



# VISUALSIM TRAINING

---

# Agenda- Part 2: About VisualSim

---

Introduction to VisualSim

Documentation and Support

Classes, User Library and Version Control

Intelligent Simulation using Post Processor

- Batch-mode simulation
- Multi-simulation
- Diagnostics

# Documentation and Support

# Documentation and Support



Documentation and Tutorials

Post Processor

Online Support Portal

VisualSim Architect

MIRABILIS design

VisualSim® Architect

New Model Open Model Text Editor Expression Evaluator

Application Library

Recent Pinned Auto Save

- Back\_to\_Back\_Read\_IBM.xml
- AHB\_12.xml
- AHB\_12.xml
- Exec\_Instr1(1).xml
- Exec\_Instr1(3).xml
- Snooping\_Protocol.xml
- Library\_Model.html
- Scheduler\_HW\_test1\_20210513\_152006.x...
- Application\_Model.html
- Brake\_CAN\_model.xml
- bruteForce.xml
- bruteForce\_with\_insightEngine.xml

version: 2140\_64b  
October 9, 2021 16:45 PT

Demo models for Libraries and Applications

Pinned Models

Recent Models



# Documentation and Support

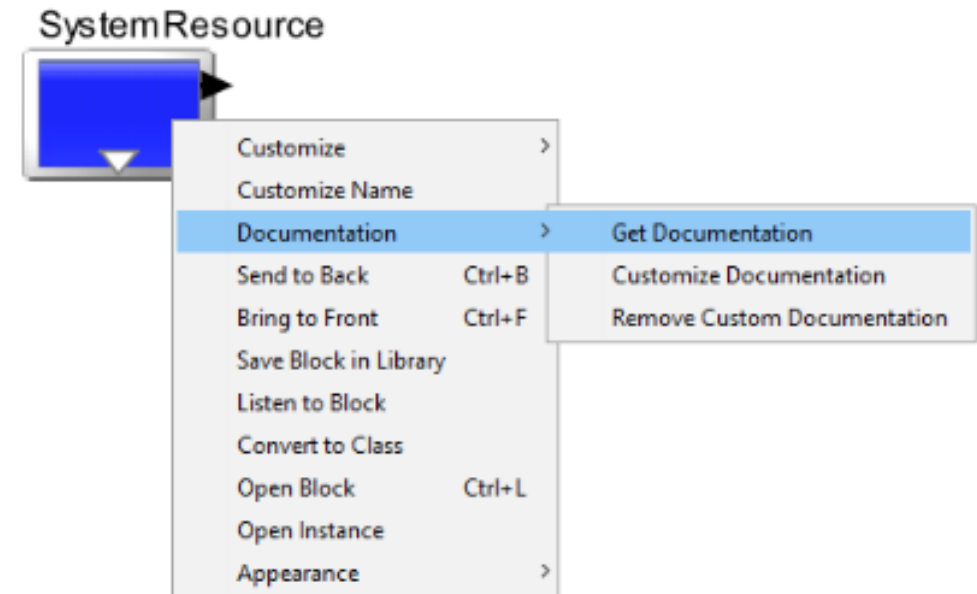
---

## • Block Description

Right click on the block and Select

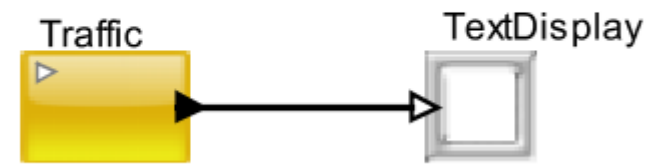
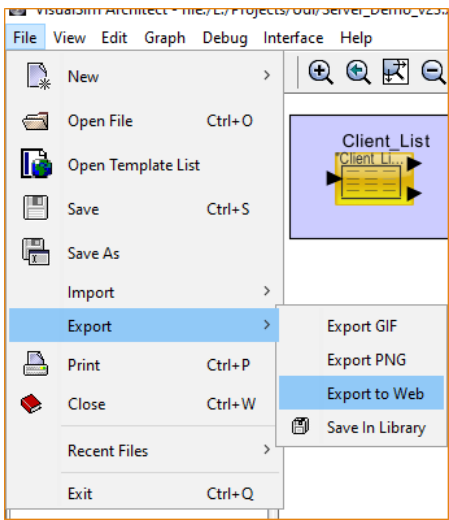
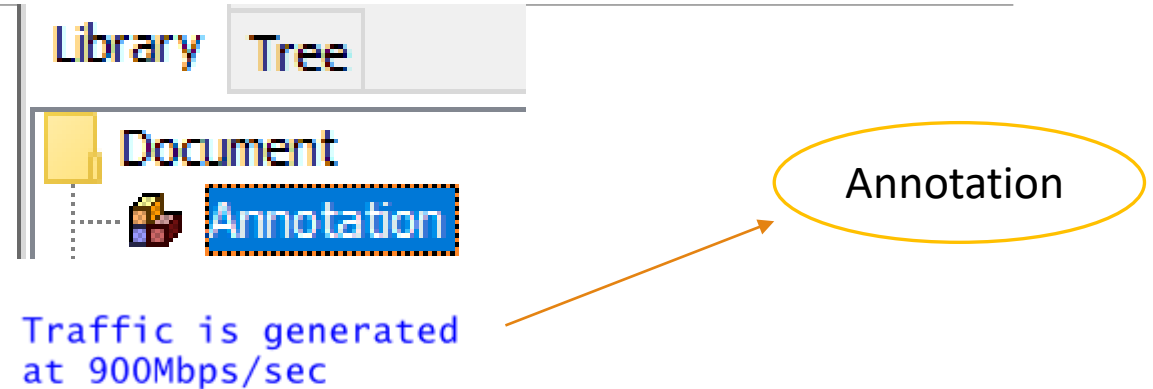
**Documentation -> Get Documentation**

- The Customize Documentation enables the user to override the default documentation and add their own information



# Documentation and Support

- User Documentation
  - ✓ Short Description in the model
  - ✓ Will be displayed like a Block Parameter
- Export to Web
  - ✓ HTML & jnlp files created
  - ✓ The document can be shared between users.



# Introduction to VisualSim



# VisualSim Homepage and Documentations

The screenshot displays the VisualSim Architect software interface. On the left is a sidebar with the following menu items: **Models**, **Learn VisualSim**, **Post Processor**, **Parser**, **Settings**, **About**, and **Contact Support**. At the bottom of the sidebar, it shows the version *2140\_64b* and the date **October 9, 2021 16:45 PT**.

The main window features a top toolbar with icons for **New Model**, **Open Model**, **Text Editor**, and **Expression Evaluator**. Below the toolbar is a search bar and tabs for **Recent**, **Pinned**, and **Auto Save**. The central area is a file explorer listing various XML and HTML files, including:

- Back\_to\_Back\_Read\_IBM.xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>demo>>Bus\_Std>>Back\_to\_Back\_Read\_IBM.xml)
- AHB\_12.xml** (C:\>>VisualSim>>VisualSim2130\_64>>VS\_AR>>demo>>Bus\_Std>>AHB\_AXI>>AHB\_12.xml)
- AHB\_12.xml** (C:\>>VisualSim>>VisualSim2030\_64>>VS\_AR>>demo>>Bus\_Std>>AHB\_AXI>>AHB\_12.xml)
- Exec\_Instr1(1).xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>lib>>mohini>>oct>>instr\_beh>>Exec\_Instr1(1).xml) *today at 12:15 PM*
- Exec\_Instr1(3).xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>lib>>mohini>>oct>>instr\_beh>>Exec\_Instr1(3).xml)
- Snooping\_Protocol.xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>demo>>Coherency>>Snooping\_Protocol.xml) *today at 11:11 AM*
- Library\_Model.html** (D:\>>ELC\_Office>>Tool>>VS\_TESTsep>>VS\_AR>>doc>>pre-built\_models>>html>>Library\_Model.html) *today at 10:52 AM*
- Scheduler\_HW\_test1\_20210513\_152006.x...** (C:\>>Users>>mohin>>.VisualSim>>Scheduler\_HW\_test1\_20210513\_152006.xml.xml) *today at 10:49 AM*
- Application\_Model.html** (D:\>>ELC\_Office>>Tool>>VS\_TESTsep>>VS\_AR>>doc>>pre-built\_models>>html>>Application\_Model.html) *today at 10:48 AM*
- Brake\_CAN\_model.xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>demo>>automotive>>CAN\_Brake>>Brake\_CAN\_model.xml) *today at 10:44 AM*
- bruteForce.xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>lib>>mohini>>oct>>cyberAttacks>>bruteForce>>gem5>>bruteForce.xml) *today at 10:43 AM*
- bruteForce\_with\_insightEngine.xml** (D:\>>ELC\_Office>>Tool>>VisualSim>>VS\_AR>>lib>>mohini>>oct>>cyberAttacks>>bruteForce>>gem5>>brute\_diagnostic>>bruteForce\_with\_insightEngine.xml) *today at 10:43 AM*

On the right side of the main window, there are two large circular buttons: **Application** (green) and **Library** (yellow).



