

Running Scripts in HyperLynx SI/PI

HyperLynx SI/PI is used to evaluate and validate a design with wide variety of simulation types. The design, be it in schematic or layout form, can be created or accessed to perform simulations by loading the application manually or by connecting to the application using an external script file. The script can use application objects to run both pre- and post- layout analysis.

There are four ways in which a script file can be executed.

- Using in-built script editor/debugger available in the application
- Using bsw.exe file of the application
- Using Windows-based or command-line-based script host
- Using Macros in Microsoft Excel

Before running the script file, a few important objects needs to be understood.

Application

It refers to the application object. In our document, the application is HyperLynx SI/PI. Thus, an object has to be created to initiate the application. The code below shows this.

```
Dim App: Set App = CreateObject("Hyperlynx.HLApplication")
```

In order to view the application as it starts, the visibility of the application can be enabled as shown below.

```
App.Visible = True
```

Design

It refers to the post-layout design in the form of .hyp or .cce. Various features in the design such as components, nets, etc are stored in this collection. The following code declares the object which contains the design elements.

```
Dim Design : Set Design = App.Design
```

Schematic

It refers to the object representing FFS schematic. Schematic can be loaded or created using this object. The object is created as follows.

```
Dim Schema : Set Schema = App.Schematic
```

File System Object

It allows to create and modify the output file. An object is created using CreateObject method with the properties of a system file. And this object is used to create and modify the file.

```
Dim fileObj : Set fileObj = CreateObject("Scripting.FileSystemObject")  
Dim reportFileObj : Set reportFileObj = fileObj.CreateTextFile("...\Loading_Comp_Output.txt", true)
```

I. Running script using in-built script debugger

The script file containing the rule can be opened using in-built debugger available in the application.

1. Opening the application – HyperLynx SI/PI

The application needs to be opened in order to use the script debugger window. Thus, creating an object explicitly in the script file to initiate the application is not necessary.

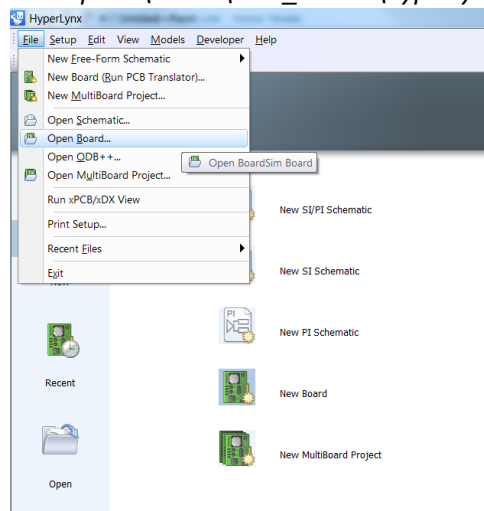
2. Loading the design board

This can be done in two ways.

a. Loading the design manually into the application

Select *File > Open Board >*

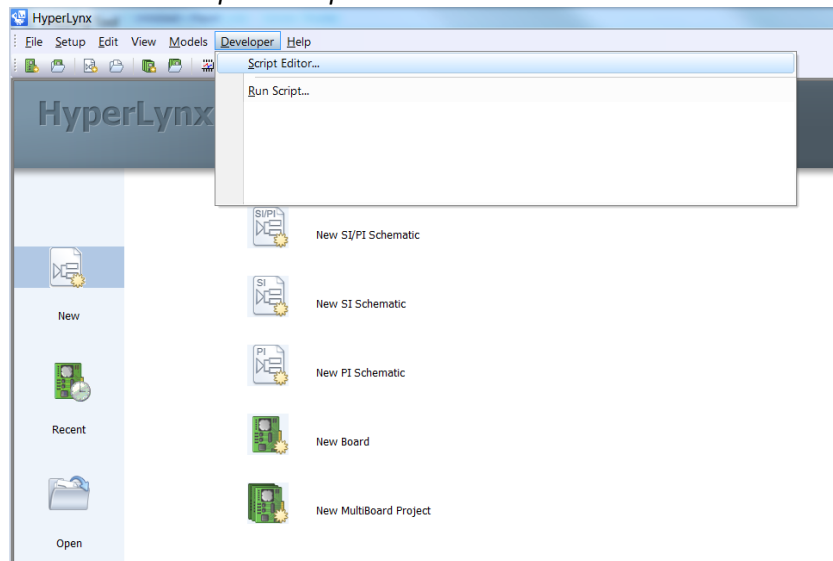
C:\MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\HypFiles\demo.hyp



b. Using *OpenFile* method in the scripting code

○ Opening Script file (.vbs file)

