « « Limiter nombre de cellules maximum à permuter », en allant dans :

> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_NBPERMU **dans le fichier MYRMIDON**

 Pour modifier la valeur **du nombre de permutation maximum à l’ouverture** du classeur, aller dans :

**Dans le fichier MYRMIDON** > Module Option\_lettrage > OPTIONRESET\_NBPERMU

A la ligne : **NBPERMU.Value = 3** > Pour 3 cellules permutées, pas de limite à ce niveau, sauf lors du calcul.

## Etape 4 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option de limite de permutation maximum a été modifiée. 

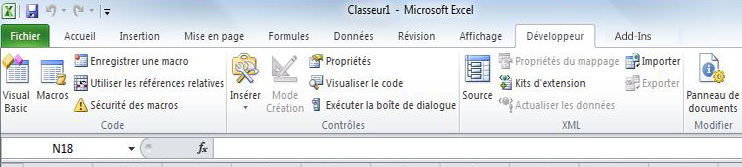
# COMMENT : MODIFIER REINITIALISATION OPTION ARRET PAR MINUTEUR

Vous trouverez ci-après un exemple de procédure, pour modifier la valeur par défaut lors de l’ouverture du classeur, dans le cadre de l’option : « Arrêt par minuteur ».



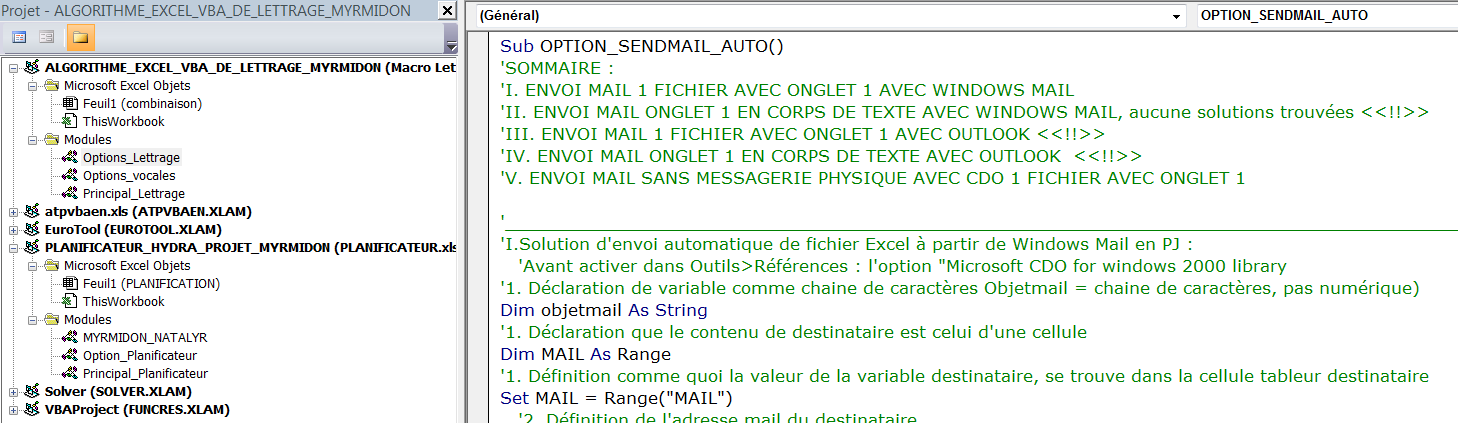
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : « Arrêt par minuteur », sont réinitialisées et prennent* ***par défaut*** *la valeur : « 00 : 00 : 00 ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** Modules **>** Options\_Lettrage

**>** OPTIONRESET\_CHOIX\_STOPTIME

**2** **PLANIFICATEUR**

**>** Modules **>** Option\_Planificateur

**>** OPTIONRESET\_CHOIX\_STOPCOMBI

La modification dans le planificateur est optionnelle.



## Etape 3 – Modification de la réinitialisation du minuteur à l’ouverture du classeur.

 Vous pouvez **annuler l’option de réinitialisation** du contenu de la cellule « Arrêt par minuteur », en allant dans :

> Dans ThisWorkbook > Private Sub Workbook\_Open > Supprimer la ligne :

OPTIONRESET\_NBPERMU **dans le fichier MYRMIDON**

OPTIONRESET\_CHOIX\_STOPCOMBI **dans le fichier Planificateur**

 Pour modifier la valeur **de l’arrêt par minuteur à l’ouverture** du classeur, aller dans :

**Dans le fichier MYRMIDON** > Module Option\_lettrage > OPTIONRESET\_CHOIX\_STOPTIME

A la ligne : **Range("STOPTIME").FormulaR1C1 = "00:00:00"** > Pour 00 :00 :00 pour valeur inactive, 00 : 00 : 10 pour 10 secondes par défaut, par exemple.

**Dans le fichier Planificateur** > Module Option\_lettrage > OPTIONRESET\_CHOIX\_STOPCOMBI

A la ligne : **Range("STOPCOMBI").FormulaR1C1 = "00:00:00"** > Pour 00 :00 :00 pour valeur inactive, 00 : 00 : 10 pour 10 secondes par défaut, par exemple.

## Etape 4 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

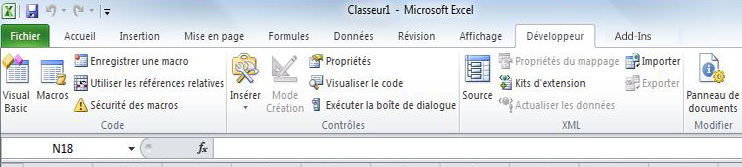
Voilà, votre option de valeur par défaut d’arrêt par minuteur. a été modifiée. 