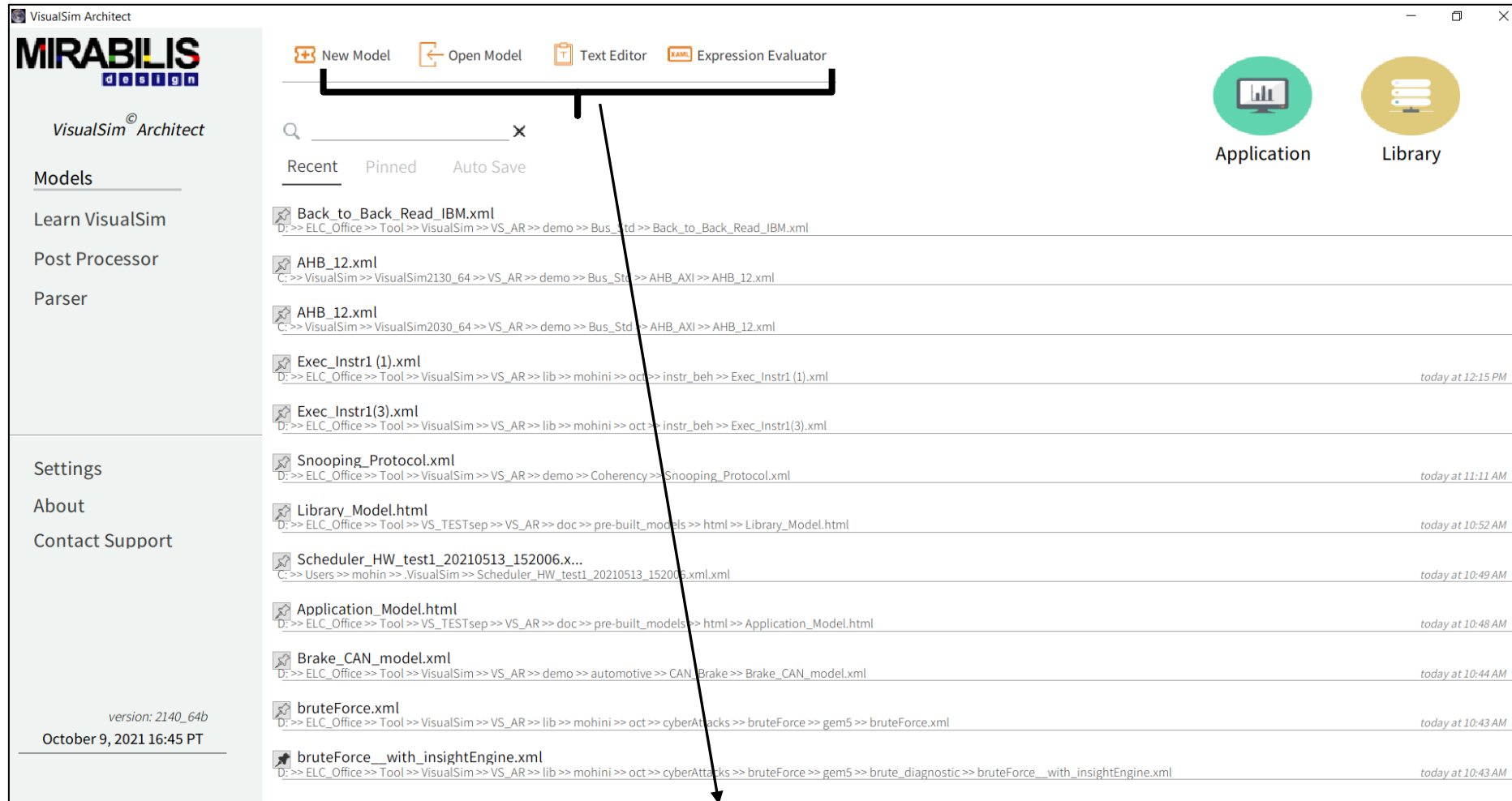


# Tool Organization

---

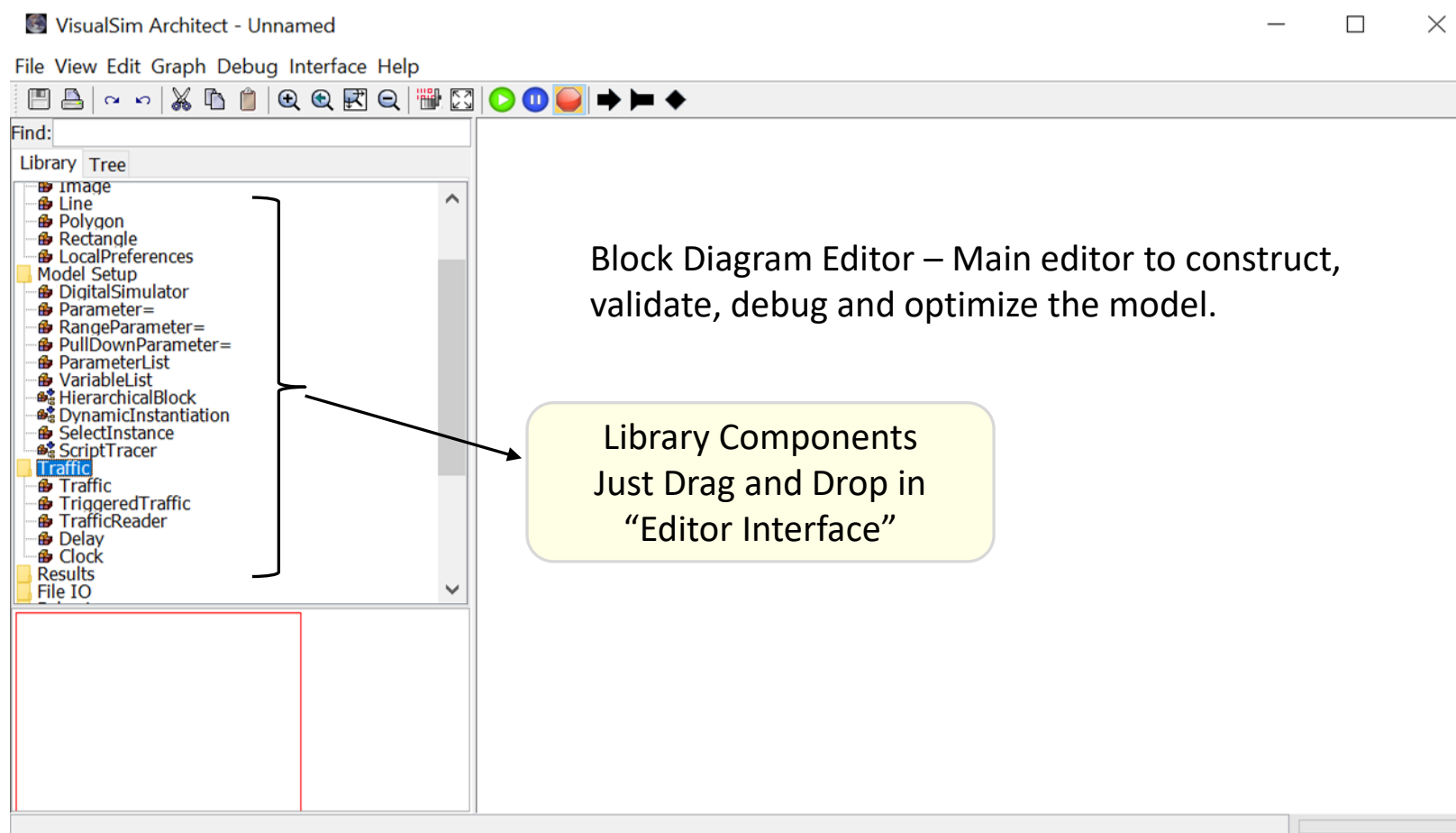
- VisualSim Starting Page
- VisualSim Architect
  - ✓ Block Diagram editor
  - ✓ Finite State Machine Editor
  - ✓ Expression Evaluator
  - ✓ Text Editor
  - ✓ HTML Editor
- VisualSim Modeling Components
  - ✓ Library: Hardware, Resources, Script.
  - ✓ Interfaces: Verilog, SystemC, C++.
  - ✓ Utilities: PowerTable
  - ✓ Technologies: DDR3, Cache and PCIe
- VisualSim reports and Statistics
- VisualSim Post-Processor
- VisualSim Diagnostic Engine
- VisualSim Cloud
- VisualSim Batch Simulation
- VisualSim multi-core execution

# VisualSim Architect- Core Menu Items

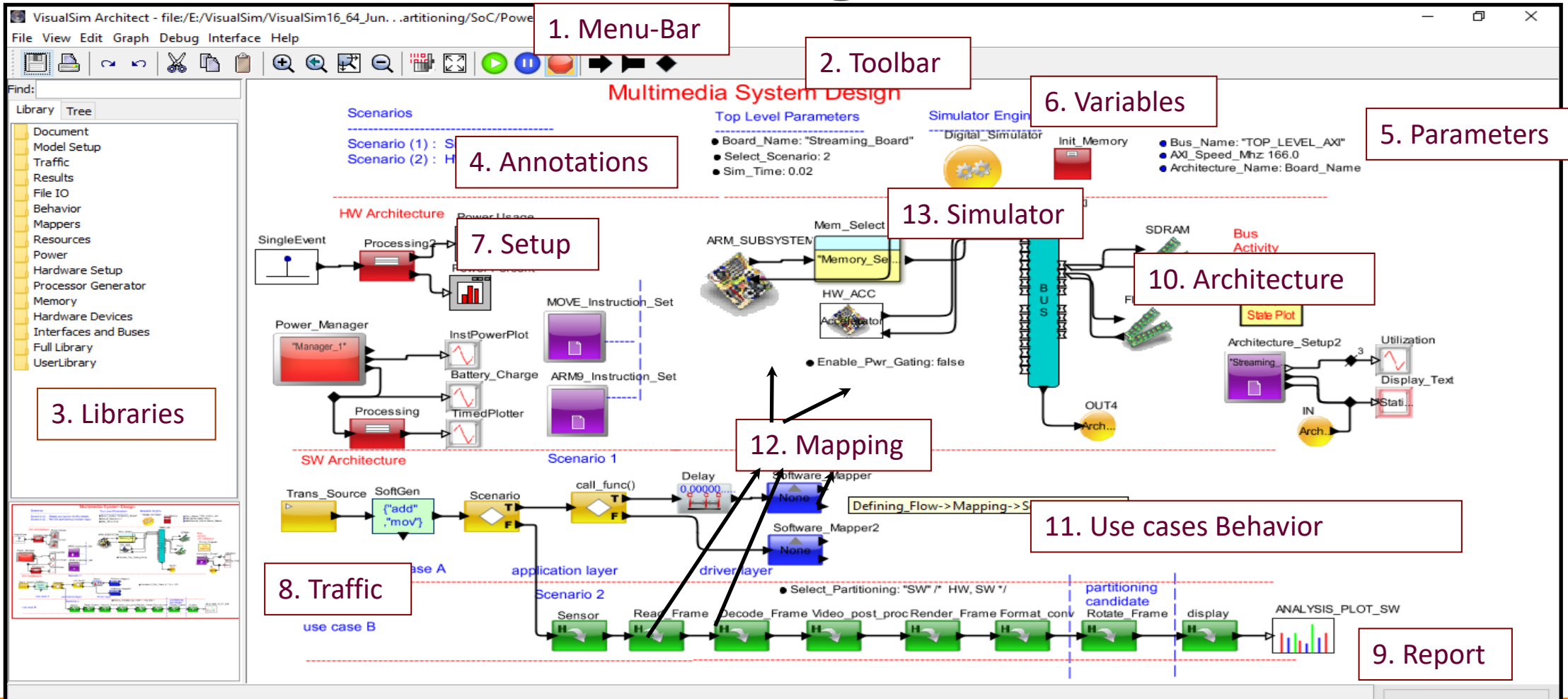


- Block Diagram Editor
- Text Editor – Extensively used for authoring code in the script
- Expression Evaluator – Evaluate Equations and Algorithms, can be used for conducting quick studies

# Block Diagram Editor

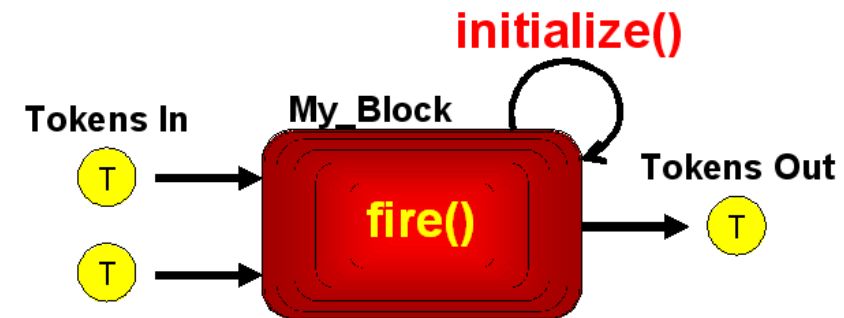
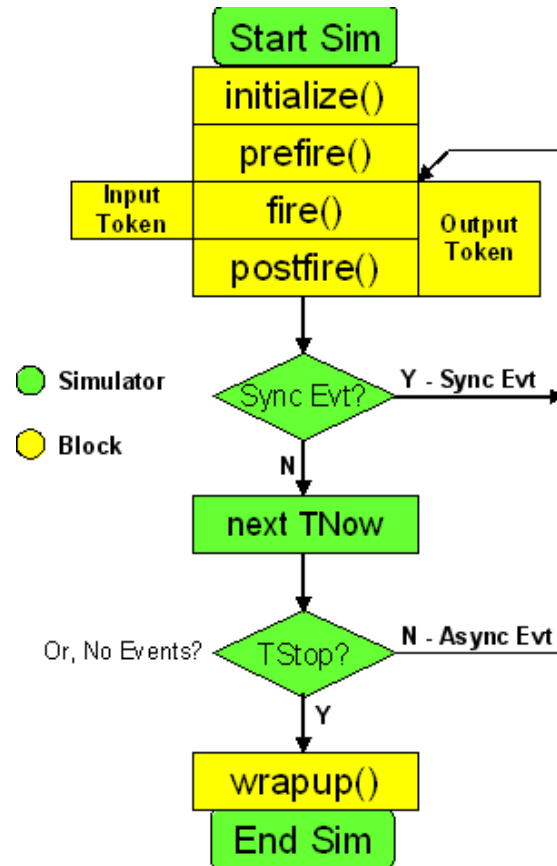


# Parts of the Block Diagram



# Model and Block

# Discrete-Event and Block Execution



# Simulation Engines (or) Models of Computation



# What are Models of Computation?

---

- Govern interaction of components in a model
- VisualSim includes
  - ✓ Digital Simulator (Discrete-Event) & Finite State Machine
  - ✓ Continuous Time & Untimed Digital (Synchronous Data Flow)
- Selection of correct model of computation
  - ✓ Discrete Event for Performance architecture analysis
  - ✓ Finite State Machine for protocols and control
  - ✓ Synchronous Dataflow for DSP and imaging algorithms and software functionality
  - ✓ Continuous Time for analog and waveform-based evaluations





# Simulation Polymorphism

---

- Simulators share kernel, event queue & library
- Use the domain that fits the question
- Transfer of control/time/data between simulators is seamless and invisible to user
- Components are created once and used extensively

