Python for the AWR VB Developer

This page is to describe differences in coding if you are used to the AWR VB scripting language.

Case Sensitivity

When developing in VB where the type library is understood, the case for items is automatically created. You don't have to be aware of the case when typing since it fixes itself.

We will see if when we can get the type library in python if this is fixed. When typing commands, you do need to be very aware of the case of each word.

Calling Methods

When you call methods in VB, you do not need to use parathesis after the method.

If VB, you could do either below

```
Project.Simulator.Analyze()
Project.Simulator.Analyze
```

In python, you must use the () at the end of the method or it won't run.

Creating Subroutines and Functions

in VB, the subroutine or function can be at any location in the file.

In Python, the subroutine or function must be defined in the file before it is used.

Quotes

Python allows the use of single or double-quotes for strings. This makes it easy to put single or double-quotes in a string by using the other type of quote to define the string.

Pathnames

If you are not specifying drive letters, python allows you to use a forward-slash in pathnames even on windows. If you want to use back-slashes then you must double them up.

```
pathname = '/windows/system32/'  # or
pathname = '\\windows\\system32'
```

When building pathnames up, it is much better to use os.path.join than just concatenating strings as it will handle the delimiters correctly.

Calling Enumerated Value

in VB that has the type library defined, you can specific enumerated values by name such as, the **mwGT_Rectangular** below**.**

```
Project.Graphs.Add("s21 and s11",mwGT_Rectangular)
```

When using python without the type library (you should never do this, see here), you need to replace these with integers, so python will be

```
 awr.Project.Graphs.Add("s21 and s11",3)

 but with the type library would be

 awr.Project.Graph.Add("s21 and s11", awrc.mwGT_Rectangular)
```

I found the easiest way to do this was to have the code in VB, use F1 help on the top object and then browse to the page that lists the enumerated types.

Examples

The code below is doing the AWR getting started guide for the liner filter by code. One is done by VB, one is done by Python