# COMMENT : SUPPRIMER L’INTERDICTION DE CREATION DE FEUILLE

Vous trouverez ci-après un exemple de procédure, pour supprimer l’interdiction de création de feuille dans les classeurs.

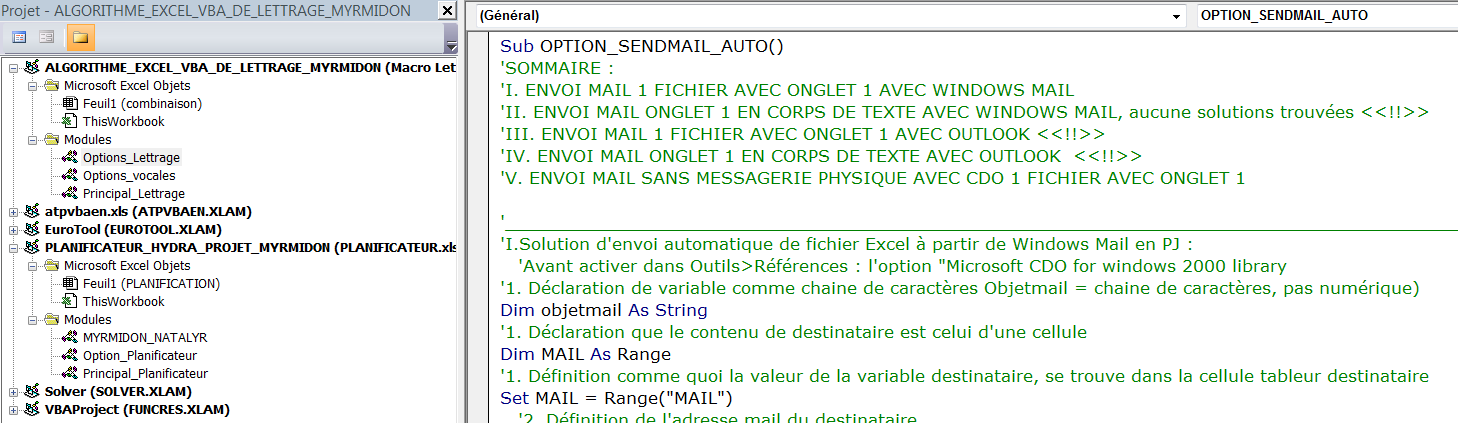
** Cette interdiction est pensée dans le cadre d’un traitement par chaine, vous ne devriez pas supprimer cette option.

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

**1** **MYRMIDON**

**>** ThisWorkbook **>** Workbook\_NewSheet

**2** **PLANIFICATEUR**

**>** Modules **>** Workbook\_NewSheet

## Etape 3 – Suppression de l’interdiction.

Supprimer l’ensemble de ces lignes :

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'I. Empêcher à l'utilisateur lambda la création d'onglet dans ce classeur

Private Sub Workbook\_NewSheet(ByVal Sh As Object)

'1. Désactivation des messages d'alerte de sécurité

Application.DisplayAlerts = False

'2. Action auto suppression de l'onglet créé

Sh.Delete

'3. Réactivation des messages d'alerte de sécurité

Application.DisplayAlerts = True

'4. Message d'info à User, avec indication si vraiment le besoin, de ce qu'il faut supprimer pour le faire

MsgBox "Création d'onglet désactivée par défaut." & Chr(10) & "Pour la réactivée, supprimer cette SUB dans ThisWorkbook : Private Sub Workbook\_NewSheet", vbInformation, "INFORMATION MYRMIDON"

End Sub

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Voilà, votre option d’interdiction de création de feuille, a été modifiée. 

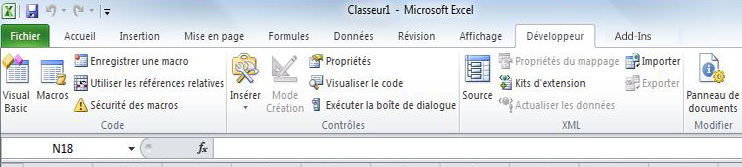
# COMMENT : MODIFIER DATE D’ARRET PAR DEFAUT DU PLANIFICATEUR

Vous trouverez ci-après 1 exemple de procédure, pour modifier la valeur par défaut, dans le cadre de l’option « Arrêt planification au plus tard ».