# Simulation and different Models of Computation

---

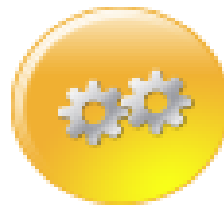
Digital Simulator



**Discrete Event  
Simulator Object**

**Timed Simulation**

Untimed Digital



**Synchronous Data Flow  
Simulator Object**

**Untimed DSP Algorithms**

Continuous Time



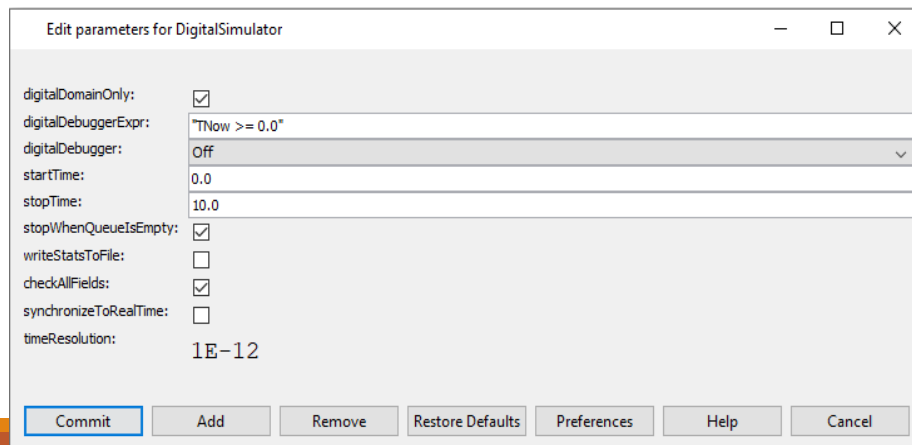
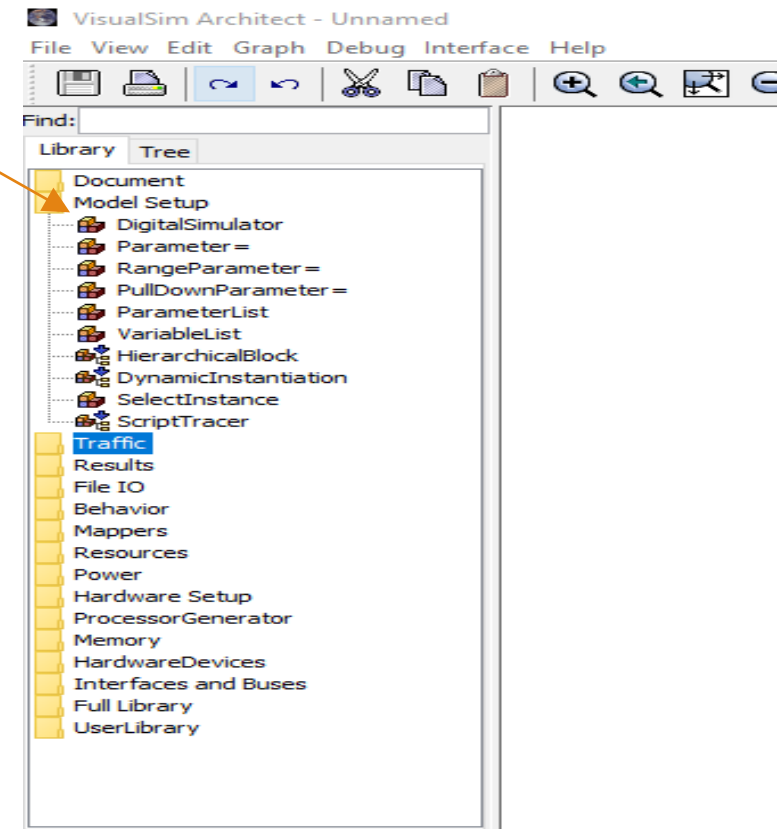
**Analog Simulator  
Object**

**Analog Simulation**

# Digital Simulator

## Where to get digital simulator?

- Integral component for every digital domain designs
- Length of simulation is specified
- Definition - Execution of synchronous events between asynchronous time
- Calendar maintains sequences of events on a timeline
- Deterministic semantics to simultaneous events - Creates a block firing list in data-precedence order
- Performance modeling uses Discrete-Event
- VisualSim has two Digital Simulators
  - ✓ Default is a high performance simulator that does not support the use of other simulators in the same model
  - ✓ Uncheck the Digital Only in the Digital block and you will get flexible simulator that can reside within or can envelope other simulators
- WriteStatsToFile – Generates Statistics for all the blocks in the model and writes into a text file



# Discrete Event Calendar Example

Time	Event	Action	# in Sys	# in Queue
0	E1 arrives	Schedule next arrival – IAT=2	1	0
		E1 start processing – time = 4		
2	E2 arrives	Schedule next arrival – IAT=2	2	1
		E2 enters queue		
4	E1 finishes	E1 depart	1	0
		E2 start processing – time=2		
4	E3 arrives	Schedule next arrival – IAT=1	2	1
		E3 enters queue		
5	E4 arrives	Schedule next arrival – IAT=3	3	2
		E4 enters queue		
6	E2 finishes	E2 departs	2	1
		E3 starts processing – time =5		
8	E5 arrives	Schedule next arrival – IAT=7	3	2
		E5 enters queue		
11	E3 finishes	E3 departs	2	1
		E4 starts processing – time=2		

Time	Event	Action	# in Sys	# in Queue
13	E4 finishes	E4 departs	1	0
		E5 starts processing – time=4		
15	E6 arrives	Schedule next arrival – IAT=5	2	1
		E6 enters queue		
17	E5 finishes	E5 departs	1	0
		E6 starts processing – time=1		
18	E6 finishes	E6 departs	0	0
20	E7 arrives	Schedule next arrival-IAT=10	1	0
		E7 starts processing – time=5		
25	E7 finishes	E7 departs	0	0
30	E8 arrives	Schedule next arrival – IAT=11	1	0
		E8 starts processing – time=8		
38	E8 finishes	E8 departs	0	0
40	End simulation			



# Untimed Digital or Synchronous Data Flow

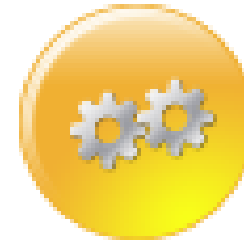
---

- Definition

- ✓ Can specify *a priori* the number of input samples consumed on each input and the number of output samples produced on each output each time the block is invoked.

- Execution is statistical determined prior to execution
- Deadlock and boundedness are decidable
- Used for embedded real-time software, DSP and hardware
- **Full Library -> Model - > Simulator - > Untimed Digital**

## Untimed Digital



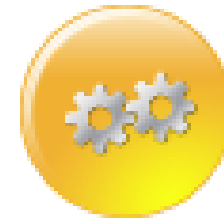


# Continuous Time

---

- Definition
  - ✓ Simulator finds a fixed-point, i.e., a set of continuous-time functions that satisfy all the relations
  - ✓ Solve algebraic or differential relations between inputs and outputs
- Represent components that interact via continuous-time signals
- Closely related to Spice circuit simulators and similar to Simulink, Saber, and VHDL-AMS
- Defines physical systems with linear or nonlinear algebraic/ differential equation descriptions
- Used for analog and control systems
- **Full Library -> Model - > Analog and CTRL- > Continuous Time**

Continuous Time



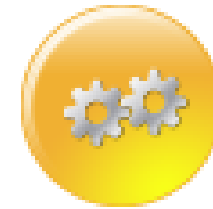


# Finite State Machine

---

- Definition
  - ✓ Case statement in a graphical form defining sequential control logic
- Entities represent states and connections represent transitions between states
- Execution is a strictly ordered sequence of state transitions
- Control and protocol modeling uses FSMs
- Can be reactive or token-based firing
- **Full Library -> Model - > FSM- > FSM Simulator**

FSM Simulator





# Multi-Domain Simulations

---

- Analog with digital electronics is mixed-signal
  - ✓ Continuous time + Discrete event
- FSM with concurrent model of computation
  - ✓ With DigitalSimulator yield Control flow
  - ✓ With ContinuousTime yield hybrid systems
  - ✓ With UntimedDigital yield Statecharts
- ContinuousTime or DigitalSimulator in FSM creates modal models
  - ✓ Simple models are valid over a fixed region
  - ✓ Modal model provides system transition between these models
- UntimedDigital with DigitalSimulator or ContinuousTime
  - ✓ Physical Layer and MAC Layer
  - ✓ Baseband and RF



# Selecting the Right Continuous Time and FSM Simulator

---

- Continuous Time
  - ✓ Top-level must use CoTMultisolver
  - ✓ Inside a ContinuousTime or DigitalSimulator can be CT Embedded or CT MultiSolver
  - ✓ Inside an FSM is CT Embedded but top-level containing FSM must be ContinuousTime/DigitalSimulator
- FSM-Hierarchical
  - ✓ Can be a FSM or an embedded FSM
  - ✓ Top-level Simulator is selected based on the mode of operation required
  - ✓ Refinement can be an FSM or any other simulator

# Events and Delays

# Time vs. Events

---

- Time is a fixed delay and is loaded into the simulator calendar
  - ✓ Used to model cycles or latency
  - ✓ Can generate one output
- Event is a trigger that is generated by the user for an instant response
  - ✓ Used to model a acknowledge or handshake
  - ✓ Can trigger multiple blocks to fire at the same time



# Delay- Review

---

- Concept is to schedule the data structure to continue execution some time in the future
- Takes the incoming data structure and sends it on the output port after certain amount of time
- Usage
  - ✓ Time to process in a resource such as hardware, mechanical and other components
  - ✓ Time between packets entering a network or a system
  - ✓ Time taken to execute a job or task
  - ✓ Special case: When there is a loop in the model, a Delay with 0.0 time must be specified to prevent an infinite loop forming.
- Used with additional blocks
  - ✓ Queue + delay can be used to model a bus, processor, stage in a pipeline

# Delay - Continued

Delay incoming Data Structure or token

- by a variable time period



Edit parameters for Delay

Block\_Documentation:  Enter User Documentation Here

Delay\_Value:



# Delay- Application

---

- System
  - ✓ Example: Application Demo->System->Functional->1\_1->1.1.1. The Delay is associated with the processing time for this decision logic.
  - ✓ Example: Application Demo->Aerospace->ACS->DCGS. The Delay block is used to emulate the time across the network
- Network
  - ✓ Example: Application Demo->Standards->SPI3.0. There is delay block that is the time associated with the Link Layer.

# Delay- Application

---

- Hardware
  - ✓ Example: Application Demo->Hardware->SoC Power Modeling. Here the Delay block is used to delay the time between bursts of activity in the traffic generation.
  - ✓ Example: Application Demo->Computer->Disk. Inside the Disk block, it used as the Delay for moving the spindle
- Software
  - ✓ Example: Delay is not used for the time associated with processing a task or for the scheduling the process. This is because the order is not preserved and the Delay schedules a task immediately. It is possible to use the Queue + Delay. In that case, there might be additional logic after the Queue block.

# Ports



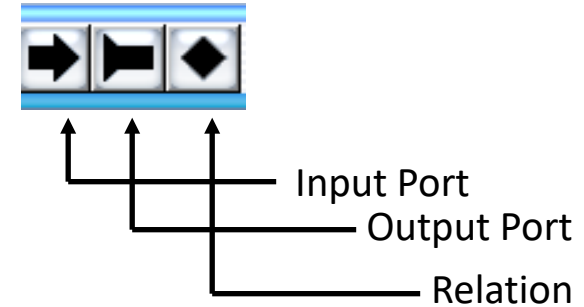
# Port Details

## Configure Ports

Edit Parameters  
Configure Ports

Name	Input	Output	Multiport	Type	Direction	Show ...	Hide	Units	Re...
input	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		DEFAULT	<input type="checkbox"/>	<input type="checkbox"/>		
output	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		DEFAULT	<input type="checkbox"/>	<input type="checkbox"/>		
sync	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	general	SOUTH	<input type="checkbox"/>	<input type="checkbox"/>		

Commit Apply Add Remove Help Cancel

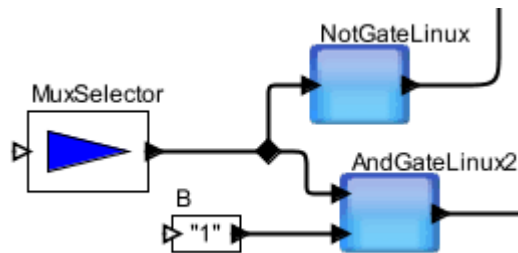


## Block Context Menu

Hierarchical\_Block

- Customize
- Customize Name
- Documentation
- Send to Back   Ctrl+B
- Bring to Front   Ctrl+F
- Save Block In Library
- Listen to Block
- Convert to Class
- Open Block   Ctrl+L
- Open Instance
- Appearance

