**Catia百格线生成宏**

 你是否为在Catia做2D图纸的时候，画百格线而伤脑筋呢？别担心，接下来我告诉你一个简单的方法

1.新建一个txt文本文档，比如Draw-Grid.txt  
2.把以下内容复制到Draw-Grid.txt中  
＝＝＝＝＝＝＝＝＝这里开始，不要复制我＝＝＝＝＝＝＝＝

Option Explicit  
  
  
' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
'   Purpose:      This macro allows you to create Grid line in CATIA drawing  
'   Author: chenqa  
'   Languages:   VBScript  
'   Locales:       English   
'   Developing CATIA Level: V5R12  
'   View mush parallel to system aixes,view angle 0deg,90deg and -90deg  
' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
Sub CATMain()  
CATIA.RefreshDisplay = False  
Dim sStatus As String   
    ' Set the CATIA popup file alerts to False  
    ' It prevents to stop the macro at each alert during its execution  
    CATIA.DisplayFileAlerts = False  
    ' Optional: allows to find the sample wherever it's installed  
    ' Variables declaration  
    Dim oDrwDocument As DrawingDocument  
    Dim oDrwSheets As DrawingSheets  
    Dim oDrwSheet As DrawingSheet  
    Dim oDrwView As DrawingView  
    Dim oFactory2D AS Factory2D  
   ' The Distance between the lines  
    Dim D As Integer  
    Dim nx As Integer  
    Dim ny As Integer  
   ' The point coordinate select from Drawing  
    Dim X1 As Integer  
    Dim Y1 As Integer  
    Dim X2 As Integer  
    Dim Y2 As Integer  
    Dim Pt1 As Point2D  
    Dim Pt2 As Point2D  
   'The view scale dAngle for rotate view scale for view scale  
    Dim dScale,dAngle As Double  
    'The view coordinate origin  
    Dim X As Int  
     Dim Y As Int  
    Dim xSel   As INFITF.Selection  
    D= InputBox("Please Input the Distance Value", "input box", "100")  
    D= Cint (D)  
    ' Retrive a new drawing document  
    Set oDrwDocument = CATIA.ActiveDocument  
' Retrieve the drawing document's sheets collection  
    Set oDrwSheets = oDrwDocument.Sheets  
' Retrieve the active sheet  
    Set oDrwSheet = oDrwSheets.ActiveSheet  
    ' Retrieve the active view of the sheet  
    Set oDrwView = oDrwSheet.Views.ActiveView  
    'Retrive the value of the view     
     X= oDrwView.xAxisData  
     Y= oDrwView.yAxisData  
     dScale = oDrwView.Scale   
     dAngle= oDrwView.Angle  
     Set oFactory2D = oDrwView.Factory2D  
   'Get the coordinate from the select two point  
    'On Error Resume Next   
     Set xSel = CATIA.ActiveDocument.Selection  
     xSel.clear  
     ReDim sFilter(0)   
     sFilter(0) = "Point2D"   
     MsgBox "Please select the left-bottom point "   
     sStatus = xSel.SelectElement2(sFilter, "Select First Point.", false)  
     If (sStatus = "Normal") Then   
      Dim SelectedPoint1 As SelectedElement  
      Set SelectedPoint1 = xSel.Item(1)   
      Dim pt1Coord(2) As Int   
      SelectedPoint1.GetCoordinates ( pt1Coord )  
      'MsgBox "The frst point has been selected "   
      Else MsgBox "Select a 2D Point 1"  
      Exit Sub  
      End If   
      MsgBox "Please select the ritht-top point "   
      sStatus = xSel.SelectElement2(sFilter, "Select The Second Point.", false)  
      If (sStatus = "Normal") Then   
       Dim SelectedPoint2 As SelectedElement  
       Set SelectedPoint2 = xSel.Item(1)   
      Dim pt2Coord(2) As Int  
    SelectedPoint2.GetCoordinates ( pt2Coord )  
     'MsgBox "The second point has been selected "   
     Else MsgBox "Select a 2D point 1"  
     Exit Sub  
     End If   
if dAngle=0 then   
   X1= Cint( (pt1Coord(0) - X)/dScale)   
   Y1= Cint ((pt1Coord(1) - Y)/dScale)   
   X2= Cint ((pt2Coord(0) - X)/dScale)   
   Y2= Cint ((pt2Coord(1) - Y)/dScale)  
end if  
   'MsgBox (pt1Coord(0))  
   'MsgBox X  
   if dAngle>0 then  
    X1=Cint((pt1Coord(1)-Y)/dScale)  
       Y1=Cint ((pt1Coord(0) -X)/dScale)   
       X2= Cint ((pt2Coord(1)-Y)/dScale)   
       Y2= Cint ((pt2Coord(0)-X)/dScale)  
end if  
if dAngle<0 then  
    X1=Cint((pt1Coord(1)-Y)/dScale)  
       Y1=Cint ((pt1Coord(0) -X)/dScale)  
       X2= Cint ((pt2Coord(1)-Y)/dScale)  
       Y2= Cint ((pt2Coord(0)-X)/dScale)  
end if  
    X1 = D \* Cint(X1/D)  
    Y1 = D \* Cint (Y1/D)  
    X2= D \* Cint (X2/D)  
    Y2 = D \* Cint (Y2/D)   
      
