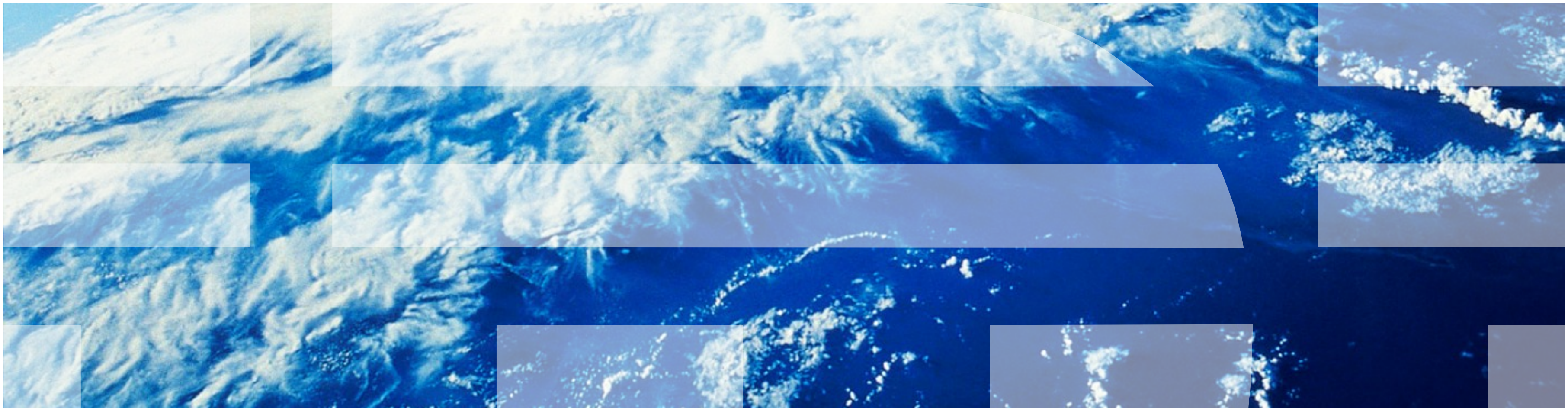
SAKAIRI Takashi (IBM Reseach – Tokyo), Eldad Palachi (Rational), Chaim Cohen (Rational)

September 25, 2012 (January 8, 2014, updated page 136)