## Relations and Connections

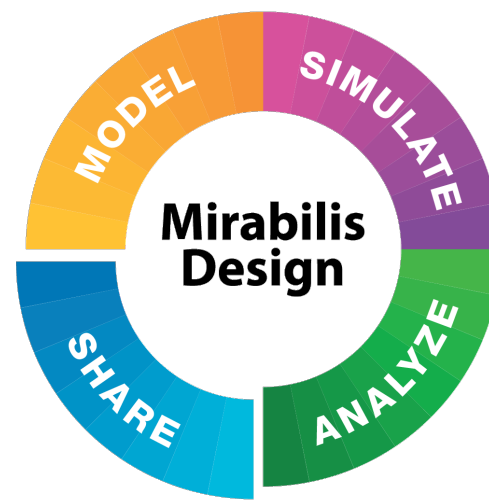




# Port Types

---

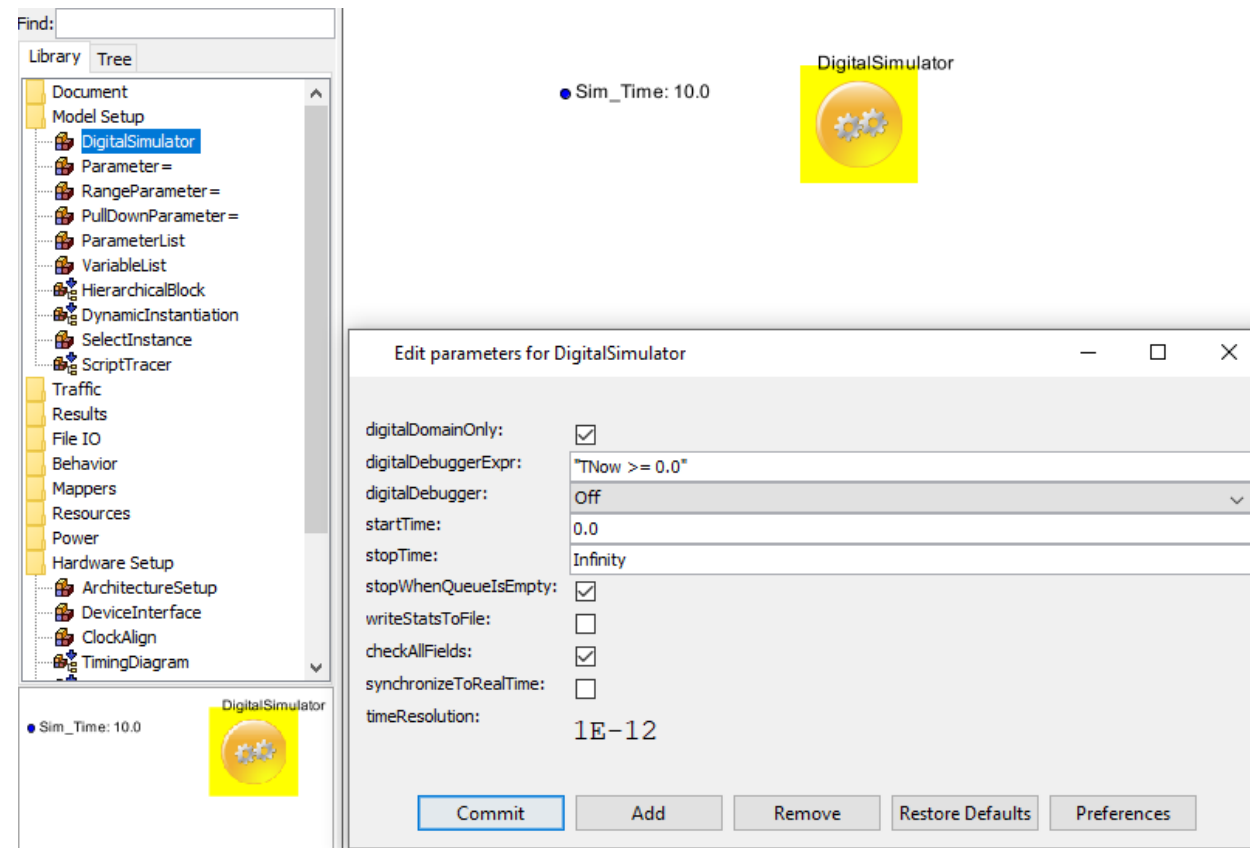
- Data types are polymorphic identified at the ports
- Ports can be specified as: int, double, long, float, boolean, string, arrays or data structure
  - ✓ **Note:** Port types on either side of link must match
- Ports can also be unspecified: general or unknown
  - ✓ Simulator dynamically adapts to the connected ports, if both sides are unspecified
  - ✓ User-added ports must always be set to a minimum of General
- Port types of library blocks
  - ✓ If a port type is fixed such as int, string or double, this cannot be modified and the port connected to it must be modified
  - ✓ **Exceptional Cases:** Only unknown and general can be modified



# CLASSES, USER LIBRARIES AND VERSION CONTROL

---

# Creating a Block Diagram



The screenshot displays a software development environment for creating a block diagram. On the left, a 'Library' pane lists various components under categories like 'Document', 'Model Setup', 'Traffic', and 'Hardware Setup'. The 'DigitalSimulator' component is highlighted. In the center, a block diagram shows a 'DigitalSimulator' block with a 'Sim\_Time: 10.0' parameter. An 'Edit parameters for DigitalSimulator' dialog box is open, showing the following settings:

Parameter	Value
digitalDomainOnly:	<input checked="" type="checkbox"/>
digitalDebuggerExpr:	"TNow >= 0.0"
digitalDebugger:	Off
startTime:	0.0
stopTime:	Infinity
stopWhenQueueIsEmpty:	<input checked="" type="checkbox"/>
writeStatsToFile:	<input type="checkbox"/>
checkAllFields:	<input checked="" type="checkbox"/>
synchronizeToRealTime:	<input type="checkbox"/>
timeResolution:	1E-12

At the bottom of the dialog, there are buttons for 'Commit', 'Add', 'Remove', 'Restore Defaults', and 'Preferences'.



# Digital Simulator

Edit parameters for DigitalSimulator

digitalDomainOnly:

digitalDebuggerExpr: "TNow >= 0.0"

digitalDebugger: Off

startTime: 0.0

stopTime: Infinity

stopWhenQueueIsEmpty:

writeStatsToFile:

checkAllFields:

synchronizeToRealTime:

timeResolution: 1E-12

Commit Add Remove Restore Defaul... Preferences Help Cancel

Start and Stop time of Model

Specifies whether to write the statistics for all blocks in the model to the \_results folder of the model Directory



# Connecting the Blocks

---



# Block Parameter Settings

Traffic

Edit parameters for Traffic

Block\_Documentation:  Enter User Documentation Here

Data\_Structure\_Name: "Header"

fileOrURL:  Browse

Start\_Time: 0.0

Value\_1: 1.0

Value\_2: 2.0

Random\_Seed: 123457L

Time\_Distribution: Fixed (Value\_1) ▾

Number\_of\_Transactions: MaxInt

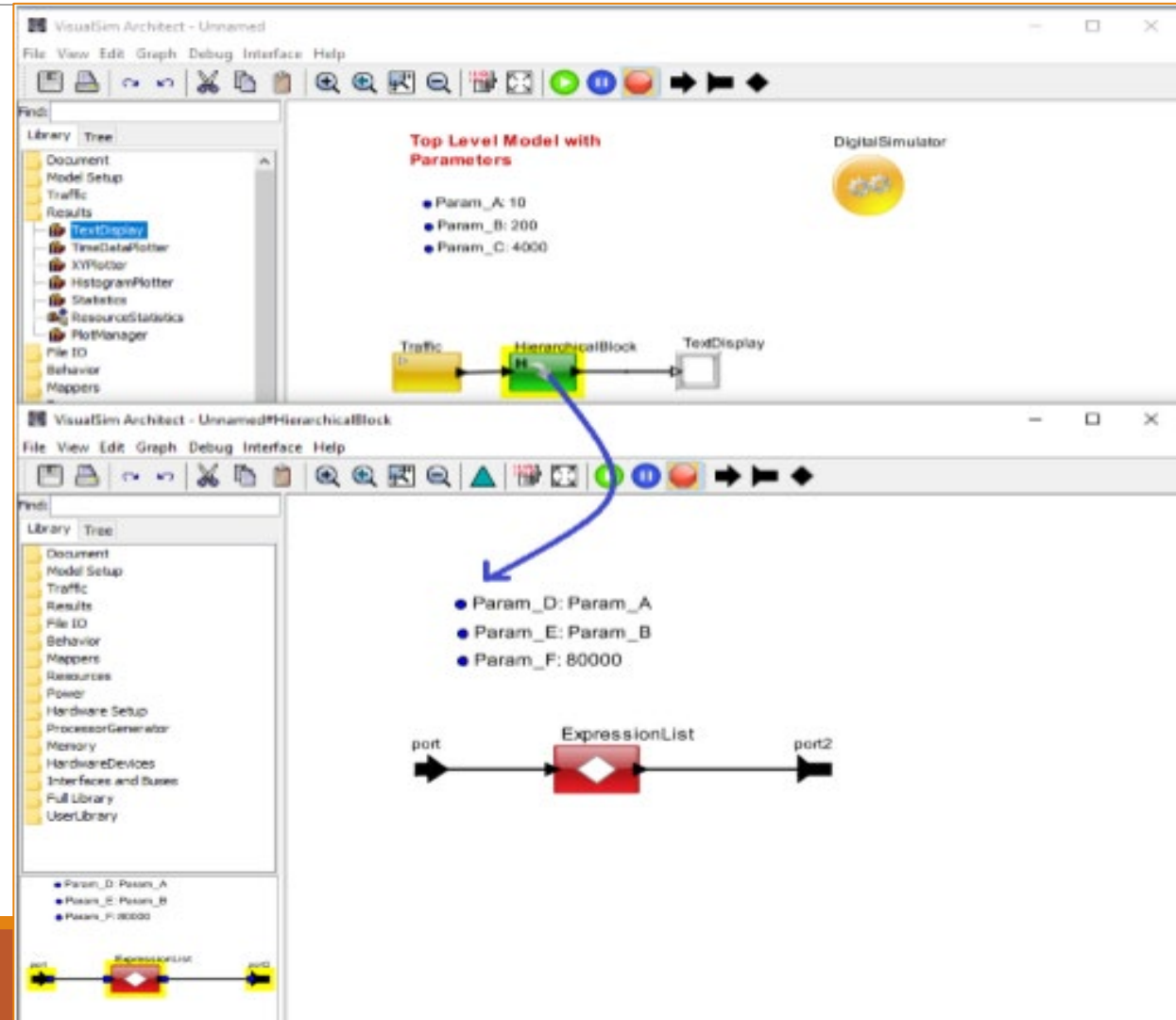
Commit Add Remove Restore Defaults Preferences Help Cancel

Expression\_List: 

```
input.Bytes = irand(500, 1000)
input.Task_Flag = True
```

# Hierarchical Blocks

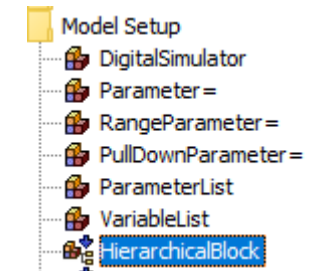
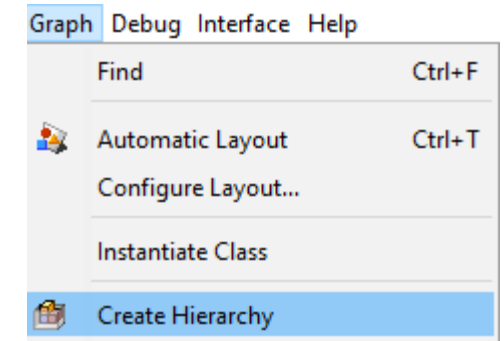
- Grouping a set of functional blocks that combine to define a function or device





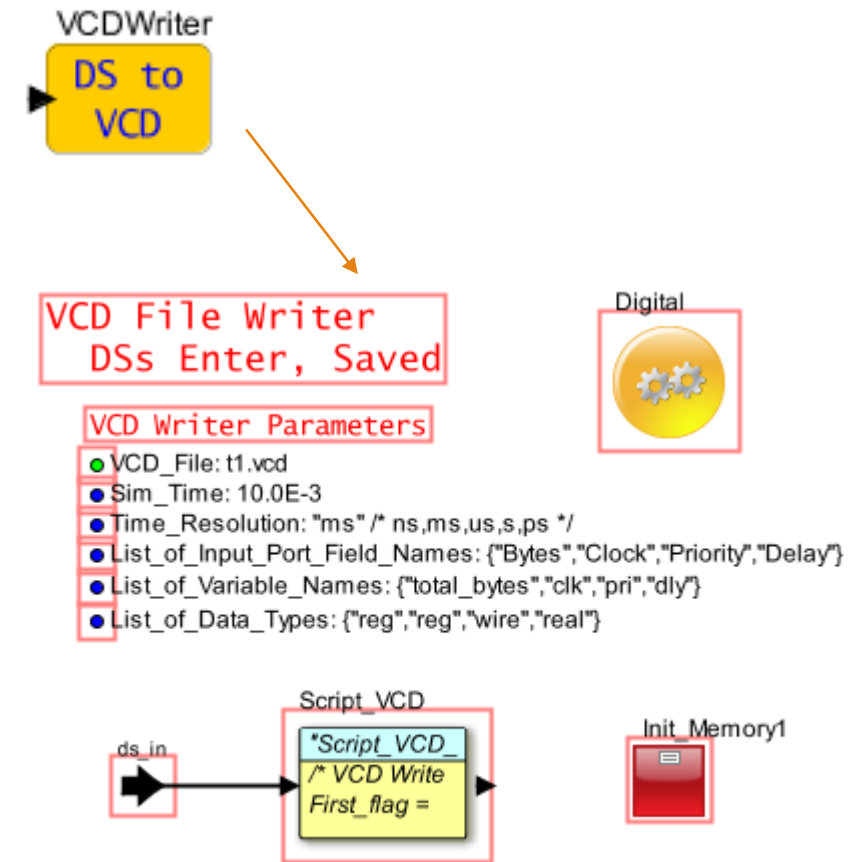
# Construct Hierarchical Block

- Select the blocks to be grouped. Select Edit -> Create Hierarchy
- Drag from Model Setup -> HierarchicalBlock
- Add input and output ports, Parameters
- To view inside the Block, **Right click- > Open Block**