VB

```
' Code Module
Sub Main
        Dim sch As Schematic

        CleanUp

        Set sch = CreateSchematic

        BuildSchematic(sch)

        SetValues(sch)

        SetFrequencies

        AddGraph(sch)

        Simulate

        AddMarkers(sch)

        AddEquations(sch)

        OptLimits(sch)

        SetOptGoals(sch)

        ArrangeWindows

        Optimize
End Sub
Private Sub Optimize
        Project.Optimizer.MaxIterations = 500
        Project.Optimizer.Type = find_opt_type("Random (Local)")
        Project.Optimizer.NewWindow
        Project.Optimizer.Start
        While Project.Optimizer.Running=True
                Wait(0.5)
        Wend
End Sub

Function find_opt_type(nm As String) As Integer
        Dim typ As Integer

        typ = -1
        For i = 1 To Project.Optimizer.TypeCount
                If nm = Project.Optimizer.TypeName(i) Then
                        typ = i
                End If
        Next i
        If typ = -1 Then
                MsgBox ("could not find optimizer name specified:" & nm)
        End If
        find_opt_type = typ
End Function
Sub SetOptGoals(s As Schematic)
        Project.OptGoals.Add(s.Name,"DB(|S(1,1)|)",mwOGT_LessThan,1,2,0,500e6,mwUT_Frequency,-17,-17,mwUT_DB)
        Project.OptGoals.Add(s.Name,"DB(|S(2,1)|)",mwOGT_GreaterThan,1,2,0,500e6,mwUT_Frequency,-1,-1,mwUT_DB)
        Project.OptGoals.Add(s.Name,"DB(|S(2,1)|)",mwOGT_LessThan,1,2,700e6,1000e6,mwUT_Frequency,-30,-30,
mwUT_DB)
End Sub
```

```
Sub OptLimits(s As Schematic)
        'setup various values for optimization including constraints.
        s.Equations("CAP").Optimize=True
        s.Equations("CAP").Constrain=True
        s.Equations("CAP").LowerConstraint = 8*0.75
        s.Equations("CAP").UpperConstraint = 8*1.25
        s.Equations("IND").Optimize=True
        s.Equations("IND").Constrain=True
        s.Equations("IND").LowerConstraint = 15*0.75
        s.Equations("IND").UpperConstraint = 15*1.25
        s.Elements("CAP.C2").Parameters("C").Optimize=True
        s.Elements("CAP.C2").Parameters("C").Constrain=True
        s.Elements("CAP.C2").Parameters("C").LowerConstraint = 10*0.75*1e-12
        s.Elements("CAP.C2").Parameters("C").UpperConstraint = 10*1.25*1e-12
End Sub
Sub AddEquations(s As Schematic)
        'add equations and assign parameters to use the equation.
        s.Equations.Add("IND=15",1500,-1000)
        s.Equations.Add("CAP=8",2500,-1000)
        s.Elements("IND.L1").Parameters("L").ValueAsString = "IND"
        s.Elements("IND.L4").Parameters("L").ValueAsString = "IND"
        s.Elements("CAP.C1").Parameters("C").ValueAsString = "CAP"
        s.Elements("CAP.C3").Parameters("C").ValueAsString = "CAP"
End Sub
Sub AddMarkers(s As Schematic)
        Dim m As Marker
        Dim m2 As Marker

        Set m = Project.Graphs(1).Markers.Add(s.Name & ":DB(|S(2,1)|)",1,0.1e9)
        m.Type = mwMT_AutoSearch
        m.AutoSearch.mode=mwMAM_Max
        Set m2 = Project.Graphs(1).Markers.Add(s.Name & ":DB(|S(2,1)|)",1,0.5e9)
        m2.Type =mwMT_Offset
        m2.Offset.Distance = -3
        m2.Offset.mode=mwMOM_Y
        m2.Offset.ReferenceMarker="m1"

End Sub
Sub Simulate
        Project.Simulator.Analyze
End Sub
Sub AddGraph(s As Schematic)
        Dim g As Graph
        Set g = Project.Graphs.Add("s21 and s11",mwGT_Rectangular)
        g.Measurements.Add(s.Name,"DB(|S(1,1)|)")
        g.Measurements.Add(s.Name,"DB(|S(2,1)|)")
End Sub

Sub SetFrequencies
        Dim freqs() As Double

        fstart = 100
        fstop = 1000
        fstep = 10
        num = (fstop-fstart)/fstep
        ReDim freqs(num)
        Project.Frequencies.Clear
        cnt = 0
        For i = 100 To 1000 Step 10
                freqs(cnt) = i * 1e6 'enter in base units, Hz.
                cnt = cnt + 1
        Next i
        Project.Frequencies.AddMultiple(freqs)
End Sub

Sub SetValues(s As Schematic)
        'sets the parameter values of the elements to be non-default
        'values as double will always be in base units (Farads, Henries, etc)
        s.Elements("IND.L1").Parameters("L").ValueAsDouble = 15e-9
        s.Elements("IND.L2").Parameters("L").ValueAsDouble = 30e-9
        s.Elements("IND.L3").Parameters("L").ValueAsDouble = 30e-9
        s.Elements("IND.L4").Parameters("L").ValueAsDouble = 15e-9
        s.Elements("CAP.C1").Parameters("C").ValueAsDouble = 8e-12
        s.Elements("CAP.C2").Parameters("C").ValueAsDouble = 10e-12
        s.Elements("CAP.C3").Parameters("C").ValueAsDouble = 8e-12

End Sub

Sub BuildSchematic(s As Schematic)
        'add inductors
        'each visible grid in the schematic has a value of 100 for the coordinates
        s.Elements.Add("IND",0,0)
        s.Elements.Add("IND",1000,0)
        s.Elements.Add("IND",2000,0)
        s.Elements.Add("IND",3000,0)
        'add capacitors
        s.Elements.Add("CAP",1000,0,270)
        s.Elements.Add("CAP",2000,0,270)
        s.Elements.Add("CAP",3000,0,270)
        'add wire
        s.Wires.Add(1000,1000,3000,1000)
        'add ports
        s.Elements.Add("PORT",0,0)
        s.Elements.Add("PORT",4000,0,180)
        'add ground
        s.Elements.Add("GND",1000,1000)
End Sub

Function CreateSchematic() As Schematic
        Set CreateSchematic = Project.Schematics.Add("lpf")
End Function
Sub CleanUp
        Dim s As Schematic
        Dim g As Graph
        Dim opt As OptGoals

        For Each s In Project.Schematics
                Project.Schematics.Remove(s.Name)
        Next s

        For Each g In Project.Graphs
                Project.Graphs.Remove(g.Name)
        Next g

        Project.OptGoals.RemoveAll

        MWOffice.Windows.Close
End Sub
Sub ArrangeWindows
        Dim w As Window
        For Each w In MWOffice.Windows
                If w.Caption = "lpf" Then
                        w.ViewAll
                End If
        Next w

        MWOffice.Windows.Tile(mwWTD_Vertical)

End Sub
```