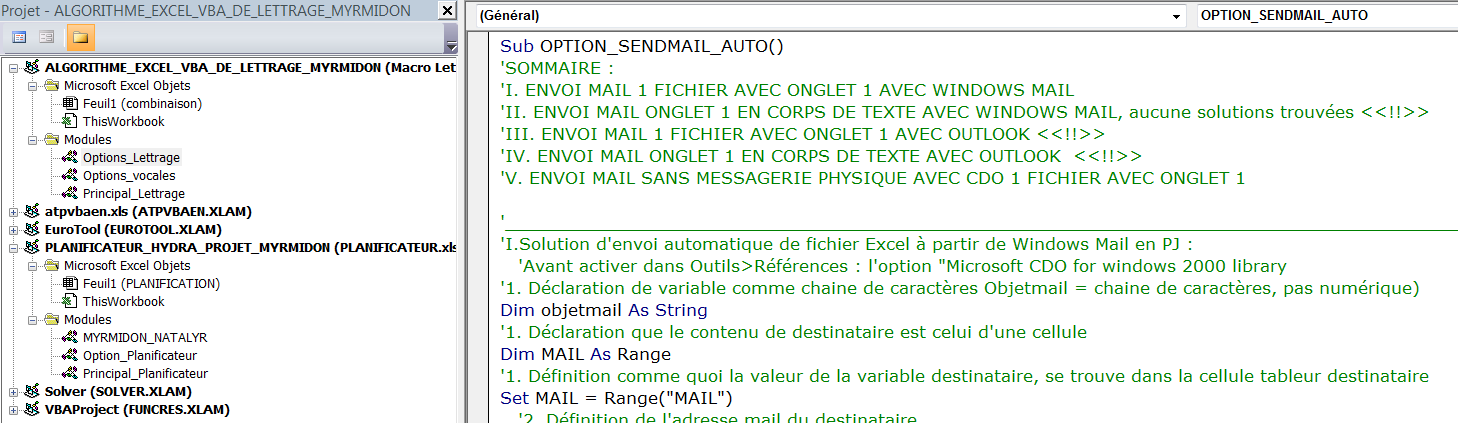
** ***A chaque ouverture*** *des fichiers, les valeurs contenues dans l’option : «*Arrêt planification au plus tard *», sont réinitialisées et prennent* ***par défaut*** *la valeur : « Aujourd’hui + 12h ».*

## Etape 1 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 2 – Emplacement des macros : OPTIONS.



** Vous trouverez les macros en cause dans :

1 **PLANIFICATEUR**

**>** Modules **>** Option\_Planificateur

**>** OPTIONRESET\_CHOIX\_STOPPLANIFICATEUR

## Etape 3 – Modification de la valeur date d’arrêt par planification, par défaut.

Le code initial dans la macro : OPTIONRESET\_CHOIX\_STOPPLANIFICATEUR est :

'1. RAZ par Ajout date/heure d'ouverture du fichier et RAZ du format d'affichage en JJ.MM.AAAA HH:mm

Range("STOPPLANIFICATEUR") = Format(Now, "dd/mm/yyyy hh:mm")

'2. Ajoute 12heures par défaut pour l'arrêt des travaux planifiés

Range("STOPPLANIFICATEUR").FormulaR1C1 = DateAdd("h", 12, Now)

**1** Le morceau de code **DateAdd("h", 12, Now)**, ajoute 12h par défaut.

La variable : **"d"** ajoute des heure

La variable : **"h"** ajoute des heure

La variable : **"n"** ajoute des minutes

La variable : **"s"** ajoute des secondes

A la place de 12, vous pouvez ajouter, n’importer quel **nombre entier**, de préférence limité au concept de temps affilié (minute (1 à 60) / heure (1 à 24) / jour (1 à 31).

**2** Si vous voulez la date et l’heure d’ouverture, écrire : Range("STOPPLANIFICATEUR").FormulaR1C1 = Now

## Etape 4 – Vérification et validation.

Si vous avez bien suivi la procédure, vous ne devriez pas avoir d’anomalie, pour valider vos modifications, vérifier l’exécution en appuyant sur le bouton : « **Lecture** » ou F8, pour avancer pas à pas, pour lancer la macro.

Si erreur, vérifier vos éléments avec cette procédure, pour voir où vous vous trompez.

Voilà, votre option de valeur par défaut d’arrêt par planificateur a été modifiée. 

# COMMENT : INSTALLER ZAZ StopMacro

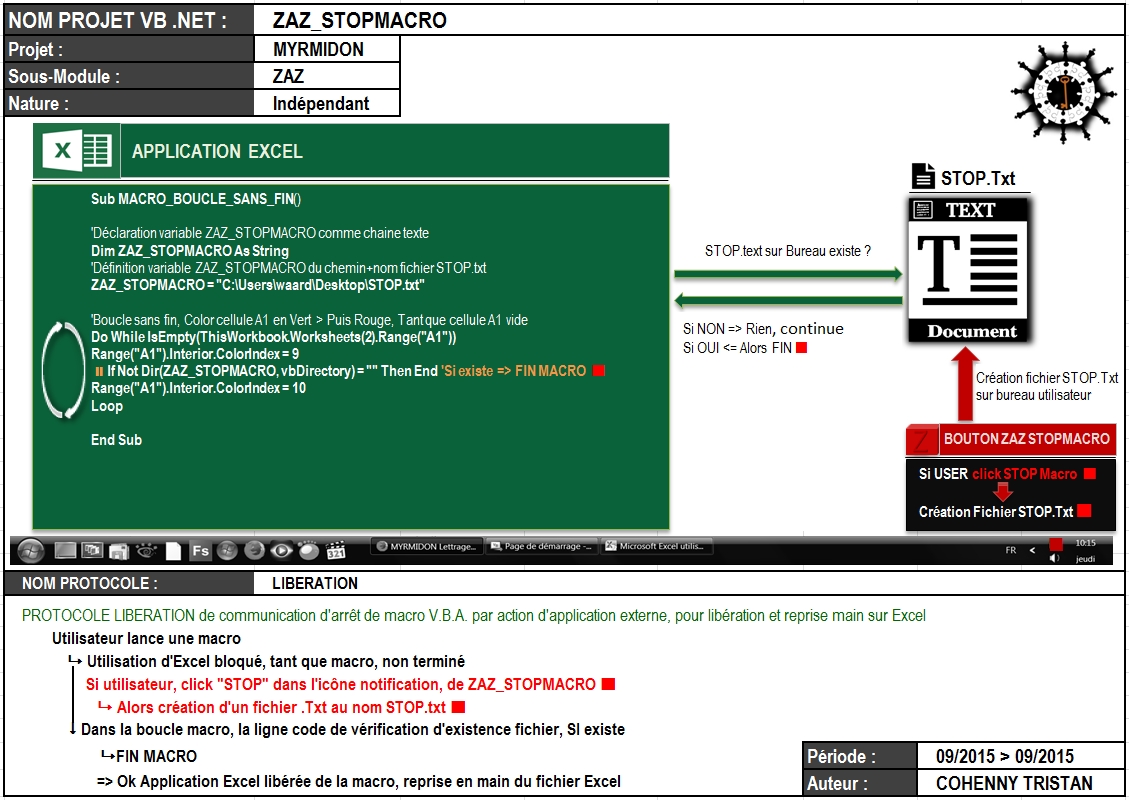
Vous trouverez ci-après la procédure d’installation de ZAZ StopMacro, afin de créer un interrupteur pour les macros MYRMIDON et PLANIFICATEUR.

