

VBA – Classes CADDoc & CADSSet

Guide d'utilisation

Par Saboh

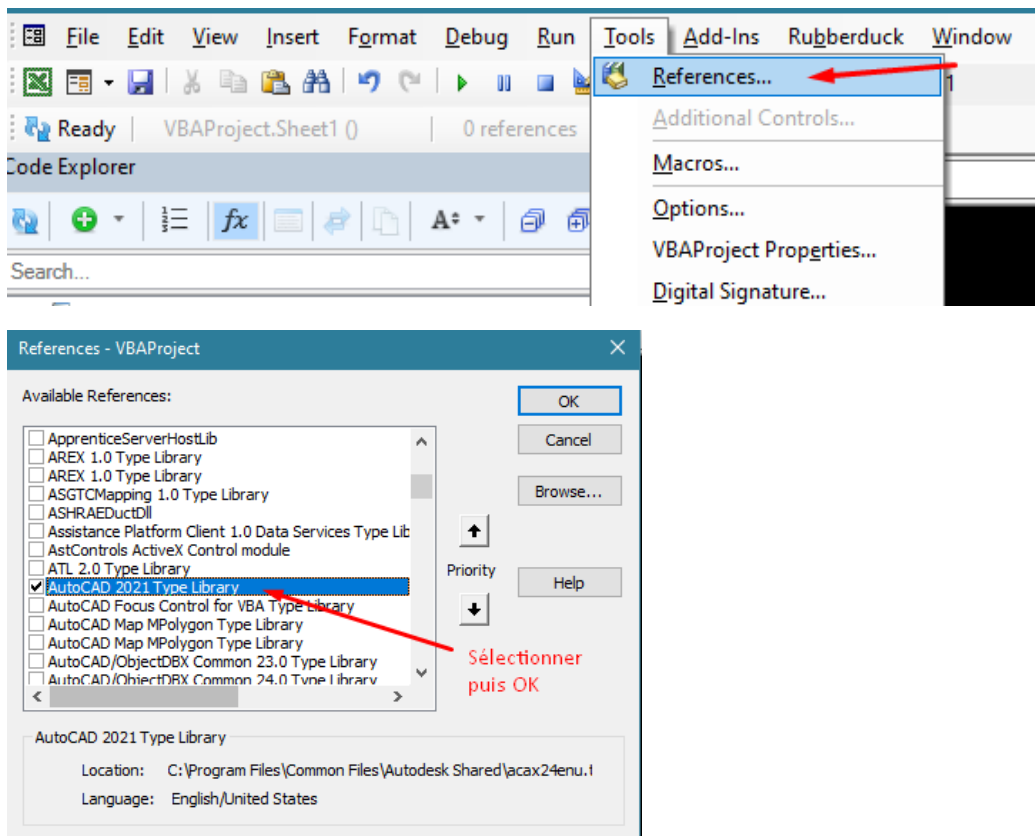
Contents

MISE EN PLACE	2
Référence à la AutoCAD Type Library	2
Réutilisation des classes	2
DESCRIPTION TECHNIQUE	3
Classe CADDoc	3
OpenFile(path)	3
SetTo(docName)	3
AcadDoc() → AcadDocument	3
AcadApp() → Acadapplication	4
AddSSet([setName]) → CADSSet	4
CreateLayer(layerName, [colorsRGB], [overwrite]) → AcadLayer	4
GetLayer(layerName) → AcadLayer	5
TableFromRange(rng, [hasHeaders], [tblPos]) → AcadTable	5
PromptYesNo(promptStr, [defaultAnswer]) → Boolean	5
PromptPickOne(objTypeName, [promptStr], [askConfirmation]) → AcadEntity	6
ZoomOn(entity,[Highlight])	7
Classe CADSSet	7
Define(setName, parentDoc)	7
SSet() → AcadSelectionSet	8
SelectAll()	8
SelectByType(typesList)	8
SelectByName(namesList)	9
SelectByColor(colorsList)	9
SelectByLayer(layersList)	10
SelectMatching([oLayers], [oTypes], [oNames], [oColors])	10
SelectOnScreen(promptStr, [allowedTypes], [allowedLayers], [allowedColors])	11

MISE EN PLACE

Référence à la AutoCAD Type Library

Tout d'abord, dans le classeur Excel contenant les macros qui vont interagir avec AutoCAD, il faut ajouter la référence à la "AutoCAD 20XX Type Library"



L'année peut changer (2019, 2021, 2022...) mais ce n'est pas très important.