# Class

- A class is a master version of the block.
- Class is an XML sub-model
- Can be instantiated multiple times in the model.
- Changes made to class block are replicated to all linked instances
- All sub-models need a Simulator





# How to construct a Class?

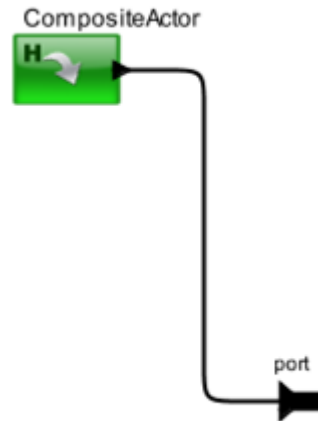
---

- **Assemble the initial block diagram**
  - ✓ Use the library blocks to assemble the model
  - ✓ Create a Hierarchical block of this block diagram
- **Create a class**
  - ✓ Convert the Hierarchical block into a Class
  - ✓ Save as a sub-model
- **Instantiate a new class for use in a model**
  - ✓ To use the Class, you need to instantiate the block in the model using Graph> Instantiate. Entity. Make sure the Class is located within the VS\_Model\_Library directory
- **To test the class**
  - ✓ Test the class by constructing a simple model around it.
- **Save in Library**
  - ✓ Right-click the block and select Save Block in Library.

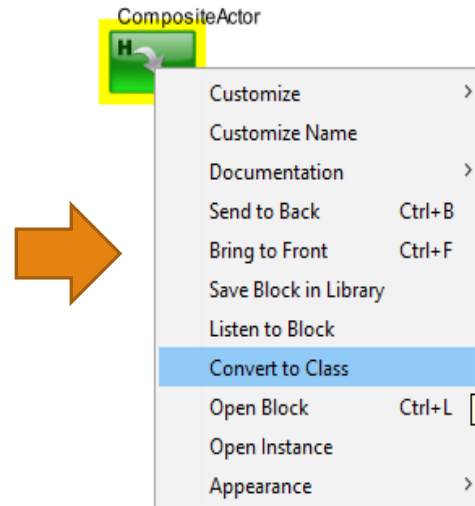


# Constructing a class

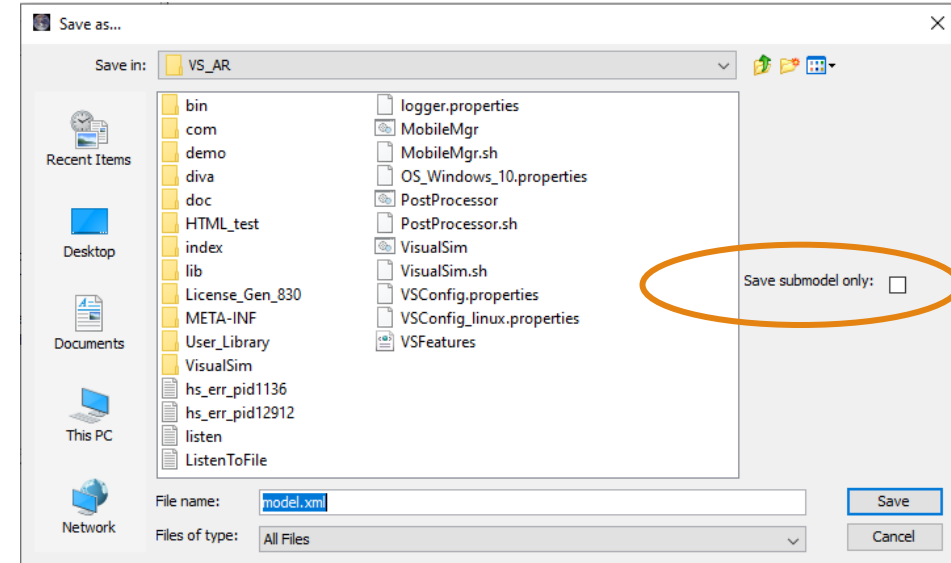
## CREATE HIERARCHY



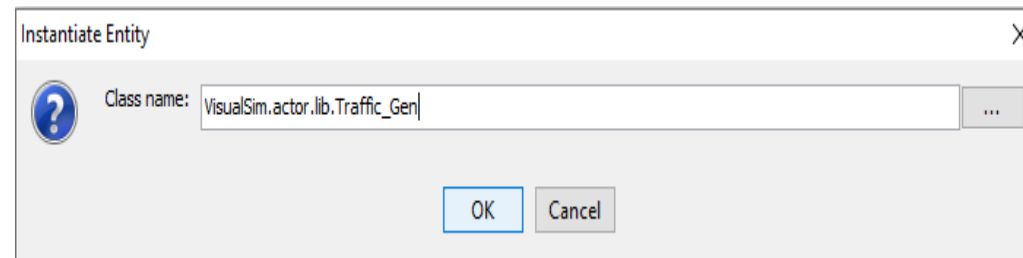
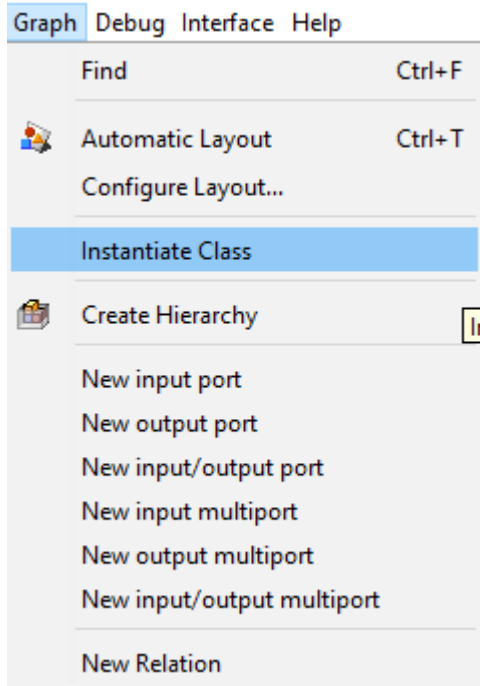
## CONVERT TO CLASS



## OPEN CLASS AND SAVE AS

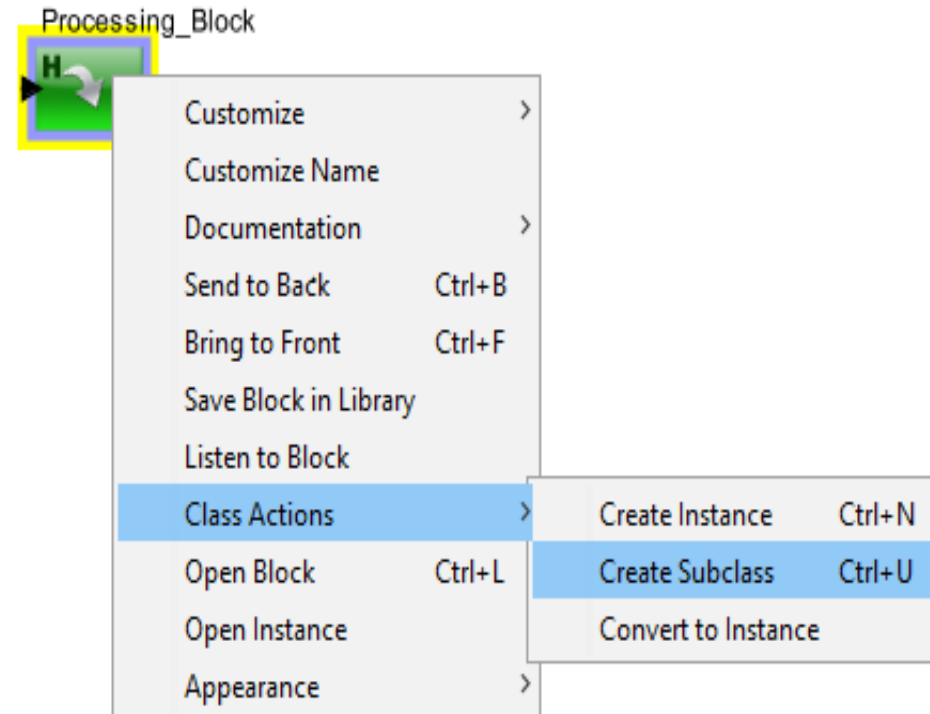


# Instantiating the Class



# Creating Sub Class

- Right click on the Class Block
- Select **Class Actions > Create Subclass**
- Can add additional parameters and Blocks



# Do's and Don't for Class Block

---

Save all class blocks in a common location

Make sure the location is below a Classpath

Organize the Classes in folders under this Classpath

Use the Open Instance for Listen to Block and Listen to Port ONLY

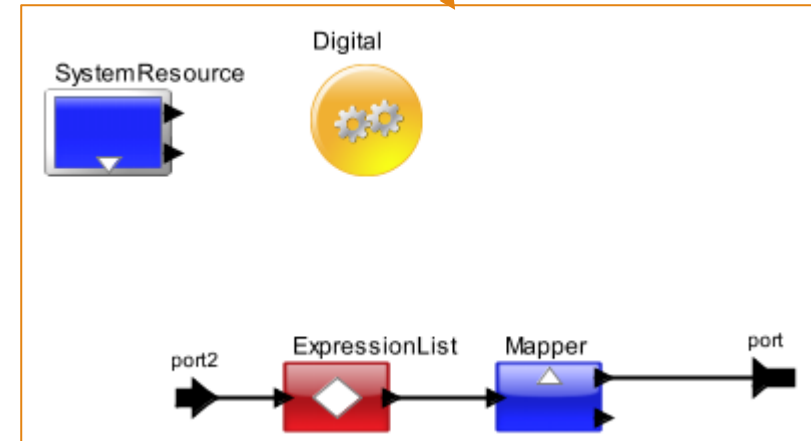
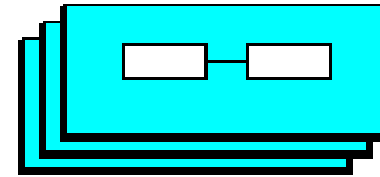
Do not modify any functionality inside the Open Instance.

- This will reflect only in that instance of this Class.
- Any changes to the instance is extremely hard to debug.

# Dynamic Instantiation

- Creates multiple instances of itself during the preinitialize phase of model execution.
- Each instance of this block behaves exactly like a Hierarchical block.
- Helps significantly in building large designs where the model structure scales.