    nx = (X2-X1) \ D 'The number of the horizontal line  
    ny = (Y2-Y1) \ D 'The number of the vertical line  
      
    Dim Line2D1 As Line2D  
    Dim Circle2D1 as Circle2D  
   Dim MyText as DrawingText  
   Dim iFontsize as Double  
    Dim i As Int  
    Dim j As Int  
    Dim R As Doubel 'the radius of the circle  
    iFontSize = 3.5  
    R=8  
    R=R / dScale  
'------------------------------------------------------  
Dim Di\_H,Di\_V as int  
Dim Text\_XYZ\_H as string  
Dim Text\_XYZ\_V as string  
Di\_H=1  
Di\_V=1  
'Compare the drawing view HV with 3D Aixes   
Dim XX1,YY1,ZZ1,XX2,YY2,ZZ2 as int  
oDrwView.GenerativeBehavior.GetProjectionPlane XX1,YY1,ZZ1,XX2,YY2,ZZ2  
if (XX1=1) then   
    Text\_XYZ\_H="X"  
End if  
if (XX1=-1) then   
    Text\_XYZ\_H="X"  
    Di\_H=-1  
End if  
if (YY1=1) then   
    Text\_XYZ\_H="Y"  
End if  
if (YY1=-1) then   
    Text\_XYZ\_H="Y"  
    Di\_H=-1  
End if  
if (ZZ1=1) then   
    Text\_XYZ\_H="Z"  
End if  
if (ZZ1=-1) then   
    Text\_XYZ\_H="Z"  
    Di\_H=-1  
End if  
if (XX2=1) then   
    Text\_XYZ\_V="X"  
End if  
if (XX2=-1) then   
    Text\_XYZ\_V="X"  
    Di\_V=-1  
End if  
if (YY2=1) then   
    Text\_XYZ\_V="Y"  
End if  
if (YY2=-1) then   
    Text\_XYZ\_V="Y"  
    Di\_V=-1  
End if  
if (ZZ2=1) then   
    Text\_XYZ\_V="Z"  
End if  
if (ZZ2=-1) then   
    Text\_XYZ\_V="Z"  
    Di\_V=-1  
End if  
if dAngle>0 then  
    Di\_V=-Di\_V  
end if  
if dAngle<0 then  
    Di\_H=-Di\_H  
end if  
Dim oSel as Selection  
Dim oVisProps as VisPropertySet  
set oSel = oDrwDocument.Selection  
oSel.Clear  
Dim TextV As int  
TextV=R/2  
  