Réutilisation des classes

Ensuite, dans les fichiers joints ci-dessous, deux possibilités : ou bien télécharger le fichier Excel d'exemple, et partir de celui-ci en y ajoutant vos modules/macros. *ATTENTION : si en lançant une macro de ce fichier vous avez une erreur : vérifiez l'étape précédente : il est très probable que vous ayez une version différente de la mienne, auquel cas il faut remplacer la référence à AutoCAD par la vôtre.*

Ou bien insérer dans votre classeur les 2 modules de classe CADDoc.cls et CADSSet.cls qui sont les bibliothèques de fonctions ajoutées. Cf. fichiers joints.

DESCRIPTION TECHNIQUE

Classe CADDoc

Module de classe. Désigné comme le "CADDoc" ci-dessous.

OpenFile(path)

Ouvre le fichier se trouvant au chemin indiqué dans AutoCAD. Si AutoCAD n'est pas déjà lancé, le lance. Gère les DWG, DXF, IGES et STEP.

Ne vérifie pas si le fichier est déjà ouvert : si c'est le cas, ouvre une copie en lecture seule.

```
Sub Example()  
    ' Creation d'une variable CADDoc qui va ouvrir le document indiqué  
    Dim myDrawing As New CADDoc  
    myDrawing.OpenFile "C:\Users\Patrick\Downloads\Drawing1.dwg"  
End Sub
```

SetTo(docName)

Assigne le CADDoc au document de titre désigné (attention, bien indiquer l'extension dans son nom). Si le fichier n'est pas déjà ouvert ou n'a jamais été enregistré, demande si vous voulez en créer un nouveau.

```
Sub Example()  
    ' Creation d'une variable CADDoc faisant le lien  
    ' avec le document "Drawing1.dwg" déjà ouvert  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
End Sub
```

AcadDoc() → AcadDocument

Renvoie le document AutoCAD contenu dans le CADDoc. Permet d'accéder à l'ensemble des fonctions de [Document Object \(ActiveX\) | Autodesk](#).

```
Sub Example()  
    ' Utilisation de la propriété AcadDoc pour accéder à toutes  
    ' les propriétés de la classe AcadDocument, comme le ModelSpace  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim myMS As AcadModelSpace  
    Set myMS = myDrawing.AcadDoc.ModelSpace  
End Sub
```

AcadApp() → AcadApplication

Renvoie l'application AutoCAD liée au CADDoc. Permet d'accéder à l'ensemble des fonctions de [Application Object \(ActiveX\) | Autodesk](#).

```
Sub Example()  
    ' Utilisation de la propriété AcadDoc pour accéder a toutes  
    ' les propriétés de la classe AcadApplication, comme le ZoomAll  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim myMS As AcadModelSpace  
    Set myMS = myDrawing.AcadApp.ZoomAll  
End Sub
```

AddSset([setName]) → CADSSet

Crée un CADSSet dans le CADDoc avec le nom renseigné (optionel). Si le nom est déjà pris, renomme le set en [setName] & i avec i = nombre de sets dans le document.

Renvoie le CADSSet créé (vide).

```
Sub Example()  
    ' Création d'une variable CADSSet (vide) dans le document  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim mySset As CADSSet  
    Set mySset = myDrawing.AddSset("premier set")  
End Sub
```

CreateLayer(layerName, [colorsRGB], [overwrite]) → AcadLayer

Crée un calque avec le nom donné et la couleur RGB donnée (colorsRGB = tableau contenant les valeurs RGB. Par exemple colorsRGB = Array(255,255,255)). Si couleur non indiquée : couleur aléatoire.