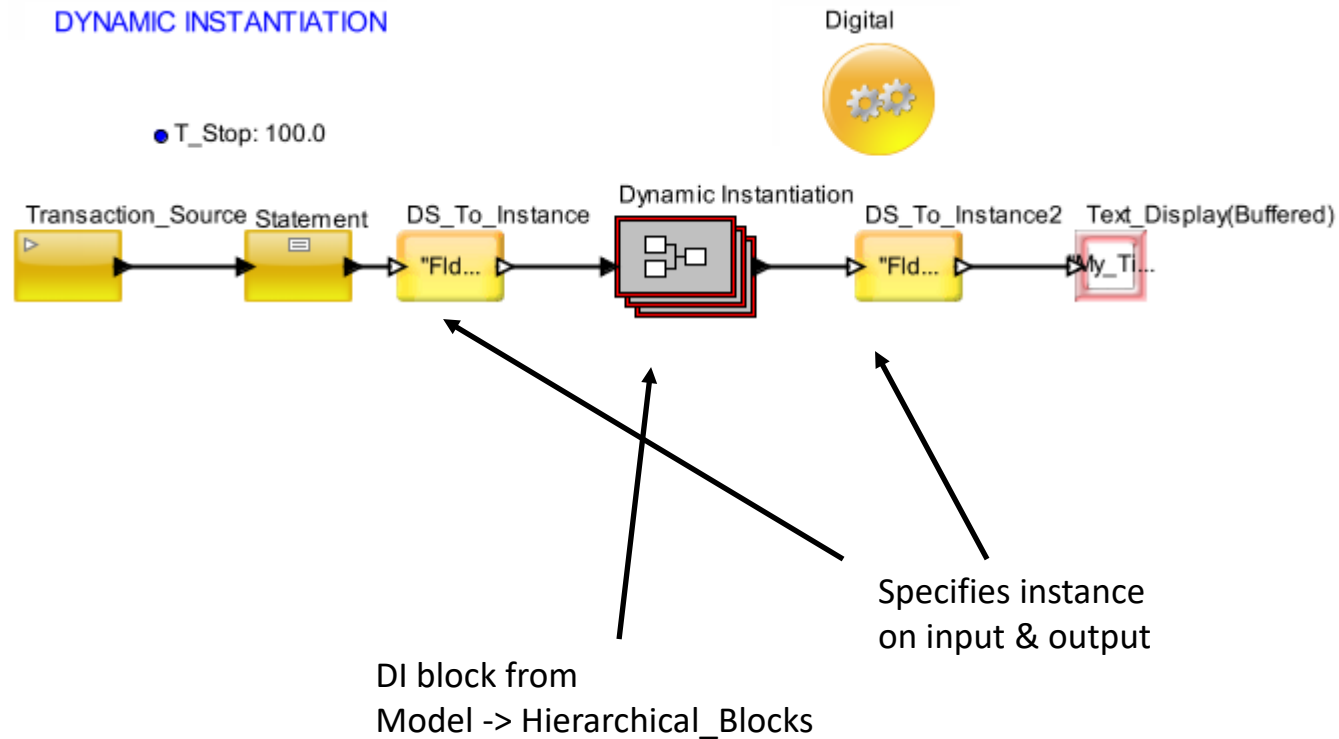
DynamicInstantiation





# Example of Dynamic Instantiation

## DYNAMIC INSTANTIATION



# Advantages of Dynamic Instantiation

- 
- Dynamic Instances solves modeling problems where objects are arbitrarily required.
  - Examples are:
    - ✓ Mobile units entering, traversing and exiting a coverage area
    - ✓ Creation and deletion of virtual circuits
    - ✓ Peer-to-peer protocol connections
  - Above problems are difficult to model using Static Instantiation
    - ✓ Place many copies of a block in a diagram and use block logic to emulate creation and deletion and to branch data to/from the appropriate blocks.
    - ✓ Create a primitive that does all the work.
  - Static Instantiation requires over-specification and makes model engineering and presentation difficult.



# Differences

---

- Modification in one Hierarchical block does not affect another
- Changes in Open Block of Class block will be reflected at all places were used
- Changes in Instance will only be reflected in that one location
- Open Block for Dynamic Instantiation and Hierarchical Block for Listeners
- Open Instance for Classes for Listeners

# VisualSim Simulation, Diagnostics and Recommendation Engine



# Simulation

---

## GUI

- Modify parameters in the top-window
- Click on GO button
- View results in real-time

## User-Created Batch simulation

- `C:\VisualSim\Java\jdk-14.0.1\bin\java -classpath C:/VisualSim/VS64 -Xms1024m -Xmx8048m VisualSim.actor.gui.VisualSimBatchModeSimulator -resultpath C:/VisualSim/VS64/User_Library C:/VisualSim/CModel/User_Library/JPL/FPGA/Multi_Core_Demo_Model_32_3.xml`
- Path to Java folder
- Set the `-classpath {VisualSim Install Directory}`
- Set the minimum and maximum memory allocation
- Define the executable `VisualSim.actor.gui.MoMLSimpleStatisticalApplication`
- `-resultpath {with full path and filename}`
- List of parameters to modify – `Top_level_parameter_name value – Hier_name.Block_Name.Parameter_name value`
- Model name: `{Path + file_name}`

# Setting up the Simulation

---

Add PlotManager to the model

Create CSV file

- 2.1) Value CSV
- 2.2) Path CSV

Create index.xml and batch file using PostProcessor

Simulation

- Run the batch simulation sequentially
- Run the batch simulation on multi core

View plot results using PostProcessor



# Post Processor

---

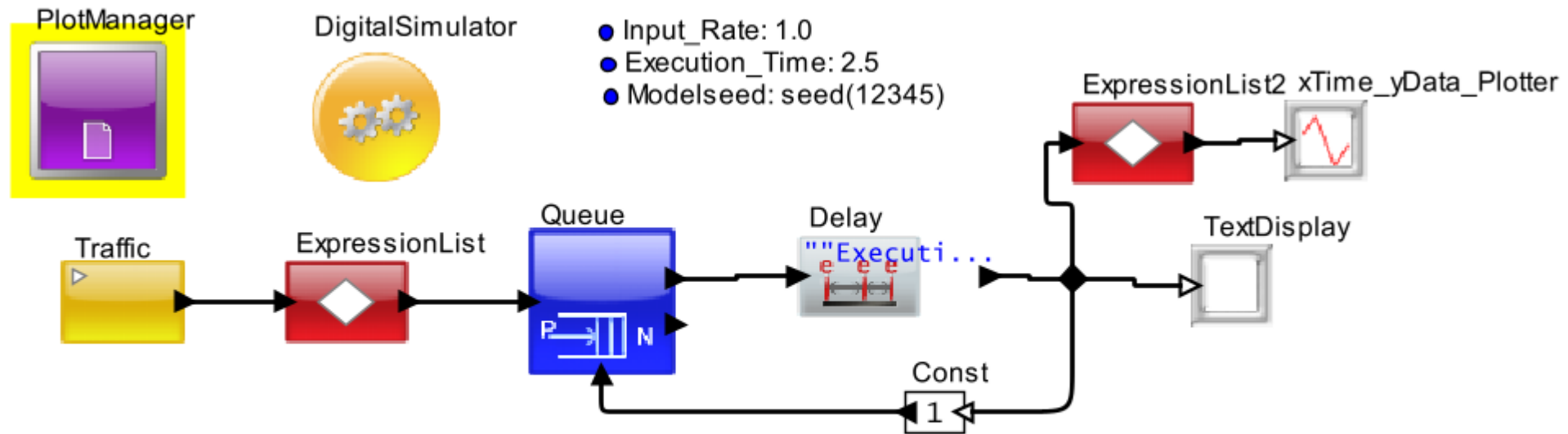
- Enables off-line viewing of results
- Analyze performance data collected from the simulation
- VisualSim Post Processor works with the Plotter, TextDisplays and the HistogramPlotter blocks of the VisualSim Block Diagram Editor.
- Operating Semantics
  - ✓ Step 1: From the VisualSim Architect Block Diagram Editor, add the PlotManager block to the model
  - ✓ Step 2: From the Post Processor window, select Activity->Configure Model. To create the model index
  - ✓ Step 3: Now run the simulation in batch or interactive mode by varying the selected parameters
  - ✓ Step 4: Now return to the Post Processor. From Activity->Open Plot Index, select the list of runs and the plot to view

# **1. Add PlotManager to block diagram**



# Steps

1. Add the Plot Manager to the model from the library. Results->PlotManager

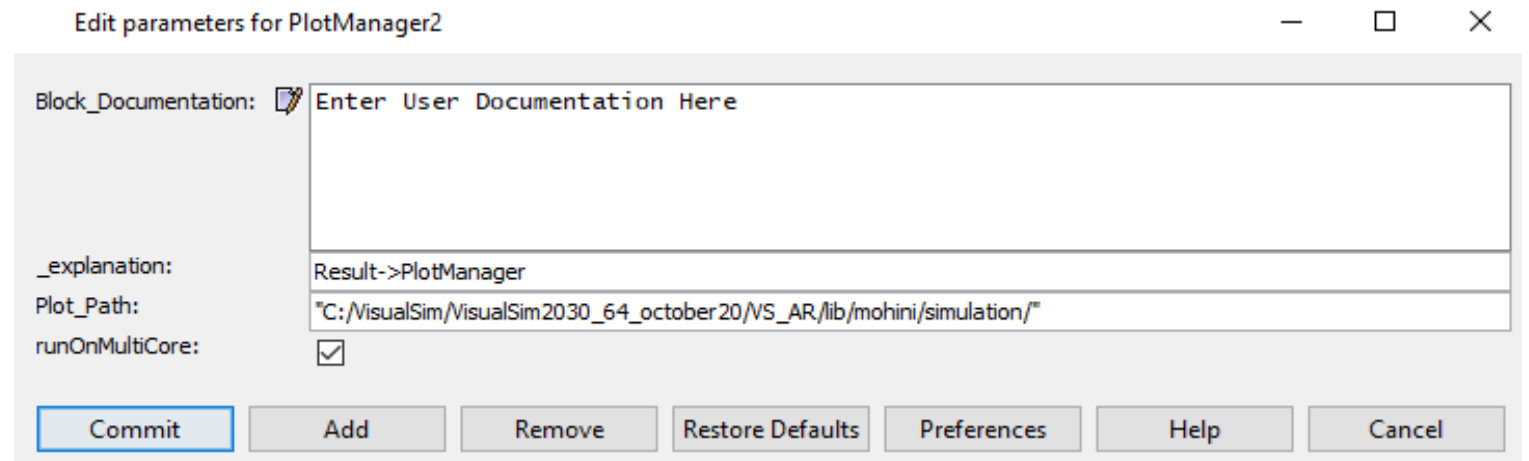


Make sure you change the Path in the PlotManager before running this model.

# Steps

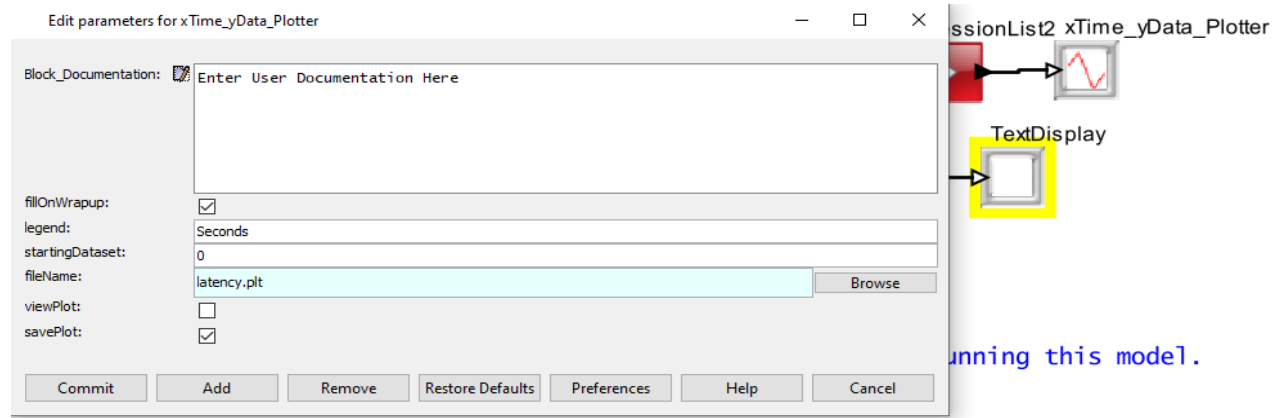
2. The PlotManager parameter Plot\_Path represents the location where the plots will be stored, so make sure that this path is valid for your environment. If you copy the location from the address bar, the address will be specified using "\". Change "\" to "/".
3. For example – if the model is saved in the directory –  
`C:\VisualSim\VisualSim2030_64_grapes_beta\VS_AR\lib\mohini\simulation\part_index_model`
4. Then the Plot\_Path is– `"C:/VisualSim/VisualSim2030_64_grapes_beta/VS_AR/lib/mohini/simulation/part_index_model/"`
5. **To run the simulation on multiple cores, check the parameter runOnMultiCore , else uncheck it .**

**Note :** backslash “\”changes to forwardslash “/”and the path has to be within double quotes.



# Steps

6. Batch simulations cannot run with graphical viewers. Therefore, all Text\_Display and Plotters must be configured to be in "Save" mode. This can be done either using the block parameters or by using the Post Processor.
7. Double Click on the TimeDataPlotter and Select SavePlot and provide a file name with extension .plt



running this model.

8. Save the model and close it.



## 2. Create CSV files

---

# 2.1 Create *Value CSV* file

## Steps

---

1. Create the csv file and save the file where the model file(.xml) is saved.
2. Give the parameter name and values in (Range column )in the csv file as shown

	A	B	C
1	Parameter	Range	Step
2	Input_Rate	{1;3}	1
3	Execution_Time	{2;8}	2

1. The parameter name should match with the model parameter name.

# Steps

4. Csv column explanation –
  1. Parameter -> contains the name of the top level parameter from the model.
  2. Range-> The upper and lower integer value of the parameter.
  3. Step -> Increase from lowest value for each run to maximum value.