Select *Developer > Script Editor*



- Opening/Viewing Design file

In the script file, an object which refers to the application is created. This is followed by invoking the OpenFile method of the application which has path of the file as argument.

```
Dim App : Set App = Application
Dim sHYPfile: sHYPfile =
"C:\MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\HypFiles\demo.hyp"
App.OpenFile (sHYPfile)
```

In the code above, *App* is the object that refers to the application. *sHYPfile* holds the file path and *OpenFile* method is invoked using application object *App*.

3. Writing the script

Once the layout is opened, its elements such as components, nets, etc are extracted by creating design object.

```
Dim Design: Set Design = App.Design
```

The figure below shows the script used to list the components which has number of pins greater than 70 in the design.

```
Dim sProjectDir : sProjectDir =
"C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_Pi_Automation_Scripts\"
Dim sHYPfile : sHYPfile =
"C:\MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\HypFiles\demo.hyp"
Dim txtfile : txtfile = sProjectDir & "Loading_Comp_Output.txt"

Dim objFso : Set objFso =
CreateObject("Scripting.FileSystemObject")
Dim objReportFile : Set objReportFile =
objFso.CreateTextFile(txtfile , true)

Dim App : Set App = Application
App.OpenFile (sHYPfile)
Dim Design : Set Design = App.Design
Dim Comps : Set Comps = Design.Components
NumofPins =70

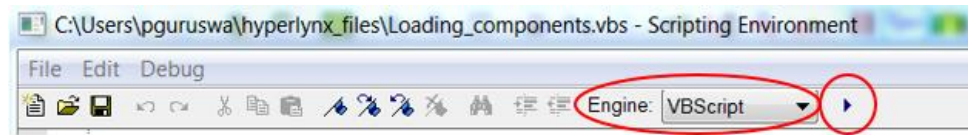
objReportFile.Write "Components with number of pins greater than
70" & vbNewLine

for k = 1 to Comps.Count
    if ( Comps.Item(k).Pins.Count > NumofPins ) then
        objReportFile.Write Comps.Item(k).RefDes & vbNewLine
    end if
```

next

4. Executing the script file

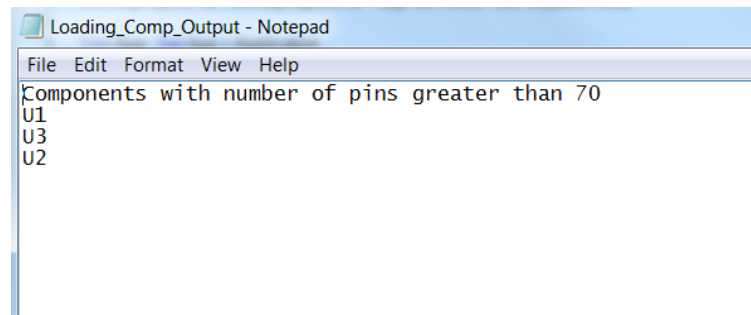
The script file containing the rule is executed by clicking the execute button. The script engine can be changed according to the scripting language (VBScript or Jscript) used.



5. Viewing the output

The output can be viewed in any file as desired. In the script file shown above, a text file named *Loading_Comp_Output.txt* is created using *CreateTextFile* method. Hence, as the script file is executed, the output is written into this file.

The figure shows the output seen in the text file which is the list of components with number of pins greater than 70.