Python

```
 # -*- coding: utf-8 -*-

"Connecting to MWO"
import win32com.client as win32
import time
def Cleanup():
    for s in awr.Project.Schematics:
        awr.Project.Schematics.Remove(s.Name)
    for g in awr.Project.Graphs:
        awr.Project.Graphs.Remove(g.Name)
    awr.Project.OptGoals.RemoveAll
    awr.Windows.Close
def CreateSchematic():
    s=awr.Project.Schematics.Add("lpf")
    return s
def BuildSchematic(s):
    "add inductors"
    "each visible grid in the schematic has a value of 100 for the coordinates"
    awr.Project.Schematics(s.Name).Elements.Add("IND",0,0)
    awr.Project.Schematics(s.Name).Elements.Add("IND",1000,0)
    awr.Project.Schematics(s.Name).Elements.Add("IND",2000,0)
    awr.Project.Schematics(s.Name).Elements.Add("IND",3000,0)
    "add capacitors"
    awr.Project.Schematics(s.Name).Elements.Add("CAP",1000,0,270)
    awr.Project.Schematics(s.Name).Elements.Add("CAP",2000,0,270)
    awr.Project.Schematics(s.Name).Elements.Add("CAP",3000,0,270)
    "add wire"
    awr.Project.Schematics(s.Name).Wires.Add(1000,1000,3000,1000)
    "add ports"
    awr.Project.Schematics(s.Name).Elements.Add("PORT",0,0)
    awr.Project.Schematics(s.Name).Elements.Add("PORT",4000,0,180)
    "add ground"
    awr.Project.Schematics(s.Name).Elements.Add("GND",1000,1000)
def SetValues(s):
    "sets the parameter values of the elements to be non-default"
    "values as double will always be in base units (Farads, Henries, etc)"
    awr.Project.Schematics(s.Name).Elements("IND.L1").Parameters("L").ValueAsDouble = 15e-9
    awr.Project.Schematics(s.Name).Elements("IND.L2").Parameters("L").ValueAsDouble = 30e-9
    awr.Project.Schematics(s.Name).Elements("IND.L3").Parameters("L").ValueAsDouble = 30e-9
    awr.Project.Schematics(s.Name).Elements("IND.L4").Parameters("L").ValueAsDouble = 15e-9
    awr.Project.Schematics(s.Name).Elements("CAP.C1").Parameters("C").ValueAsDouble = 8e-12
    awr.Project.Schematics(s.Name).Elements("CAP.C2").Parameters("C").ValueAsDouble = 10e-12
    awr.Project.Schematics(s.Name).Elements("CAP.C3").Parameters("C").ValueAsDouble = 8e-12
def ArrangeWindows():
    for i in range(awr.Windows.Count):
        awr.Windows(i+1).ViewAll()
    awr.Windows.Tile(1)
def SetFrequencies():
    freqs=[]
    awr.Project.Frequencies.Clear
    for i in range(100, 1000, 10):
        freqs.append(i*1e6)
    awr.Project.Frequencies.AddMultiple(freqs)
def AddGraph(s):
    g = awr.Project.Graphs.Add("s21 and s11",3)
    g.Measurements.Add(s.Name,"DB(|S(1,1)|)")
    g.Measurements.Add(s.Name,"DB(|S(2,1)|)")
def Simulate():
    awr.Project.Simulator.Analyze()
def AddMarkers(s):
    m = awr.Project.Graphs(1).Markers.Add(s.Name + ":DB(|S(2,1)|)",1,0.1e9)
    m.Type = 3
    m.AutoSearch.mode=0