Si le nom existe déjà, renvoie le calque existant. Si overwrite est défini sur True (par défaut False), change la couleur du calque existant.

```
Sub Example()  
    ' Création d'un nouveau calque, de couleur grise  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim myLayer As AcadLayer  
    Set myLayer = myDrawing.CreateLayer("fondations", Array(90, 90, 90))  
End Sub
```

GetLayer(layerName) → AcadLayer

Renvoie le calque de nom fourni s'il existe. Sinon renvoie Nothing.

```
Sub Example()  
    ' Récupération du calque "toiture" si il existe  
    ' et sinon, création d'un nouveau calque toiture de couleur aléatoire  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim myLayer As AcadLayer  
    Set myLayer = myDrawing.GetLayer("toiture")  
    If myLayer Is Nothing Then  
        Set myLayer = myDrawing.CreateLayer("toiture")  
    End If  
End Sub
```

TableFromRange(rng, [hasHeaders], [tblPos]) → AcadTable

Crée une copie de la plage Excel (Range.Value) dans AutoCAD. Pour utiliser avec les tableaux structurés (listObjects), utiliser simplement ListObject.Range pour récupérer la plage correspondante.

Si hasHeaders = True (défaut False), la 1e ligne du tableau garde le style "En-tête" des tableaux AutoCAD.

tblPos : position (x,y,z) du tableau dans le dessin. Par défaut (0,0,0). Bien fournir une Array de 3 valeurs, même si vous travaillez en 2D (vue plan) il faut indiquer z = 0.

```
Sub Example()  
    ' Copie des cellules "A1:B5" de la feuille active  
    ' dans une table dans le dessin, aux coordonnées (100,100,0)  
    Dim myDrawing As New CADDoc  
    myDrawing.SetTo "Drawing1.dwg"  
  
    Dim myTable As AcadTable  
    Set myTable = myDrawing.TableFromRange( _  
        ActiveSheet.Range("A1:B5"), , Array(100, 100, 0))  
End Sub
```

PromptYesNo(promptStr, [defaultAnswer]) → Boolean

Demande à l'utilisateur **DANS AutoCAD**, de choisir entre Yes et No.

PromptStr = la question.

DefaultAnswer = la réponse par défaut si appui sur Entrée sans choisir une option (par défaut : No).

Renvoie True/False selon le choix de l'utilisateur.

```

Sub Example()
    ' Demande a l'utilisateur s'il veut créer une ligne
    Dim myDrawing As New CADDoc
    myDrawing.SetTo "Drawing1.dwg"

    Dim ptIni(0 To 2) As Double
    Dim ptFin(0 To 2) As Double
    ptIni(0) = 0#: ptIni(1) = 0#: ptIni(2) = 0#
    ptFin(0) = 10#: ptFin(1) = 0#: ptFin(2) = 0#

    Dim reponse As Boolean
    reponse = myDrawing.PromptYesNo("Voulez-vous créer une ligne ?")
    If reponse Then
        myDrawing.AcadDoc.ModelSpace.AddLine ptIni, ptFin
    End If
End Sub

```

PromptPickOne(objTypeName, [promptStr], [askConfirmation]) → AcadEntity

Demande à l'utilisateur **DANS AutoCAD**, de cliquer sur un objet. Renvoie cet objet.

objTypeName = le TypeName de l'objet cf. [TypeName function \(Visual Basic for Applications\) | Microsoft Learn](#) (pour les objets AutoCAD, faites un test pour avoir le nom exact. Par exemple les polylines "plates" sont *IAcadLWPolyline*. Souvent *"IAcad"* & nom courant).

PromptStr = la question posée (par défaut "Select an an object on the drawing")

askConfirmation = True/False (défaut False) : met en surbrillance l'objet et demande à l'utilisateur si son choix est correct (utile quand les objets sont proches). Si il choisit non, redemande...