*L’application peut être utilisée avec d’autres macros.*



 *Chapitre un peu long, mais c’est pour bien être certain que vous fassiez la procédure correctement, il s’agit seulement de vérifier le fonctionnement, puis de supprimer des ‘ devant chaque occurrence VBA « ZAZ », ça prend moins de 5 minutes.*

## Figure 1 : Présentation



Le fonctionnement est simple, une fois lancé une boucle macro, Excel sera occupé tant que la macro ne sera pas terminée, or si vous avez besoin de reprendre la main sur l'application :

- Soit vous avez prévu une **action de sortie** (*comme l’utilisation de ZAZ StopMacro*),

- Soit vous réussissez à afficher le **Debugger** (*Raccourci* ***Ctrle + Pause***),

- Soit vous l'arrêtez par le **gestionnaire de tâches** et vous perdez tout le travail déjà réalisé.

Ainsi, au lieu de pratiquer comme cela, vous pouvez lancer une action **STOP**, à l'aide de l'application, votre code V.B.A étant codé pour interpréter cette action, réalisera l'arrêt et Excel sera libéré et vous pourrez y reprendre la main, sans pertes.

**Les seuls paramètres à prendre en compte :**

**>** Que l'application ZAZ StopMacro, crée bien un fichier .txt, au nom de STOP.txt, sur votre bureau utilisateur.

**>** Que votre code V.B.A. vérifie bien sur votre bureau utilisateur, l'existence du fichier STOP.txt, afin d'être raccord avec ZAZ.

**>** D'éviter de créer des fichiers .txt, au nom de "STOP" pendant l’exécution des macros, mais ce doit être rare...

**>** D'éviter de lancer la macro, si le fichier STOP est toujours présent sur votre bureau, d'où l'intérêt de prévoir sa suppression dans votre code, une fois l'application arrêtée par ZAZ.

## Etape 1 – Télécharger l’application :

 L’application est **gratuite**, programmée en langage VB .Net, le code source est archivé sur fichier .txt.

 Le dossier de l’application peut être placé n’importe où, il n’y a pas besoin d’installation.

info,information,about **Lien de téléchargement :**

<http://www.excel-pratique.com/fr/telechargements/utilitaires/zaz-stopmacro-no213.php>

## Etape 2 – Forum de suivi :

 Un forum de suivi existe, sur lequel est présent masterclass, exemples de code,…

info,information,about **Lien de téléchargement :**

<http://forum.excel-pratique.com/applications/zaz-stopmacro-bouton-interrupteur-externe-t68644.html>

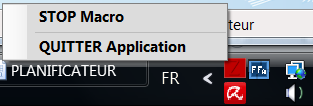
## Etape 3 – Lancement et 1er test ZAZ StopMacro :

**1** - Pour lancer l’application, cliquer sur le fichier application : **ZAZ STOPMACRO ▶** I:\Travail\Projets Spéciaux\Projet ZAZ STOPMACRO\Logo, Présentation\screen zaz stopmacro grand.jpg



 L’application se place dans la barre de notification.

**2** – Tester que l'application, crée bien un fichier .txt, au nom de *STOP.txt*, sur votre bureau utilisateur.



 Faire un clic droit sur l’icône ZAZ I:\Travail\Projets Spéciaux\Projet ZAZ STOPMACRO\Logo, Présentation\screen zaz stopmacro grand.jpg et cliquer sur le bouton : **STOP MACRO**

**3** – Un fichier *STOP.txt* doit maintenant s’être créé sur votre bureau utilisateur.



 Le code VB .Net crée ce fichier à cette adresse : Environ("userprofile") & "\desktop\STOP.txt", donc adresse du dossier utilisateur de la session windows ouverte, le seul problème pouvant survenir, serait que votre bureau ne s’appelle pas desktop/bureau.

**info,information,about Information :**

Si à partir de là, l’application ne fonctionne pas ou que vous ne retrouvez pas le fichier, inutile d’aller plus loin.

La moitié du travail est terminé. 