'Draw the horizontall line  
    for i=0 TO ny  
      if dAngle=0 then  
          set Line2D1 = oFactory2D.CreateLine (X1-D/3,Y1+D\*i,X1+nx\*D+D/3,Y1+D\*i)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(X1-D/3 -R,Y1+D\*i,R)  
          oSel.Add Circle2D1  
          set Line2D1= oFactory2D.CreateLine(X1-D/3-R\*2,Y1+D\*i,X1-D/3,Y1+D\*i)  
          oSel.Add Line2D1  
          Set MyText = oDrwView.Texts.Add(Text\_XYZ\_V,X1-D/3 -R,Y1+D\*i+TextV)  
         MyText.AnchorPosition = catMiddleCenter   
         MyText.SetFontSize 0, 0, iFontSize  
          Set MyText = oDrwView.Texts.Add((Y1+D\*i)\*Di\_V,X1-D/3 -R,Y1+D\*i-TextV)  
           MyText.AnchorPosition = catMiddleCenter   
          MyText.SetFontSize 0, 0, iFontSize  
       end if  
      if dAngle>0 then  
          set Line2D1 = oFactory2D.CreateLine (X1-D/3,-Y1-D\*i,X1+nx\*D+D/3,-Y1-D\*i)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(X1+nx\*D+D/3 +R,-Y1-D\*i,R)  
          oSel.Add Circle2D1  
          set Line2D1= oFactory2D.CreateLine(X1+nx\*D+D/3+R,-Y1-D\*i+R,X1+nx\*D+D/3+R,-Y1-D\*i-R)  
          oSel.Add Line2D1  
          Set MyText = oDrwView.Texts.Add(Text\_XYZ\_V,X1+nx\*D+D/3+R +TextV,-Y1-D\*i)  
          MyText.AnchorPosition = catMiddleCenter   
          MyText.SetFontSize 0, 0, iFontSize  
          Set MyText = oDrwView.Texts.Add((Y1+D\*i)\*Di\_V,X1+nx\*D+D/3+R -TextV,-Y1-D\*i)  
           MyText.AnchorPosition = catMiddleCenter   
          MyText.SetFontSize 0, 0, iFontSize  
       end if  
      if dAngle<0 then  
          set Line2D1 = oFactory2D.CreateLine (-X1+D/3,Y1+D\*i,-(X1+nx\*D+D/3),Y1+D\*i)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(-(X1+nx\*D+D/3)-R,Y1+D\*i,R)  
          oSel.Add Circle2D1  
          set Line2D1= oFactory2D.CreateLine(-X1-nx\*D-D/3-R,Y1+D\*i+R,-X1-nx\*D-D/3-R,Y1+D\*i-R)  
          oSel.Add Line2D1  
          Set MyText = oDrwView.Texts.Add(Text\_XYZ\_V,-X1-nx\*D-D/3-R +TextV,Y1+D\*i)  
          MyText.AnchorPosition = catMiddleCenter   
          MyText.SetFontSize 0, 0, iFontSize  
          Set MyText = oDrwView.Texts.Add((Y1+D\*i)\*Di\_V,-X1-nx\*D-D/3-R -TextV,Y1+D\*i)  
           MyText.AnchorPosition = catMiddleCenter   
         MyText.SetFontSize 0, 0, iFontSize  
       end if  
    next  
  