Rhapsody V8.0
Control System Designer Tutorial



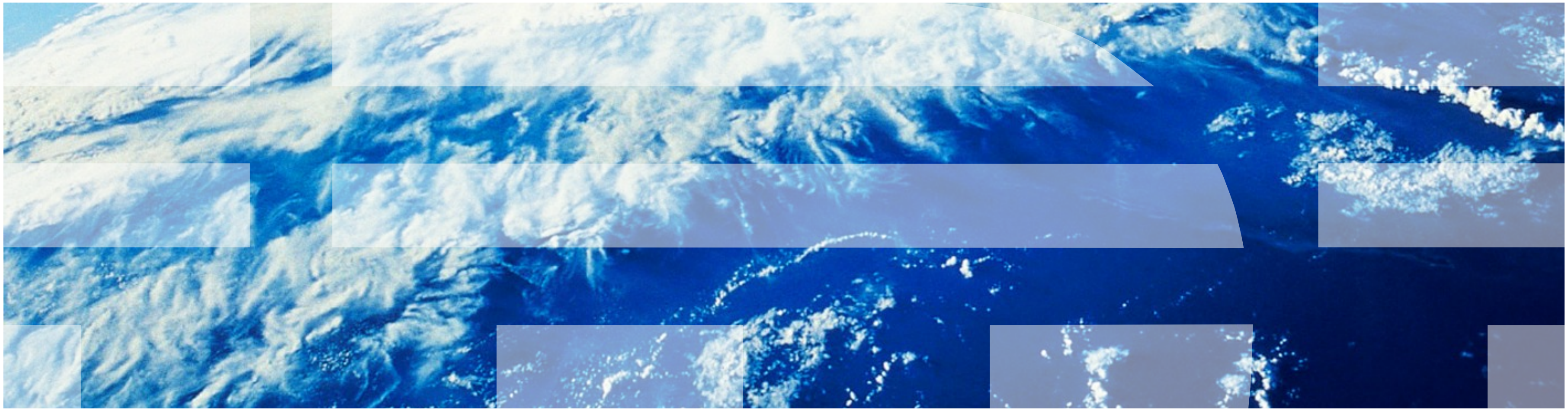
Purpose of this tutorial

- This tutorial shows basic usages of Control System Designer introduced in IBM® Rational Rhapsody V8.0. Control System Designer is an enhanced version of Plant Model Integration.
- Plant Model Integration in Rhapsody V7.6 has the following capabilities:
  - Exports a Simulink model from an Internal Block Diagram (IBD).
  - Generates an S-Function from statecharts.
- Control System Designer in Rhapsody V8.0 has the following enhancements:
  - Exports a skeleton Simulink model from a <<SimulinkBlock>> block.
  - Supports Simulink bus objects and enumerations.
  - Improves type mapping between SysML and Simulink.
  - Synchronizes SysML attributes with MATLAB variables and model arguments.
  - Supports new Simulink file format (.SLX).
  - Simulates multiple times without re-exporting Simulink models.
  - Defines signals to be plotted in a SysML model.
  - Invokes only one MATLAB process.
  - Improves S-Function's animation.
  - Improves Panel Diagram's usability.

Table of contents

- Install
- Lesson 1: Using two simple Simulink models
- Lesson 2: Exporting templates of Simulink sub-models
- Lesson 3: Using a bus object and an enumeration
- Lesson 4: Exporting a three layer model
- Lesson 5: Generating an S-Function
- Lesson 6: Cruise Control system

Install



Required software

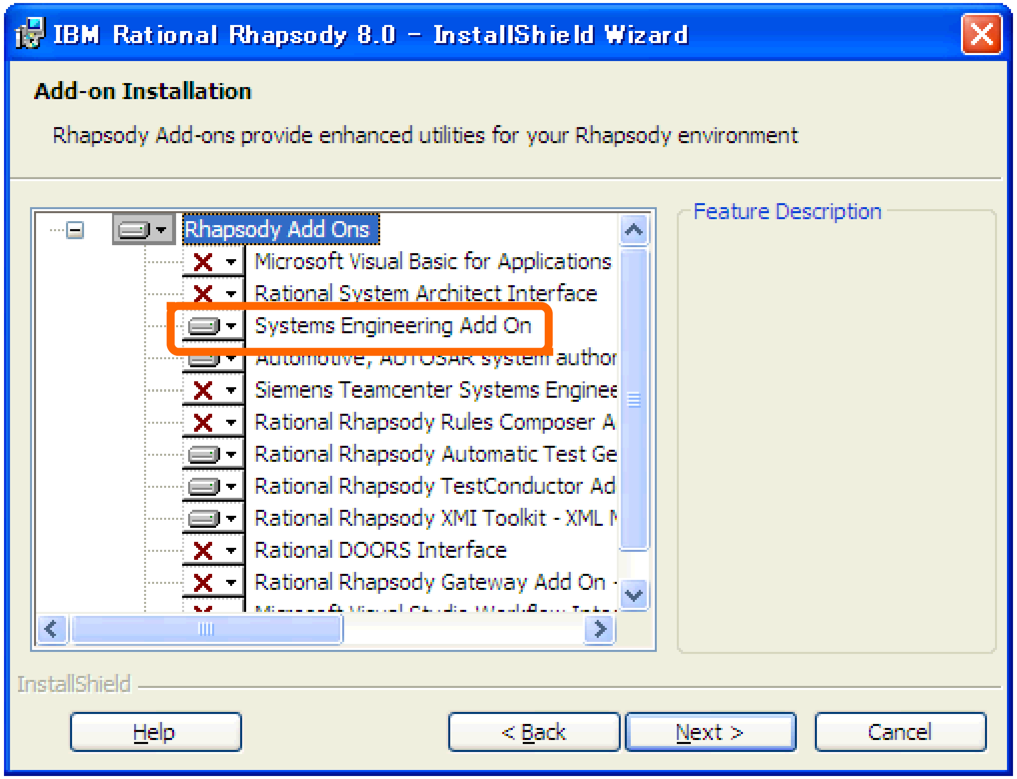
- Operating system
  - Microsoft Windows
  - Note: We cannot use Control System Designer in Linux.
- IBM Rational Rhapsody
  - Rhapsody V8.0 or later.
    - We have to use Rhapsody Developer for C++ or Rhapsody Developer for C to generate an S-Function without animation.
    - We can use Rhapsody Designer for Systems Engineers to generate an S-Function with animation.
- MATLAB/Simulink
  - MATLAB V6 (?) or later.
  - Simulink V7 (?) or later.
  - Note: Control System Designer does not use Real-Time Workshop.
- C/C++ compiler
  - We need a C/C++ compiler to generate an S-Function.
  - Refer the next page.