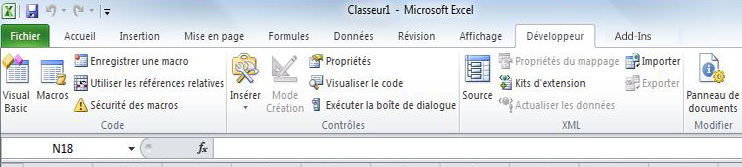


## Etape 4 – Activation de l’option ZAZ dans les macros MYRMIDON :

Ouvrir les fichiers Excel : MYRMIDON Lettrage / PLANIFICATEUR

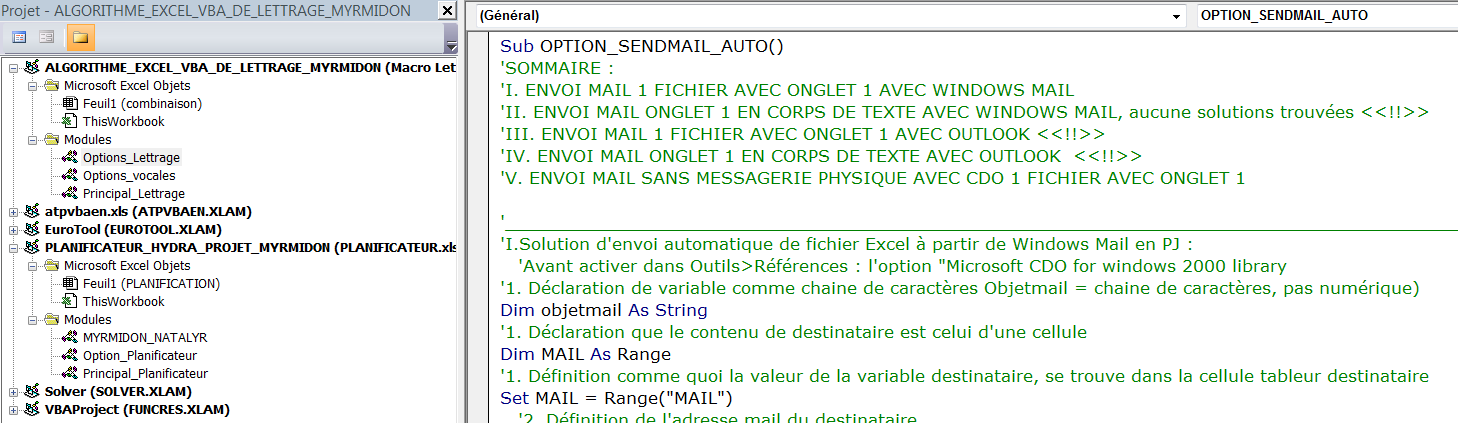
 Le code VBA gérant ZAZ est préparé, mais nécessite d’y insérer votre chemin propre menant au fichier STOP.Text et d’activer le code.

## Etape 5 – Ouvrir le développeur Visual Basic.



* Excel et le Développeur Visual Basic dispose chacun d’un* ***volet d’aide****, pouvant être appelé par la touche «****F1****», par contre, si l’application est identique aux 2 logiciels, la base de données n’est pas identique, ouvrir l’aide à partir d’Excel, ne permettra pas de vous apporter de réponse pour des problèmes VBA.*

## Etape 6 – Emplacement des macros : PRINCIPAL.



** Vous trouverez les macros à modifiées dans :

1 **MYRMIDON LETTRAGE**

**>** Modules **>** PRINCIPAL\_LETTRAGE

**>** ALGO\_COMBINATOIRE\_MYRMIDON

**>** recursivite

**>** OPTION\_ARRET\_PROG

**>** ALGO\_COMBINATOIRE\_MYRMIDON\_NUMERIQUE

**>** RECURSIVITE\_PERMUTATION

2 **PLANIFICATEUR**

**>** Modules **>** PRINCIPAL\_PLANIFICATEUR

**>** PLANIFICATION\_RECHERCH\_COMBI\_MYR

## Etape 7 – Activation du code ZAZ dans MYRMIDON Lettrage.

Le caractère **‘** permet de désactiver le code, celui sera considéré par VBA, comme étant du texte.

* Les lignes de texte, ne doivent surtout pas être activées, sous peine d’erreurs.*

*Vous pouvez utiliser la recherche texte Ctrl + F en recherchant le mot clef : « ZAZ » et aller directement sur les lignes concernées, pour supprimer le caractère* **‘** *devant la ligne de code ZAZ et l’ajouter sur la ligne de code de base se trouve en générale en dessous de la ligne de code ZAZ.*



**Dans le fichier MYRMIDON** > Module Principal\_lettrage > Ligne tout en haut **Public**

** Code inactivé de base

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'0. Variables Publiques communes à ce classeur

Public PERMUMAXI As Integer, TOTAL As Currency, CIBLE As Currency, DEPART As Variant, FIN As Variant, Compteur As Byte, TFIN As Date, TDEBUT As Date, TSTOPTIME As Date, CHOIX As Range, NBPERMU As Range, MONTANT As Range, STOPTIME As Range, CELCOLOR As Integer, NBPERMUMINI As Range, PERMUMINI As Integer, STOPDATEPLANI As Date, STOPINTERRUPTOR As Boolean, STOPERROR As Boolean, K As Integer, TABNUM() As Double, PLAGE\_SOL As Range, TAILLE\_PLAGE As Integer, CTRLCALCNUM As Boolean ', ZAZ\_STOPMACRO As String

** **Code activé -> Supprimer le caractère ‘ et l’espace devant, ZAZ\_STOPMACRO As String**

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'0. Variables Publiques communes à ce classeur