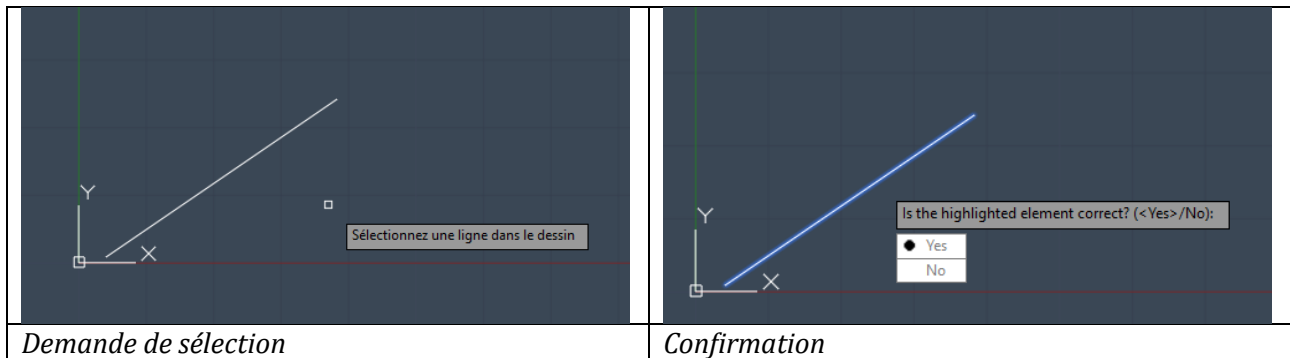
Renvoie l'objet sélectionné.

```

Sub Example()
    ' Demande a l'utilisateur de sélectionner une ligne dans le dessin
    ' et lui demande de confirmer sa sélection
    Dim myDrawing As New CADDoc
    myDrawing.SetTo "Drawing1.dwg"

    Dim maLigne As AcadLine
    Set maLigne = myDrawing.PromptPickOne( _
        "IAcadLine", "Sélectionnez une ligne dans le dessin", True)
End Sub

```



ZoomOn(entity, [Highlight])

Zoom sur un objet du dessin.

Entity : l'objet.

Highlight : True/False (défaut False). Si défini sur True, met l'objet en surbrillance.

```
Sub Example()
  ' Zoom sur le premier objet du dessin (indice 0)
  ' et le met en surbrillance
  Dim myDrawing As New CADDoc
  myDrawing.SetTo "Drawing1.dwg"

  Dim premierObj As AcadEntity
  Set premierObj = myDrawing.AcadDoc.ModelSpace.Item(0)
  myDrawing.ZoomOn premierObj, True
End Sub
```

Classe CADSSet

Module de classe. Désigné comme le "set" ci-dessous.

Il est recommandé d'utiliser la méthode AddSSet de la classe CADDoc pour créer un CADSSet. Les SelectionSets sont liés à un document, mais n'ont, par défaut, pas de propriété .Document. Donc lorsque l'on travaille avec plusieurs documents, on peut donc vite se perdre.

Define(setName, parentDoc)

Si le set est créé via "Set monSet = New CADSSet", il est nécessaire d'appeler cette fonction. Si on utilise CADDoc.AddSSet cet appel est fait automatiquement.

Instancie le set et initialise ses variables.

setName = nom du set. Si existe déjà : le set (nouveau) est renommé en setName & i (i = nombre de sets dans le document).

parentDoc = le document contenant le set. AcadDocument.

```
' La fonction AddSSet garantie l'instanciation du SelectionSet
' et de son document associé
Public Function AddSSet( _
    Optional ByVal setName As String = vbNullString) _
    As CADSSet
    Dim newSet As CADSSet: Set newSet = New CADSSet
    newSet.Define setName, this.doc
    Set AddSSet = newSet
End Function
```

SSet() → AcadSelectionSet

Renvoie le SelectionSet AutoCAD contenu dans le CADSSet. Permet d'accéder à l'ensemble des fonctions de [SelectionSet Object \(ActiveX\) | Autodesk](#).

```
Sub Example()
    ' Utilisation de la propriété SSet pour accéder a toutes
    ' les propriétés de la classe AcadSelectionSet
    ' comme Clear pour le vider

    Dim myDrawing As New CADDoc
    myDrawing.SetTo "Drawing1.dwg"

    Dim monSet As CADSSet
    Set monSet = myDrawing.AddSSet("monSet")
    monSet.SSet.Clear
End Sub
```

SelectAll()

Insère dans le set tous les objets du dessin.

```
Sub Example()
    ' Mise de tous les éléments du dessin dans le set

    Dim myDrawing As New CADDoc
    myDrawing.SetTo "Drawing1.dwg"

    Dim monSet As CADSSet
    Set monSet = myDrawing.AddSSet("monSet")
    monSet.SelectAll
End Sub
```

SelectByType(typesList)

Insère dans le set tous les objets du dessin de types listés.

typesList = String contenant les types d'objets séparés par des virgules. ATTENTION : types ≠ TypeName. Il s'agit du nom en haut de la page d'aide. Par exemple, pour les [Region object \(ActiveX\) | Autodesk](#), il s'agit de "Region".

```
Sub Example()  
  ' Mise de tous les éléments Polylines (2D) et Lines du dessin  
  ' dans mon set  
  
  Dim myDrawing As New CADDoc  
  myDrawing.SetTo "Drawing1.dwg"  
  
  Dim monSet As CADSSet  
  Set monSet = myDrawing.AddSSet("monSet")  
  monSet.SelectByType "LWPolyline,Line"  
End Sub
```