Selecting a C/C++ compiler and install it

- We need a C/C++ compiler to generate an S-Function.
- Select a C/C++ compiler that are supported both by Rhapsody and by Simulink (compiling an S-Function), and install it.
- Rhapsody V8.0 supports the following C/C++ compilers:
  - Microsoft Visual Studio 2010
  - Microsoft Visual Studio 2008
  - etc.
- Check C/C++ compilers supported by your Simulink version for compiling an S-Function.
  - Note: Each Simulink version supports different C/C++ compilers.
    - For example, “Microsoft Visual C++ 2008 Express Edition and Windows SDK 6.1” are supported by Simulink R2010b, but not supported by Simulink R2011a.
- If you want to use 64-bit Simulink, you have to install a 64-bit C/C++ compiler.

Installing Rhapsody V8.0

- If you do not have Rhapsody V8.0, download it.
  - http://www.ibm.com/developerworks/downloads/r/rhapsodydeveloper/
- Refer the following document, and install Rhapsody V8.0.
  - http://pic.dhe.ibm.com/infocenter/rhaphlp/v8/index.jsp?nav=%2F5
- Note: Install Systems Engineering Add On.



IBM Rational Rhapsody 8.0 - InstallShield Wizard

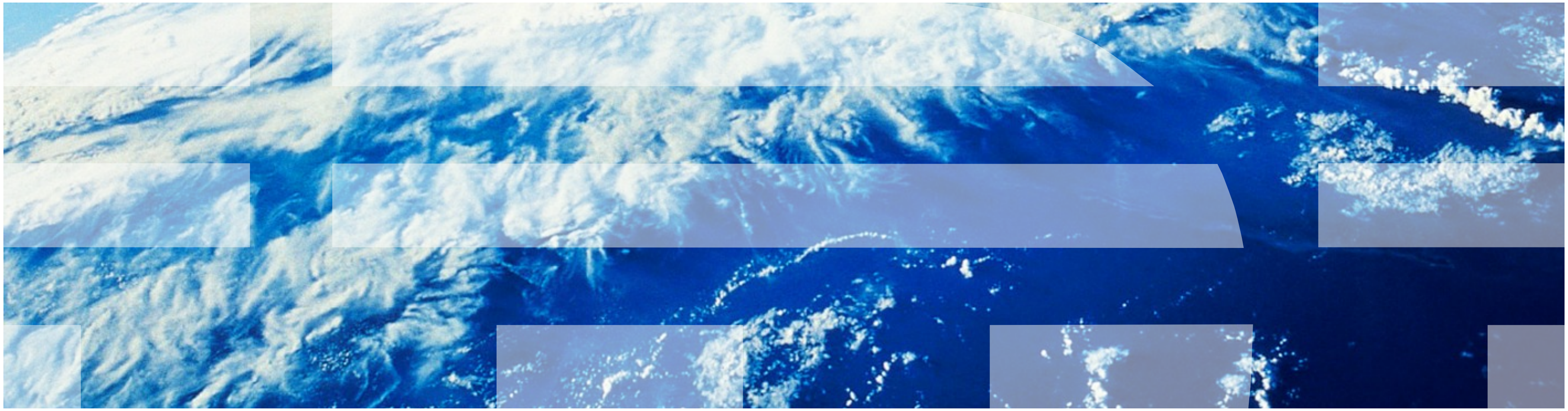
Add-on Installation

Rhapsody Add-ons provide enhanced utilities for your Rhapsody environment

- ▭ Rhapsody Add Ons
  - ✗ Microsoft Visual Basic for Applications
  - ✗ Rational System Architect Interface
  - ▭ Systems Engineering Add On
  - ▭ Automotive, AUTOSAR system author
  - ✗ Siemens Teamcenter Systems Enginee
  - ✗ Rational Rhapsody Rules Composer A
  - ▭ Rational Rhapsody Automatic Test Ge
  - ▭ Rational Rhapsody TestConductor Ad
  - ▭ Rational Rhapsody XMI Toolkit - XML N
  - ✗ Rational DOORS Interface
  - ✗ Rational Rhapsody Gateway Add On -
  - ✗ Microsoft Visual Studio Workflow Inte