Public PERMUMAXI As Integer, TOTAL As Currency, CIBLE As Currency, DEPART As Variant, FIN As Variant, Compteur As Byte, TFIN As Date, TDEBUT As Date, TSTOPTIME As Date, CHOIX As Range, NBPERMU As Range, MONTANT As Range, STOPTIME As Range, CELCOLOR As Integer, NBPERMUMINI As Range, PERMUMINI As Integer, STOPDATEPLANI As Date, STOPINTERRUPTOR As Boolean, STOPERROR As Boolean, K As Integer, TABNUM() As Double, PLAGE\_SOL As Range, TAILLE\_PLAGE As Integer, CTRLCALCNUM As Boolean, ZAZ\_STOPMACRO As String

**Dans le fichier MYRMIDON** > Module Principal \_lettrage > PLANIFICATION\_RECHERCH\_COMBI\_MYR

** Code inactivé de base

'I. DECLARATION VARIABLES DE COMPTEURS POUR LA RECHERCHE

'Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

‘ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le chemin C:\Users\waard\Desktop\ le fichier stop.txt doit se trouver sur le bureau user

** Code activé

'I. DECLARATION VARIABLES DE COMPTEURS POUR LA RECHERCHE

'Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le chemin C:\Users\waard\Desktop\ le fichier stop.txt doit se trouver sur le bureau user

Changer l’adresse ci-après : « C:\Users\waard\Desktop\STOP.txt », par la vôtre où doit se trouver le fichier STOP.txt chez vous (en théorie dans les cas courant, seul votre nom d’utilisateur « waard » doit être différent de cette adresse), bien veiller à conserver les caractères « … » sous peine d’erreur.

**Dans le fichier MYRMIDON** > Module Principal \_lettrage > recursivite(debut)

** Code inactivé de base

'II. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

▶'If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation de ZAZ Stop macro

▶ If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

▶'If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation de ZAZ Stop macro

▶ If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Sub

End If

End If

End If

'III. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

▶ ‘If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

▶ ‘If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Sub

End If

End If

End If

** Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'II. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation de ZAZ Stop macro

‘If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation de ZAZ Stop macro

‘ If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Sub

End If

End If

End If

'III. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Sub

End If

End If

End If

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

**Dans le fichier MYRMIDON** > Module Principal \_lettrage > OPTION\_ARRET\_PROG

Code inactivé de base



'0.a. Process de sortie si date planifié ou timer atteint

▶ ‘If Now > TFIN Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > TFIN Or Now > STOPDATEPLANI Then 'Si temps en cours n'est pas au delà de la programmation alors arrêt

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.1. Procédure 0 combinaison trouvé

If Compteur <= 0 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "Aucune combinaison trouvée", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "NON" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

▶ ‘If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.2. Procédure 1 combinaison trouvé

If Compteur = 1 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "1 Combinaison trouvée", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

▶ ‘If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.3. Procédure 2 combinaison trouvé

If Compteur = 2 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "2 Combinaisons trouvées", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

▶ ‘If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.4. Procédure 3 combinaison trouvé

If Compteur = 3 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "3 Combinaisons trouvées", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

▶ ‘If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'0.b. Process de sortie si date planifié ou timer atteint

▶ ‘If Now > TFIN Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Now > TFIN Then 'Si temps en cours n'est pas au delà de la programmation alors arrêt

'1.3 ZAZ STOPMACRO Suppression du fichier STOP

▶ ‘If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then Kill ZAZ\_STOPMACRO 'Suppression fichier STOP.txt, pour éviter arrêt macro non voulu, si relance

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

 Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'0.a. Process de sortie si date planifié ou timer atteint

If Now > TFIN Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > TFIN Or Now > STOPDATEPLANI Then 'Si temps en cours n'est pas au delà de la programmation alors arrêt

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.1. Procédure 0 combinaison trouvé

If Compteur <= 0 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "Aucune combinaison trouvée", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "NON" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.2. Procédure 1 combinaison trouvé

If Compteur = 1 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "1 Combinaison trouvée", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.3. Procédure 2 combinaison trouvé

If Compteur = 2 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "2 Combinaisons trouvées", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'6.4. Procédure 3 combinaison trouvé

If Compteur = 3 Then

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'ARIANNE\_END\_RESEARCH 'Message vocale arianne d'avis de fin de calculs

info = MsgBox("Traitement terminé suite à arrêt programmé <<!>>" & Chr(10) & "3 Combinaisons trouvées", vbExclamation, "INFORMATION - TRAITEMENT ARRÊTÉ")

'Procédure message planifié

Else

Range("J1").Value = "OUI" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Range("J2").Value = "Combinaisons de" & " " & Format(Range("MONTANT").Value, "# ##0.00€") & " - " & Format(Date, "dd-mm-yyyy") & " - " & Format(Time, "h\Hmm") 'Pour reprise titre classeur dans planificateur

'Si arrêt suivant date planifié valeur spécifique en J5

If Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > STOPDATEPLANI Then

Range("J5").Value = "STOPP" 'Pour reprise dans planificateur, pour MàJ des indicateurs, valeur supprimé quant retour planificateur

Else

Range("J5").Value = "STOP" 'STOP Normal sur timer

End If

End If

End If

'0.b. Process de sortie si date planifié ou timer atteint

If Now > TFIN Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Now > TFIN Then 'Si temps en cours n'est pas au delà de la programmation alors arrêt

'1.3 ZAZ STOPMACRO Suppression du fichier STOP

If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then Kill ZAZ\_STOPMACRO 'Suppression fichier STOP.txt, pour éviter arrêt macro non voulu, si relance

**Dans le fichier MYRMIDON** > Module Principal \_lettrage > ALGO\_COMBINATOIRE\_MYRMIDON\_NUMERIQUE

 Code inactivé de base

'0. DECLARATION VARIABLES GENERALES

'Déclaration de variable d'adresse plage des combinaisons

Dim PLAGE\_RECHERCHE As Range

'Déclaration de variable pour marcheur compteur ligne

Dim LIGNE As Double

'Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

▶ ‘ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le chemin C:\Users\waard\Desktop\ le fichier stop.txt doit se trouver sur le bureau user

'----------------------------------------------------------------------------------------------------------------

Changer l’adresse ci-après : « C:\Users\waard\Desktop\STOP.txt », par la vôtre où doit se trouver le fichier STOP.txt chez vous (en théorie dans les cas courant, seul votre nom d’utilisateur « waard » doit être différent de cette adresse), bien veiller à conserver les caractères « … » sous peine d’erreur.