    m2 = awr.Project.Graphs(1).Markers.Add(s.Name + ":DB(|S(2,1)|)",1,0.5e9)
    m2.Type =4
    m2.Offset.Distance = -3
    m2.Offset.mode=1
    m2.Offset.ReferenceMarker="m1"
def AddEquations(s):
    awr.Project.Schematics(s.Name).Equations.Add("IND=15",1500,-1000)
    awr.Project.Schematics(s.Name).Equations.Add("CAP=8",2500,-1000)
    awr.Project.Schematics(s.Name).Elements("IND.L1").Parameters("L").ValueAsString = "IND"
    awr.Project.Schematics(s.Name).Elements("IND.L4").Parameters("L").ValueAsString = "IND"
    awr.Project.Schematics(s.Name).Elements("CAP.C1").Parameters("C").ValueAsString = "CAP"
    awr.Project.Schematics(s.Name).Elements("CAP.C3").Parameters("C").ValueAsString = "CAP"
def OptLimits(s):
    awr.Project.Schematics(s.Name).Equations("CAP").Optimize=True
    awr.Project.Schematics(s.Name).Equations("CAP").Constrain=True
    awr.Project.Schematics(s.Name).Equations("CAP").LowerConstraint = 8*0.75
    awr.Project.Schematics(s.Name).Equations("CAP").UpperConstraint = 8*1.25
    awr.Project.Schematics(s.Name).Equations("IND").Optimize=True
    awr.Project.Schematics(s.Name).Equations("IND").Constrain=True
    awr.Project.Schematics(s.Name).Equations("IND").LowerConstraint = 15*0.75
    awr.Project.Schematics(s.Name).Equations("IND").UpperConstraint = 15*1.25
    awr.Project.Schematics(s.Name).Elements("CAP.C2").Parameters("C").Optimize=True
    awr.Project.Schematics(s.Name).Elements("CAP.C2").Parameters("C").Constrain=True
    awr.Project.Schematics(s.Name).Elements("CAP.C2").Parameters("C").LowerConstraint = 10*0.75*1e-12
    awr.Project.Schematics(s.Name).Elements("CAP.C2").Parameters("C").UpperConstraint = 10*1.25*1e-12
def SetOptGoals(s):
    awr.Project.OptGoals.Add(s.Name,"DB(|S(1,1)|)",1,1,2,0,500e6,1,-17,-17,15)
    awr.Project.OptGoals.Add(s.Name,"DB(|S(2,1)|)",2,1,2,0,500e6,1,-1,-1,15)
    awr.Project.OptGoals.Add(s.Name,"DB(|S(2,1)|)",1,1,2,700e6,1000e6,1,-30,-30,15)
def Optimize():
    awr.Project.Optimizer.MaxIterations = 500
    awr.Project.Optimizer.Type = find_opt_type("Random (Local)")
    awr.Project.Optimizer.NewWindow()
    awr.Project.Optimizer.Start()
    while awr.Project.Optimizer.Running == True:
        time.sleep(0.5)
def find_opt_type(nm):
    typ = -1
    for i in range(awr.Project.Optimizer.TypeCount):
        if nm == awr.Project.Optimizer.TypeName(i+1):
            typ = i+1
    if typ == -1:
        print ("could not find optimizer name specified:" + nm)
    return typ
"main program"
awr = win32.Dispatch("MWOApp.MWOffice")
Cleanup()
sch = CreateSchematic()
BuildSchematic(sch)
SetValues(sch)
SetFrequencies()
AddGraph(sch)
Simulate()
AddMarkers(sch)
AddEquations(sch)
ArrangeWindows()
OptLimits(sch)
SetOptGoals(sch)
Optimize()
```