**For example –**

Input\_Rate range is from {1 to 3} and it will increment(Step) by 1 – so we will get 1,2,3.

Same with Execution\_Time, Execution\_time range is from {2 to 8} and Increment by 2 each time, so we will get 2,4,6,8.

Now, Input\_Rate= {1,2,3} and Execution\_Time={2,4,6,8}

so possible combination is -

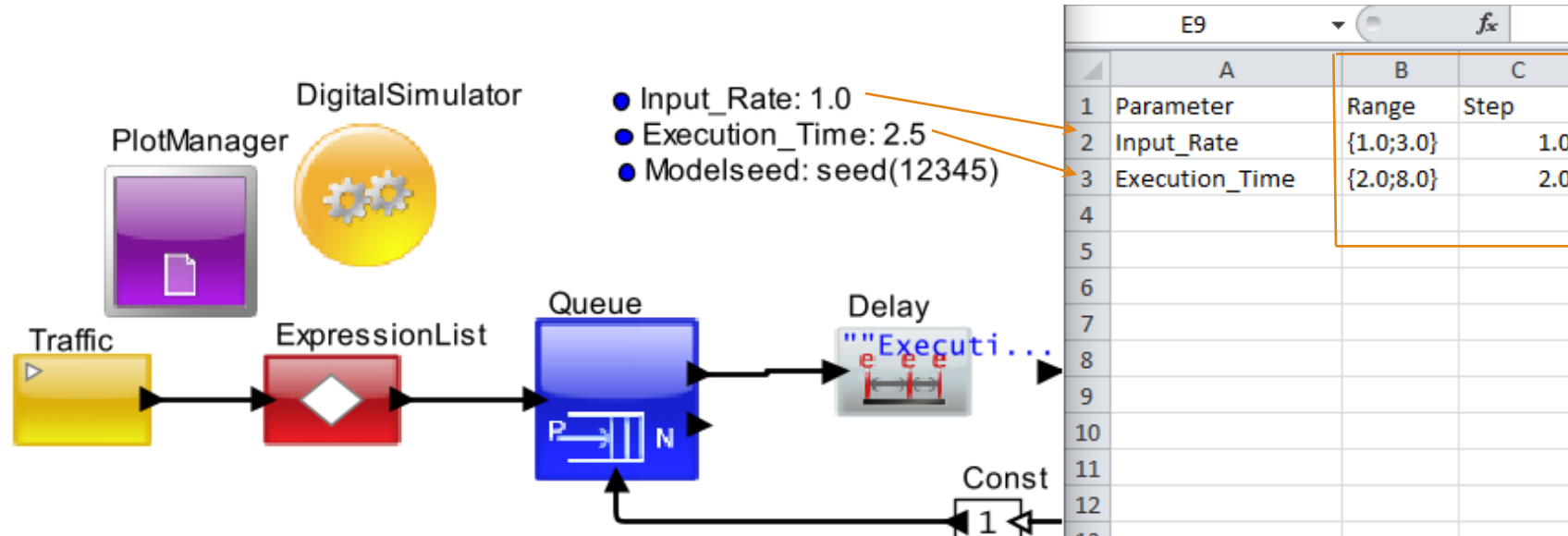
(1,2), (1,4), (1,6), (1,8), (2,2), (2,4), (2,6), (2,8), (3,2), (3,4), (3,6), (3,8)

Total of 12 combination.

	A	B	C
1	Parameter	Range	Step
2	Input_Rate	{1;3}	1
3	Execution_Time	{2;8}	2

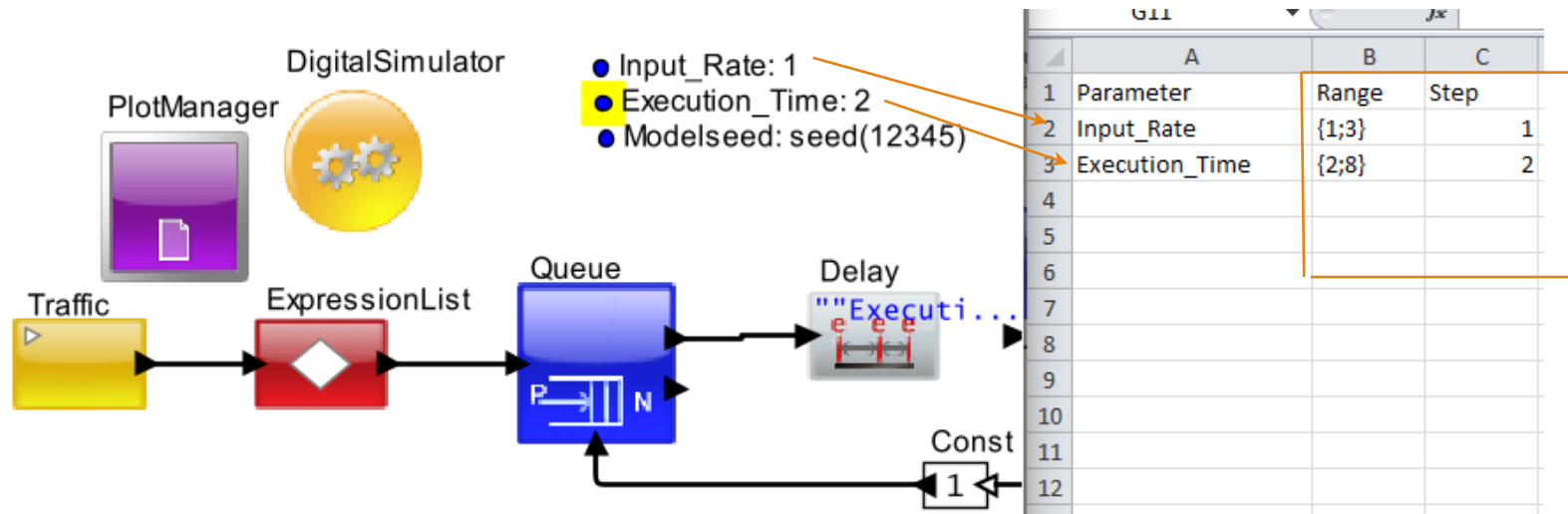
# Steps

5. If the values in the model parameter are in decimal, provide the decimal values in the csv column – Range and Step.



# Steps

- If the values in the model parameter are in integer, provide the integer values in the csv column – Range and Step.



- Save the file and close it.

**Note:** The format provided in the sample csv for range column should be same as the file you create.



## 2.2 Create *Path CSV* file

### Steps

---

1. Create the csv file and save the file where the model file(.xml) is saved.
2. Provide the Javapath- (Java location in the system) and classpath - VS\_AR (must),along with any other path (seperated by semicolon) as shown -

	A	B
1	javapath	classpath
2	C:\Program Files\Java\jdk-14	C:\VisualSim\VisualSim2030_64_october20\VS_AR
3		

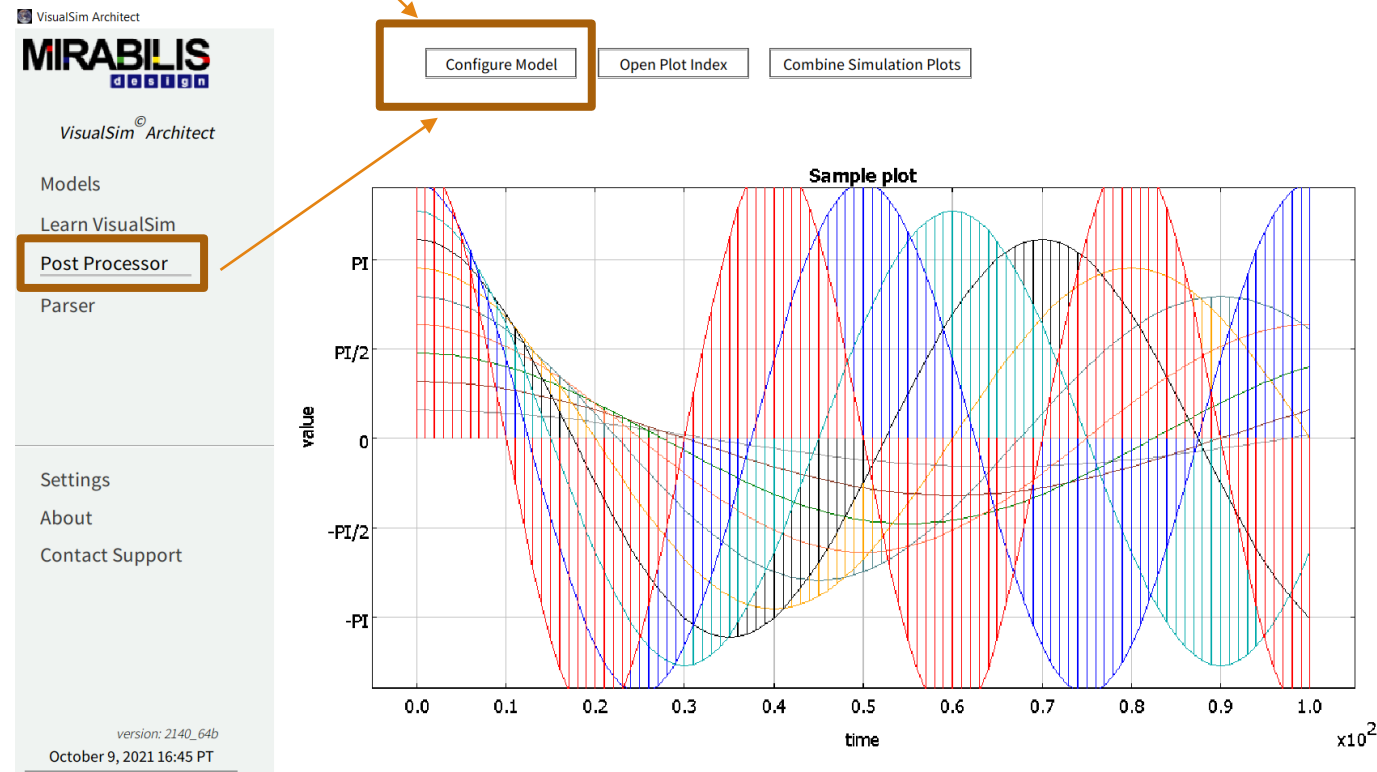
3. Save the file.

### 3. Create index.xml and batch file using Post Processor

---

# Steps

1. Click Activity->Configure Model and select the model with the PlotManager (make sure model is saved).



# Steps


---

2. There will be a mixture of Text\_Displays and Plotters. First, select parameters that we will vary during this simulation. Click on Save for all Plots and Displays that will needed to be saved. Do not select the View for any item. This will ensure that no real-time displays are available when the Batch-mode simulation is executed. Add a file name for each item that needs to be saved. All plotter views must have a file name with a .plt extension or you will get an error. All Text\_Displays must have a file name with a .txt extension or you will get an error.
3. Select the **Value csv** file you have created in (2.1 Topic) with parameter names, ranges and steps.
4. Select *the Path csv* file you have created in (2.2 Topic) with javapath and classpath.
5. Click “Apply” and “Ok”
6. Refer the image in the next slide


Step 2

Configure Model    Open Plot Index    Combine Simulation Plots

**Postprocessor Configuration**

Parameters 

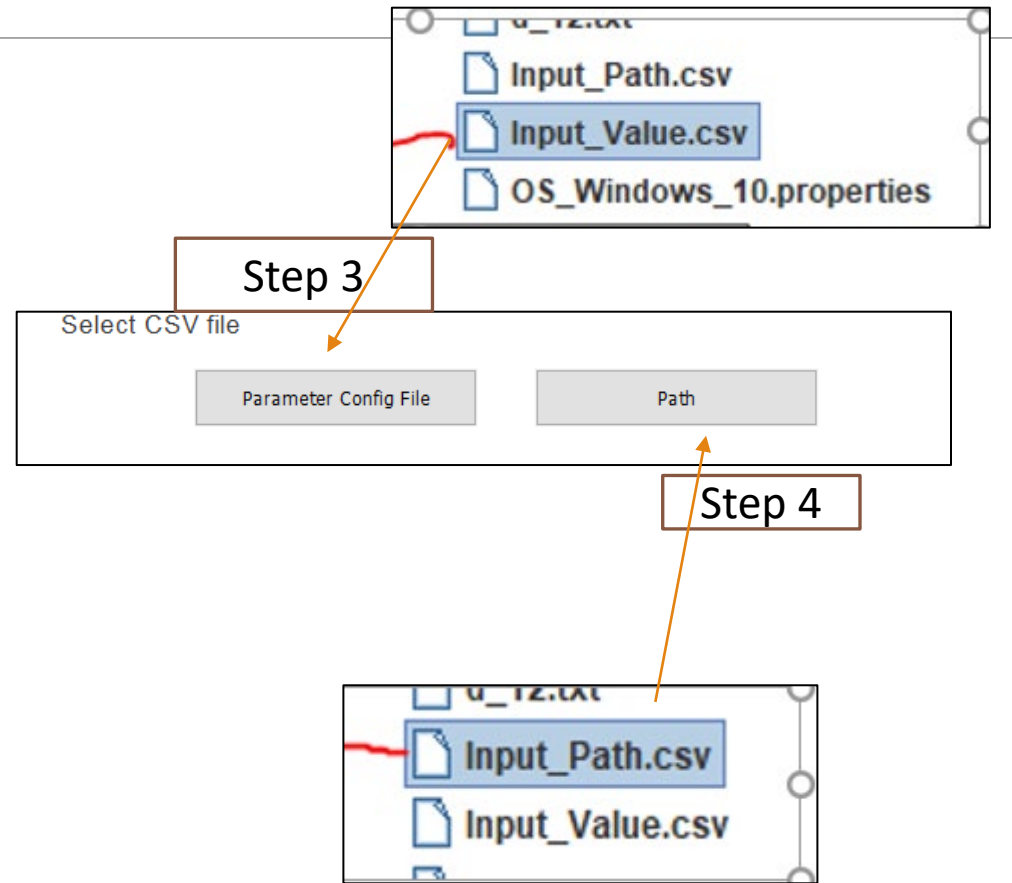
<input checked="" type="checkbox"/> Sim_Time	<input checked="" type="checkbox"/> Scheduler_Type
<input checked="" type="checkbox"/> Start_Time	<input checked="" type="checkbox"/> File_Name