SelectByName(namesList)

Insère dans le set tous les objets du dessin de noms listés.

namesList = String contenant les noms des objets, séparés par des virgules.

```
Sub Example()  
  ' Mise de tous les éléments de nom "blockMur" dans mon set  
  ' typiquement, les blockRefs (copie de blocs) partagent le meme nom  
  
  Dim myDrawing As New CADDoc  
  myDrawing.SetTo "Drawing1.dwg"  
  
  Dim monSet As CADSSet  
  Set monSet = myDrawing.AddSSet("monSet")  
  monSet.SelectByName "blockMur"  
End Sub
```

SelectByColor(colorsList)

Insère dans le set tous les objets du dessin de couleurs listées.

colorsList = String contenant les couleurs des objets, séparées par des virgules. Vous pouvez directement utiliser "BYBOCK" et "BYLAYER" pour référer à ces modes de coloriage.

```

Sub Example()
  ' Mise de tous les éléments de couleur ByLayer (du calque) et 0
  ' dans mon set

  Dim myDrawing As New CADDoc
  myDrawing.SetTo "Drawing1.dwg"

  Dim monSet As CADSSet
  Set monSet = myDrawing.AddSSet("monSet")
  monSet.SelectByColor "BYLAYER,0"
End Sub

```

SelectByLayer(layersList)

Insère dans le set tous les objets du dessin dans les calques listés.

layersList = String contenant les noms des calques, séparés par des virgules.

```

Sub Example()
  ' Mise de tous les éléments des calques fondations et RDC
  ' dans mon set

  Dim myDrawing As New CADDoc
  myDrawing.SetTo "Drawing1.dwg"

  Dim monSet As CADSSet
  Set monSet = myDrawing.AddSSet("monSet")
  monSet.SelectByLayer "fondations,RDC"
End Sub

```

SelectMatching([oLayers], [oTypes], [oNames], [oColors])

Permet des combinaisons complexes des différentes fonctions susmentionnées. Chaque argument, optionnel, est équivalent à son homologue.

Note importante : les conditions au sein d'un même "groupe" sont complémentaires ("OU" : par exemple, type = Region OU LWPolyline). Par contre elles sont additionnelles entre les groupes (ET : par exemple, type = Region ET layer = 0).

Cela suit la logique "je sélectionne les objets de [ce type] dans [ces calques] ayant [cette couleur]".

```

Sub Example()
  ' Mise de toutes les lignes et polylines des calques fondations et RDC
  ' de couleur ByLayer dans mon set

  Dim myDrawing As New CADDoc
  myDrawing.SetTo "Drawing1.dwg"

  Dim monSet As CADSSet
  Set monSet = myDrawing.AddSSet("monSet")
  monSet.SelectMatching _
    oLayers:="fondations,RDC", _
    oTypes:="LWPolyline,Line", _
    oColors:="BYLAYER"
End Sub

```

SelectOnScreen(promptStr, [allowedTypes], [allowedLayers], [allowedColors])

Demande à l'utilisateur de sélectionner un/des objets (appui sur Entrée pour finir/confirmer) en respectant les filtres définis.

promptStr = le message montré à l'utilisateur.

allowedTypes = les types d'objets permis, séparés par une virgule (identique au cas précédents)

allowedLayers = les calques permis (idem)

allowedColors = les couleurs permises (idem)

Ainsi, si l'utilisateur clique sur un objet non autorisé (par exemple sur le mauvais calque), l'objet ne sera pas sélectionné. Utile pour limiter les erreurs.

```

Sub Example()
  ' Demande a l'utilisateur de selectionner des lines/polylines
  ' dans le dessin, puis suppression de ces dernieres

  Dim myDrawing As New CADDoc
  myDrawing.SetTo "Drawing1.dwg"

  Dim monSet As CADSSet
  Set monSet = myDrawing.AddSSet("monSet")
  monSet.SelectOnScreen _
    "Selectionnez les lignes a supprimer", _
    "Line,Polyline"
  monSet.SSet.Erase
End Sub

```