'Draw the vertical line  
    for j=0 TO nx  
      if dAngle=0 then  
          set Line2D1 = oFactory2D.CreateLine (X1+D\*j,Y1-D/3,X1+D\*j,Y1+ny\*D+D/3)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(X1+D\*j,Y1+ny\*D+D/3+R,R)  
          oSel.Add Circle2D1  
          set Line2D1 = oFactory2D.CreateLine(X1+D\*j-R,y1+ny\*D+D/3+R,X1+D\*J+R,y1+ny\*D+D/3+R)  
          oSel.Add Line2D1  
           Set MyText = oDrwView.Texts.Add(Text\_XYZ\_H,X1+D\*j,Y1+ny\*D+D/3+R+TextV)  
                 MyText.AnchorPosition = catMiddleCenter  
                 MyText.SetFontSize 0, 0, iFontSize  
           Set MyText = oDrwView.Texts.Add((X1+D\*j)\*Di\_H,X1+D\*j,y1+ny\*D+D/3+R-TextV)  
                 MyText.AnchorPosition = catMiddleCenter  
                MyText.SetFontSize 0, 0, iFontSize  
      end if  
      if dAngle>0 then  
          set Line2D1 = oFactory2D.CreateLine (X1+D\*j,-Y1+D/3,X1+D\*j,-Y1-ny\*D-D/3)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(X1+D\*j,-Y1+D/3+R,R)  
          oSel.Add Circle2D1  
          set Line2D1 = oFactory2D.CreateLine(X1+D\*j,-Y1+D/3+R\*2,X1+D\*J,-Y1+D/3)  
          oSel.Add Line2D1  
           Set MyText = oDrwView.Texts.Add(Text\_XYZ\_H,X1+D\*j+TextV,-Y1+D/3+R)  
                 MyText.AnchorPosition = catMiddleCenter  
                MyText.SetFontSize 0, 0, iFontSize  
           Set MyText = oDrwView.Texts.Add((X1+D\*j)\*Di\_H,X1+D\*j-TextV,-Y1+D/3+R)  
                 MyText.AnchorPosition = catMiddleCenter  
                MyText.SetFontSize 0, 0, iFontSize  
      end if  
      if dAngle<0 then  
          set Line2D1 = oFactory2D.CreateLine (-X1-D\*j,Y1-D/3,-X1-D\*j,Y1+ny\*D+D/3)  
          oSel.Add Line2D1  
          set Circle2D1=oFactory2D.CreateClosedCircle(-X1-D\*j,Y1-D/3-R,R)  
          oSel.Add Circle2D1  
          set Line2D1 = oFactory2D.CreateLine(-X1-D\*j,Y1-D/3-R\*2,-X1-D\*J,Y1-D/3)  
          oSel.Add Line2D1  
           Set MyText = oDrwView.Texts.Add(Text\_XYZ\_H,-X1-D\*j+TextV,Y1-D/3-R)  
                 MyText.AnchorPosition = catMiddleCenter  
                MyText.SetFontSize 0, 0, iFontSize  
           Set MyText = oDrwView.Texts.Add((X1+D\*j)\*Di\_H,-X1-D\*j-TextV,Y1-D/3-R)  
                 MyText.AnchorPosition = catMiddleCenter  
                MyText.SetFontSize 0, 0, iFontSize  
      end if  
   next  
dim oFontSize As long  
' MyText.SetFontSize 0, 0, iFontSize  
    set oVisProps = oSel.VisProperties  
    oVisProps.SetRealWidth 1,0 '1st parameter line width 1-63 2nd parameter inheritance flag 1 or 0  
    oVisProps.SetRealColor 0,255,0,1   
    Set oVisProps = Nothing  
    Set oSel = Nothing  
    ' Update drawing table modifications  
    CATIA.ActiveWindow.ActiveViewer.Reframe   
End Sub

＝＝＝＝＝＝＝＝＝这里结束，不要复制我＝＝＝＝＝＝＝＝  
3.保存Draw-Grid.txt，然后将文件后缀改为CATScript即Draw-Grid.CATScript，就可以使用了  
  
使用方法：  
1.在Catia工程制图界面激活要创建百格线的视图  
2.在改视图里边创建两个点，这两个点分别是百格线的左下角和右上角点  
3.按ALT+F8打开宏库，导入刚才创建的宏   
   a.按ALT+F8出现一个界面，然后点宏库  
   b.库类型中选择“目录”—>“添加现有库”—>“浏览”，选取刚才创建的Draw-Grid.CATScript，确定后关闭  
4.点运行—>确定—>选左下角点—>确定—>选右上角点  
  
百格线至此就全部完成了