Plots 

Select CSV file

Parameter Config File    Path

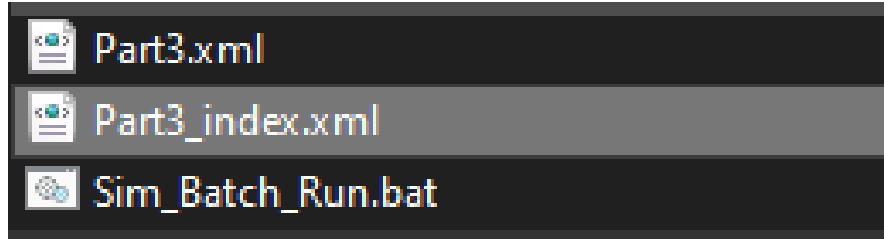
Ok    Apply    Cancel



# Steps

---

7. A file called *ModelName\_index.xml* and *Sim\_Batch\_Run* file will be created in the same directory as the model.
8. For example – if the model name is Part3.xml , then the Part3\_index.xml file will be created.





## 4. Run the batch simulation

---

# Steps

---

1. **To run**, click on the Sim\_Batch\_Run.bat / Sim\_Batch\_Run.sh file
2. This run the simulation from command line.
3. This will create the .plt and .txt file in the same folder as the model.





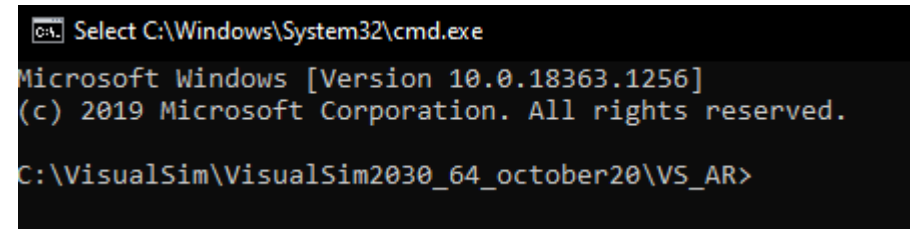
## 5. Run the batch simulation on multi core

---

# Steps

1. To run on different CPU core –

I. Open the cmd terminal and navigate to VS\_AR folder



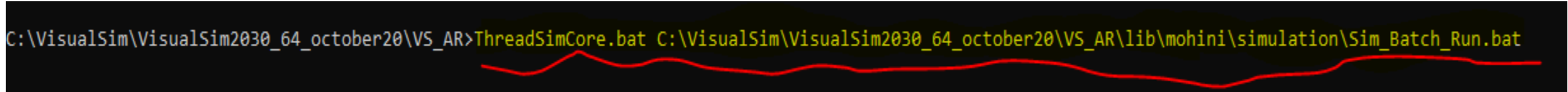
```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\VisualSim\VisualSim2030_64_october20\VS_AR>
  
```

*space*

I. To execute, type ThreadSimCore.bat *space* path of/Sim\_Batch\_Run.bat

II. For example : ThreadSimCore.bat C:\VisualSim\VisualSim2030\_64\_october20\VS\_AR\lib\mohini\simulation\Sim\_Batch\_Run.bat



```

C:\VisualSim\VisualSim2030_64_october20\VS_AR>ThreadSimCore.bat C:\VisualSim\VisualSim2030_64_october20\VS_AR\lib\mohini\simulation\Sim_Batch_Run.bat
  
```

I. Press enter

II. This will run the simulation using command line on the different CPU cores.

III. This will create the .plt and .txt file in the same folder as the model.

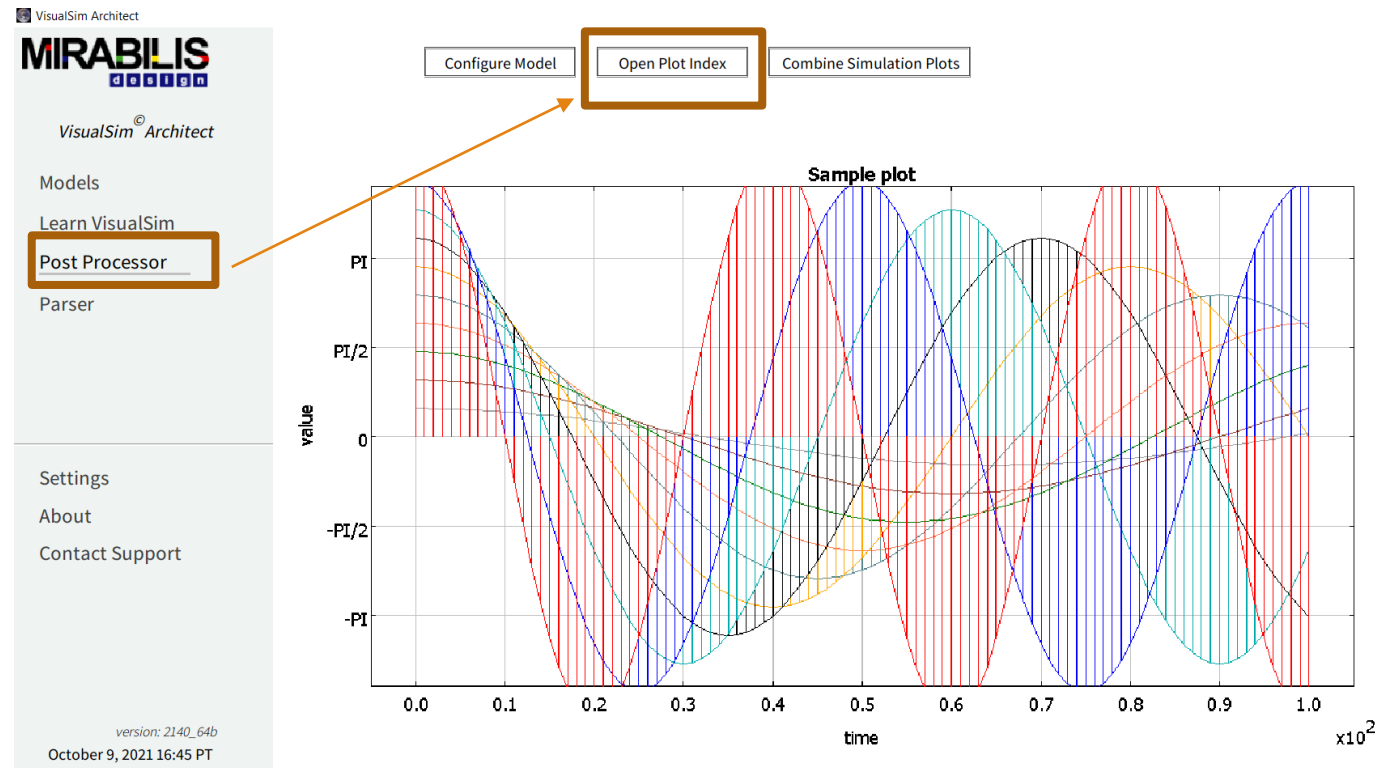


## 6. View plot results using Post Processor

---

# Steps

After following the above steps , click on the Postprocessor and go to Activity> Open Plot Index

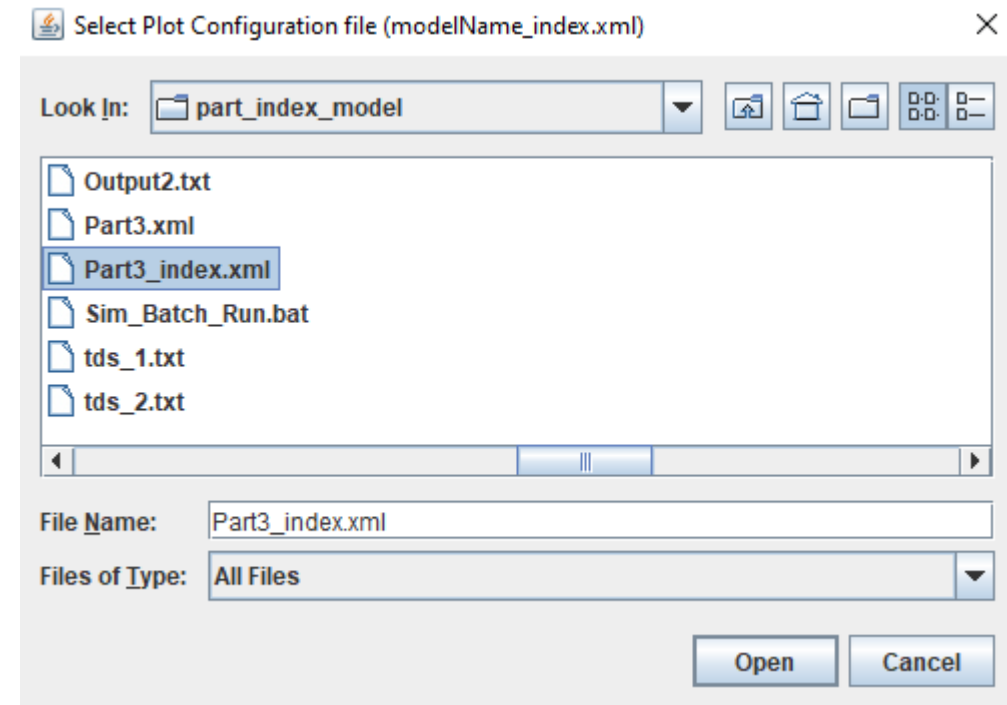


# Steps

---

Go to the folder which contain the model and where the index file is also created.

Select the index file (index.xml) . Click open.



# Steps

Top-half displays list of simulations

Bottom half shows list of available plots.

The user may do the following:

- Select a plot name
- Select a list of simulations
- Select one among three options below:

**Model Plots**

**Select Parameters**

Start\_Time=0,File\_Name="/lookup\_1.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"  
 Start\_Time=0,File\_Name="/lookup\_2.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"  
 Start\_Time=0,File\_Name="/none.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"  
 Start\_Time=3,File\_Name="/lookup\_1.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"  
 Start\_Time=3,File\_Name="/lookup\_2.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"  
 Start\_Time=3,File\_Name="/none.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"

---

**Select Plot**

Batch\_Latency  
TextDisplay

# Steps

6. Select traces and view the plots

**Model Plots**

**Select Parameters**

- Start\_Time=0,File\_Name="/lookup\_1.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"
- Start\_Time=0,File\_Name="/lookup\_2.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"
- Start\_Time=0,File\_Name="/none.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"
- Start\_Time=3,File\_Name="/lookup\_1.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"
- Start\_Time=3,File\_Name="/lookup\_2.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"
- Start\_Time=3,File\_Name="/none.csv",Sim\_Time=2,Scheduler\_Type="Batch\_Mode"

**Select Plot**

Batch\_Latency

TextDisplay

**xTime\_yData\_Plotter**

The graph displays seven data series: Seconds\_1 (red), Seconds\_2 (blue), Seconds\_3 (cyan), Seconds\_4 (black), Seconds\_5 (orange), Seconds\_6 (green), and Seconds\_7 (purple). The x-axis represents time from 2 to 10, and the y-axis represents values from 0 to 6. The series show varying trends, with some peaking around time 8 and others remaining relatively flat or slightly decreasing.

View Plot
View Selected Traces
View All Traces