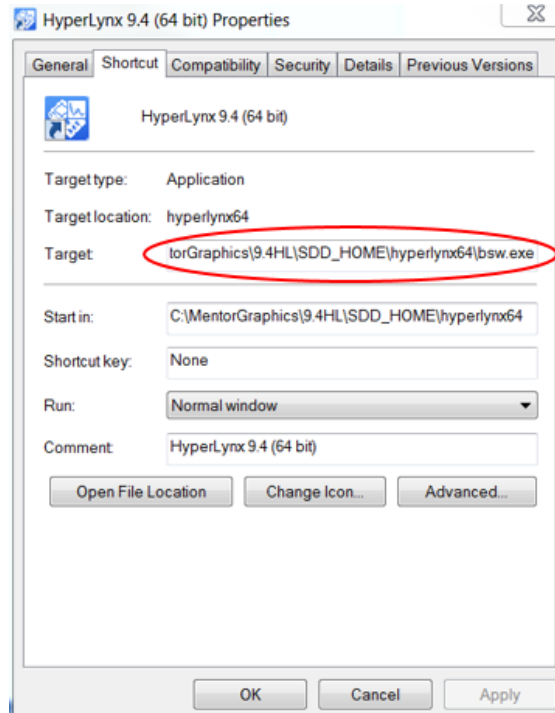
II. Running script using bsw.exe file of the application

Another way of executing the script file without creating an object to initiate the application is by using bsw.exe file.

1. Finding the path of bsw.exe file

One way of finding the path is by doing the following.

- a. Go to *Start* menu
- b. Find *HyperLynx* application and right click to select *Properties*. *Hyperlynx Properties* window opens with *Shortcut* tab selected
- c. In *Target* column, the *path* of bsw.exe file could be found



2. Executing script file
 - a. Open command window
 - b. Type the path of bsw.exe file, followed by a space, type "-run:" and full path of the script file with the file extension
 - c. Click Enter and the script starts executing

```
C:\>MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\bsw.exe -run:C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_PI_Automation_Scripts>Loading_components-bsw_run.vbs
```

III. Running script using Windows-based or command-line-based script host

The script file can also be executed from command prompt using windows-based script host or command-line-based script host.

1. Creating object to initiate the application

The code in the script file remains the same except for creation of an object to initiate the application.

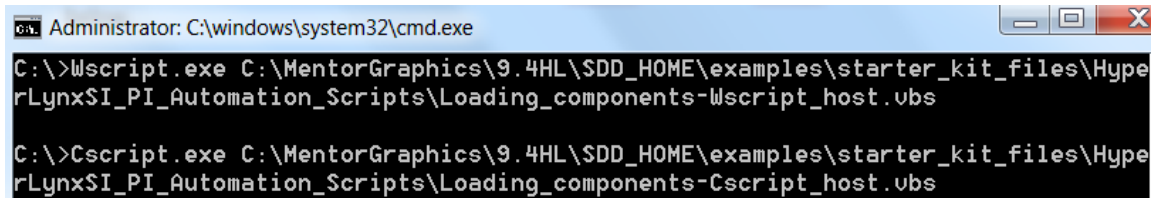
Dim App : Set App = Application
is replaced by

Dim App : Set App = CreateObject("Hyperlynx.HLApplication")

Windows starts HyperLynx SI/PI in an embedded mode that makes the user interface invisible. This behavior with the Visible property of the Application object can be overridden.

2. Executing the script file

The script file is executed by typing in the command prompt, the name of the Windows or command line Host executable file (Wscript.exe or Cscript.exe), followed by a space and full path name of the script file as shown below. File name extension of the script file should also be included. On clicking Enter, the file execution begins.



```
Administrator: C:\windows\system32\cmd.exe
C:\>Wscript.exe C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_Pi_Automation_Scripts>Loading_components-Wscript_host.vbs
C:\>Cscript.exe C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_Pi_Automation_Scripts>Loading_components-Cscript_host.vbs
```

3. Viewing output in command window

To view the output in the command window, the StdOut method of Cscript file is used as shown below.

```
Dim stdOutObj: Set stdOutObj = WScript.StdOut
stdOutObj.Write " Components with number of pins greater than 70 "
```