'4. Test arrêt si nb solution atteinte ou erreur ou minuteur terminé ou date planificateur atteinte

'4.1. Test d'arrêt macro à la colonne Combinaison suivant choix maximum quand récursivité repart du début Profondeur -1

'Contrôle qu'une fois bouclé sur choix maxi combinaisons trouvées, arrêter le test de récursivité

▶ ‘If Compteur >= CHOIX Or STOPERROR = True Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If Compteur >= CHOIX Or STOPERROR = True Then

'suivant choix utilisateur dans options si j = 1 colonne combi1, si j = 2 colonne combi2, si j = 3 colonne combi3, si j = 4 colonne hors tableur FIN macro"

▶ ‘If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then

▶ ‘OPTION\_ARRET\_PROG

▶ ‘Else

Exit For 'Sort de la boucle for pour sortir de la sub

▶ ‘End If

End If

'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'4.2. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

▶ ‘If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

▶ ‘If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit For

End If

End If

End If

'+++++++++

Next K 'change de permutation à K terme en allant au prochain terme K, jusqu'au terme maxi de K

'----------------------------------------------------------------------------------------------------------------

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

 Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'----------------------------------------------------------------------------------------------------------------

'0. DECLARATION VARIABLES GENERALES

'Déclaration de variable d'adresse plage des combinaisons

Dim PLAGE\_RECHERCHE As Range

'Déclaration de variable pour marcheur compteur ligne

Dim LIGNE As Double

'Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le chemin C:\Users\waard\Desktop\ le fichier stop.txt doit se trouver sur le bureau user

'----------------------------------------------------------------------------------------------------------------

'4. Test arrêt si nb solution atteinte ou erreur ou minuteur terminé ou date planificateur atteinte

'4.1. Test d'arrêt macro à la colonne Combinaison suivant choix maximum quand récursivité repart du début Profondeur -1

'Contrôle qu'une fois bouclé sur choix maxi combinaisons trouvées, arrêter le test de récursivité

If Compteur >= CHOIX Or STOPERROR = True Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If Compteur >= CHOIX Or STOPERROR = True Then

'suivant choix utilisateur dans options si j = 1 colonne combi1, si j = 2 colonne combi2, si j = 3 colonne combi3, si j = 4 colonne hors tableur FIN macro"

If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then

OPTION\_ARRET\_PROG

Else

Exit For 'Sort de la boucle for pour sortir de la sub

End If

End If

'++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

'4.2. Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

'If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit For

End If

End If

End If

'+++++++++

Next K 'change de permutation à K terme en allant au prochain terme K, jusqu'au terme maxi de K

'----------------------------------------------------------------------------------------------------------------

**Dans le fichier MYRMIDON** > Module Principal \_lettrage > RECURSIVITE\_PERMUTATION

 Code inactivé de base

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

▶ ‘If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

▶ ‘If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

▶ If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Function

End If

End If

End If

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

 Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'Test arrêt, Pour activation de l'arrêt de recherche si programmé, vérification par si pas = 0

'1 - Si arrêt date planifié non activé, alors arrêt sur temps TIMER d'arrêt, 2 -sinon si Timer d'arrêt ou Date d'arrêt planifié atteinte procédure d'arrêt

If Not worksheets(1).CTRLCALCAUTO.Value = True Then

'Option planification non activé, arrêt par controle TIMER

If TSTOPTIME > 0 Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

‘If TSTOPTIME > 0 Then

OPTION\_ARRET\_PROG

End If

'Option planification activé, arrêt par controle TIMER ou date planifié atteinte

Else

If TSTOPTIME > 0 Or Now > STOPDATEPLANI Or Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then 'pour activation zaz stopmacro

‘If TSTOPTIME > 0 Or Now > STOPDATEPLANI Then

OPTION\_ARRET\_PROG

If STOPINTERRUPTOR = True Then

Exit Function

End If

End If

End If

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sauvegarder le fichier et c’est OK pour celui-là.

## Etape 8 – Activation du code ZAZ dans PLANIFICATEUR.

Le caractère **‘** permet de désactiver le code, celui sera considéré par VBA, comme étant du texte.