The figure below shows the modification in the code to initiate the application HyperLynx SI/PI and print result in the command window.

```
Dim sProjectDir : sProjectDir =
"C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_Pi_Automation_Scripts\"
Dim sHYPfile : sHYPfile =
"C:\MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\HypFiles\demo.hyp"
Dim txtfile : txtfile = sProjectDir & "Loading_Comp_Output.txt"

Dim objFso : Set objFso =
CreateObject("Scripting.FileSystemObject")
Dim objReportFile : Set objReportFile =
objFso.CreateTextFile(txtfile , true)

Dim App : Set App = CreateObject("Hyperlynx.HLApplication")
App.Visible = True
App.OpenFile (sHYPfile)
Dim Design : Set Design = App.Design
Dim Comps : Set Comps = Design.Components
NumofPins =70

objReportFile.Write "Components with number of pins greater than
70" & vbNewLine

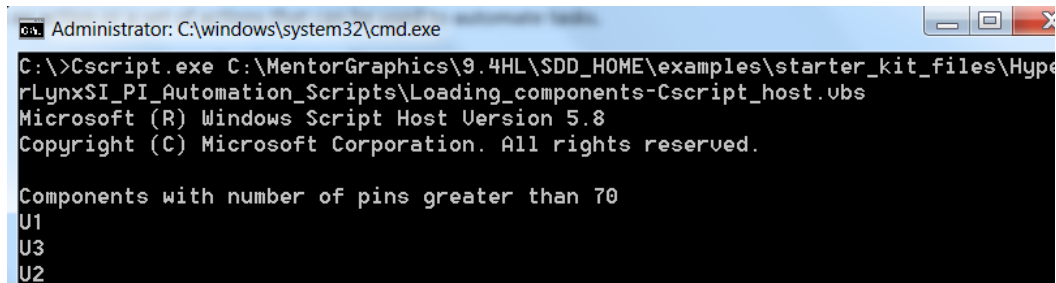
Dim stdOutObj: Set stdOutObj = WScript.StdOut
stdOutObj.Write "Components with number of pins greater than 70"
& vbNewLine
```

```

for k = 1 to Comps.Count
    if ( Comps.Item(k).Pins.Count > NumofPins ) then
        objReportFile.Write Comps.Item(k).RefDes & vbNewLine
        stdOutObj.Write Comps.Item(k).RefDes & vbNewLine
    end if
next

```

The result appears in the command prompt as follows.



```

Administrator: C:\windows\system32\cmd.exe

C:\>Cscript.exe C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperLynxSI_Pi_Automation_Scripts>Loading_components-Cscript_host.vbs
Microsoft (R) Windows Script Host Version 5.8
Copyright (C) Microsoft Corporation. All rights reserved.

Components with number of pins greater than 70
U1
U3
U2

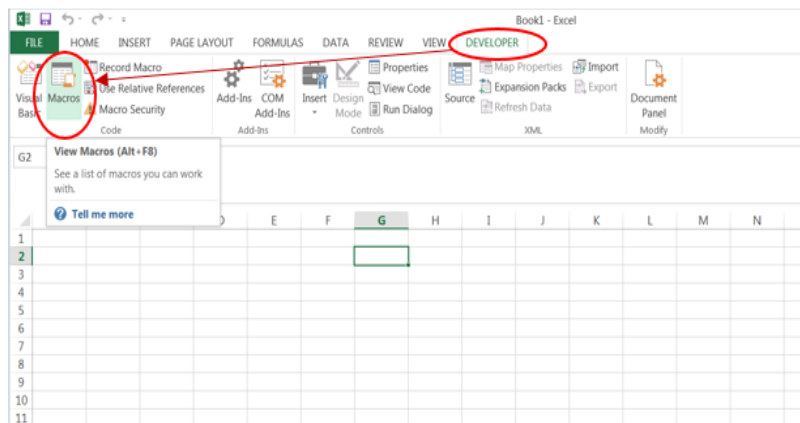
```

IV. Running script using macros in Excel VBA

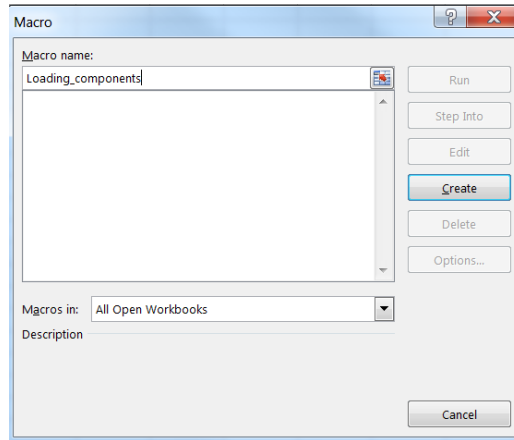
Another way of executing the script file is using macros in Excel Visual Basics for Applications (VBA). Macro is an action or a set of actions that can be used to automate tasks.

1. Creating macro for the design file

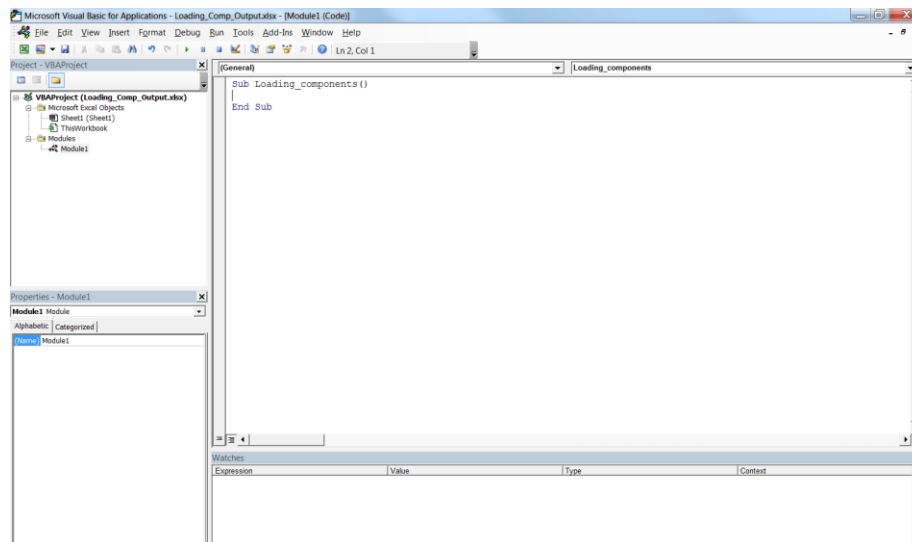
- a. Open Microsoft Excel
- b. On the Developer tab, click Macros.




- c. A window to create macros appears. In the Macro name tab, enter the desired name and click Create.



This opens a new Microsoft Visual Basic window with a method having the name as the macro name. The code in the script file can be copied and pasted inside this method.



2. Executing script file

The macro containing the script can be executed by clicking the run  button seen in VBA window.

3. Viewing the result

The result can be viewed in any file as in Section I. In order to make the output more presentable, excel file is used.

Instead of using *FileSystemObject* to create the text file, an object with *Excel.Application* is created to initiate Microsoft Excel.

```
Dim objFso: Set objFso = CreateObject("Excel.Application")
```

The excel file which is already created is opened using *Workbooks.Open* method. An object is created to access the worksheet and using this object the cells are updated with output data.

```
Dim objReportFile: Set objReportFile = objFso.Workbooks.Open(excelfile)
```



```
Set objReportWorkSheet = objReportFile.Worksheets(1)
objReportWorkSheet.Cells(1, 1) = "Components with number of pins greater than 70"
```

The script below shows the macro used to display the components with number of pins greater than 70 in excel file.

```
Sub Loading_Components()
Dim sProjectDir: sProjectDir =
"C:\MentorGraphics\9.4HL\SDD_HOME\examples\starter_kit_files\HyperlynxSI_Pi_Automation_Scripts\"
Dim sHYPfile: sHYPfile =
"C:\MentorGraphics\9.4HL\SDD_HOME\hyperlynx64\HypFiles\demo.hyp"
Dim excelFile: excelFile = sProjectDir &
"Loading_Comp_Output.xlsx"

Dim App: Set App = CreateObject("Hyperlynx.HLApplication")
App.Visible = True
App.OpenFile (sHYPfile)
Dim Design: Set Design = App.Design
Dim Comps: Set Comps = Design.Components
NumofPins = 70

Dim objFso: Set objFso = CreateObject("Excel.Application")
Dim objReportFile: Set objReportFile =
objFso.Workbooks.Open(excelFile)
Set objReportWorkSheet = objReportFile.Worksheets(1)
objFso.Visible = True
Count = 0

objReportWorkSheet.Cells(1, 1) = "Components with number of pins
greater than 70"

For k = 1 To Comps.Count
    If (Comps.Item(k).Pins.Count > NumofPins) Then
        Count = Count + 1
        objReportWorkSheet.Cells(Count + 1, 1) =
Comps.Item(k).RefDes
    End If
Next

objReportFile.Save
End Sub
```

The output looks as follows.