* Les lignes de texte, ne doivent surtout pas être activées, sous peine d’erreurs.*

* Vous pouvez utiliser la recherche texte Ctrl + F en recherchant le mot clef : « ZAZ » et aller directement sur les lignes concernées, pour supprimer le caractère* **‘** *devant la ligne de code ZAZ et l’ajouter sur la ligne de code de base se trouve en générale en dessous de la ligne de code ZAZ.*

**Dans le fichier PLANIFICATEUR** > Module Principal\_planificateur > Ligne tout en haut **Public**

** Code inactivé de base

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'0. Variables Publiques communes à ce classeur

Public STOPDATEPLANI As Date, PLANIFICATEUR As Workbook, FICHIERCOMBI As String, NAMEFICHIERCOMBI As String, MACROFICHIERCOMBI As String, TSTOPCOMBI As String, MAILCOMBI As String, ONEDRIVECOMBI As String, SAVEHARDCOMBI As String ▶ ‘, ZAZ\_STOPMACRO As String

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

** **Code activé -> Supprimer le caractère ‘ et l’espace devant, ZAZ\_STOPMACRO As String**

'\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'0. Variables Publiques communes à ce classeur

Public STOPDATEPLANI As Date, PLANIFICATEUR As Workbook, FICHIERCOMBI As String, NAMEFICHIERCOMBI As String, MACROFICHIERCOMBI As String, TSTOPCOMBI As String, MAILCOMBI As String, ONEDRIVECOMBI As String, SAVEHARDCOMBI As String, ZAZ\_STOPMACRO As String

** Code inactivé de base

'2.5. Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

▶ ‘ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le fichier stop.txt doit se trouver sur le bureau user car l'appli. ZAZ est censée le créer à cet emplacement

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

Changer l’adresse ci-après : « C:\Users\waard\Desktop\STOP.txt », par la vôtre où doit se trouver le fichier STOP.txt chez vous (en théorie dans les cas courant, seul votre nom d’utilisateur « waard » doit être différent de cette adresse), bien veiller à conserver les caractères « … » sous peine d’erreur.

 Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'2.5. Définition variable ZAZ\_STOPMACRO du chemin+nom fichier STOP.txt

ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\STOP.txt" 'Le fichier stop.txt doit se trouver sur le bureau user car l'appli. ZAZ est censée le créer à cet emplacement

** Code inactivé de base

'+3.3.1. SI Fichier STOP de ZAZ Stopmacro, alors arrêt

▶ ‘If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then

▶ ‘ARIANNE\_ZAZ 'Message vocale d'info user d'arrêt de macro par ZAZ

▶ ‘Range("B" & NUMLIGNELISTFILE).Interior.ColorIndex = 46 '46 = orange

▶ ‘Range("B" & NUMLIGNELISTFILE).Value = "Traitement arrêté par utilisateur" 'Etat

▶ ‘Range("B" & NUMLIGNELISTFILE).Font.Bold = True 'En gras

▶ ‘Windows.Application.Visible = True 'Rendre visible l'application Excel

▶ ‘Kill ZAZ\_STOPMACRO 'Suppression du fichier STOP.txt de ZAZ à l'emplacement prévu

▶ ‘Exit For 'Sortie de boucle et lancement des process de fin de planification (save, mail,...)

▶ ‘End If

Supprimer le caractère **‘** devant ces lignes pour activer le code ZAZ : ▶

Mettre le caractère **‘** devant ces lignes pour désactiver le code de base : ▶

 Code activé que vous pouvez copier/coller, mais attention à bien copier/coller les bonnes parties au bon endroit.

'+3.3.1. SI Fichier STOP de ZAZ Stopmacro, alors arrêt

If Not Dir(ZAZ\_STOPMACRO, vbDirectory) = "" Then

‘ARIANNE\_ZAZ 'Message vocale d'info user d'arrêt de macro par ZAZ

Range("B" & NUMLIGNELISTFILE).Interior.ColorIndex = 46 '46 = orange

Range("B" & NUMLIGNELISTFILE).Value = "Traitement arrêté par utilisateur" 'Etat

Range("B" & NUMLIGNELISTFILE).Font.Bold = True 'En gras

Windows.Application.Visible = True 'Rendre visible l'application Excel

Kill ZAZ\_STOPMACRO 'Suppression du fichier STOP.txt de ZAZ à l'emplacement prévu

Exit For 'Sortie de boucle et lancement des process de fin de planification (save, mail,...)

End If

Sauvegarder le fichier et c’est OK pour celui-là.

## Etape 9 – Facultatif – Lancer automatiquement ZAZ Stopmacro à l’ouverture fichier.

Option facultative, à insérer dans l’événement OPEN du module THISWORKBOOK dans VBA, si vous souhaitez que ZAZ Stopmacro se lance automatiquement à l’ouverture d’un fichier MYRMIDON.

 Code activé, il n’y a qu’a collé le code tout en haut de la SUB

‘Private Sub Workbook\_Open()

'A chaque ouverture lancer l'application  
'0. Gestion erreur, si erreur passe, application non lancée ou déjà ouverte  
**On** **Error** **Resume** **Next**  
'1. Déclaration variable PATH\_LANCEMENT\_ZAZ\_STOPMACRO de chemin d'accès à application comme étant du texte  
**Dim** PATH\_LANCEMENT\_ZAZ\_STOPMACRO **As** **String**  
'2. Définition du chemin d'accès de l'application, casse majuscule pris en compte, sinon erreur  
PATH\_LANCEMENT\_ZAZ\_STOPMACRO = "C:\Users\waard\Desktop\ZAZ STOPMACRO\ZAZ STOPMACRO.exe"  
'3. Lancement de l'application, visible dans barre de notification  
Shell PATH\_LANCEMENT\_ZAZ\_STOPMACRO

‘…….. reste du code de la sub OPEN initiale à ne pas modifier.

Changer l’adresse ci-après : « C:\Users\waard\Desktop\ZAZ STOPMACRO\ZAZ STOPMACRO.exe », par la vôtre où doit se trouver le fichier application ZAZ chez vous, bien veiller à conserver les caractères « … » et la casse majuscule/minuscule sous peine d’erreur.

Voilà, votre application ZAZ StopMacro a été installée. 