Feature Description

Lesson 1: Using two simple Simulink models

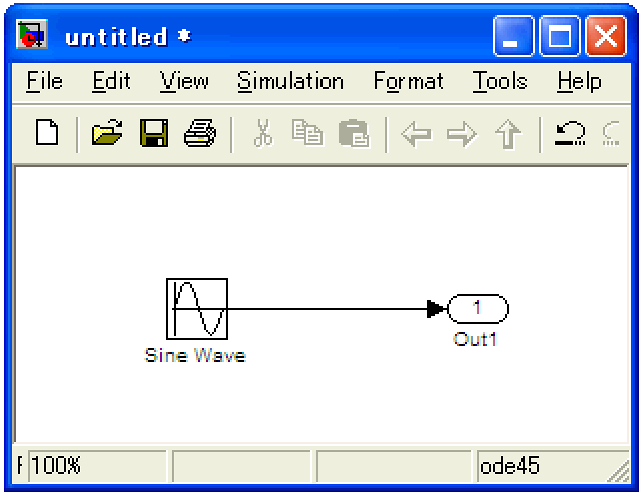


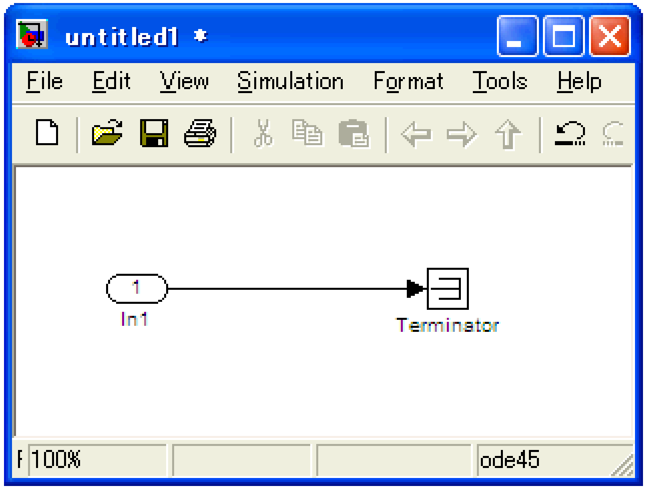
Contents of this lesson

- Develop two simple Simulink sub-models.
- Create a new Rhapsody project.
- Add the Simulink profile to the model.
- Define a relation between two Simulink sub-models in Rhapsody.
- Export a Simulink model that refers two Simulink sub-models from IBD.
- Execute simulation.
- Change configuration parameters.

Creating sub-models (1/2)

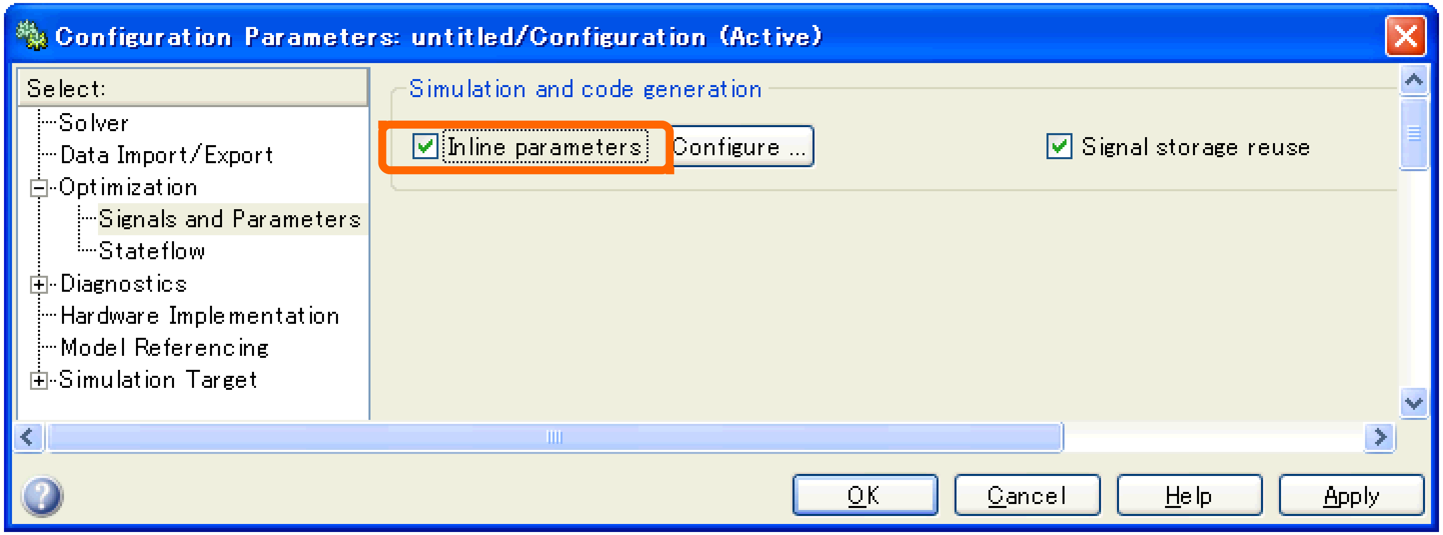
- Create two simple sub-models using MATLAB/Simulink.





Creating sub-models (2/2)

- Check “Inline parameters” in “Configuration Parameters” for each model.
- Save the created models.
  - Create a new folder “C:\Work\Lesson1\Simulink” to save the Simulink models.
  - Name the left model of the previous page MySource.mdl, and the right model MySink.mdl.



Configuration Parameters: untitled/Configuration (Active)

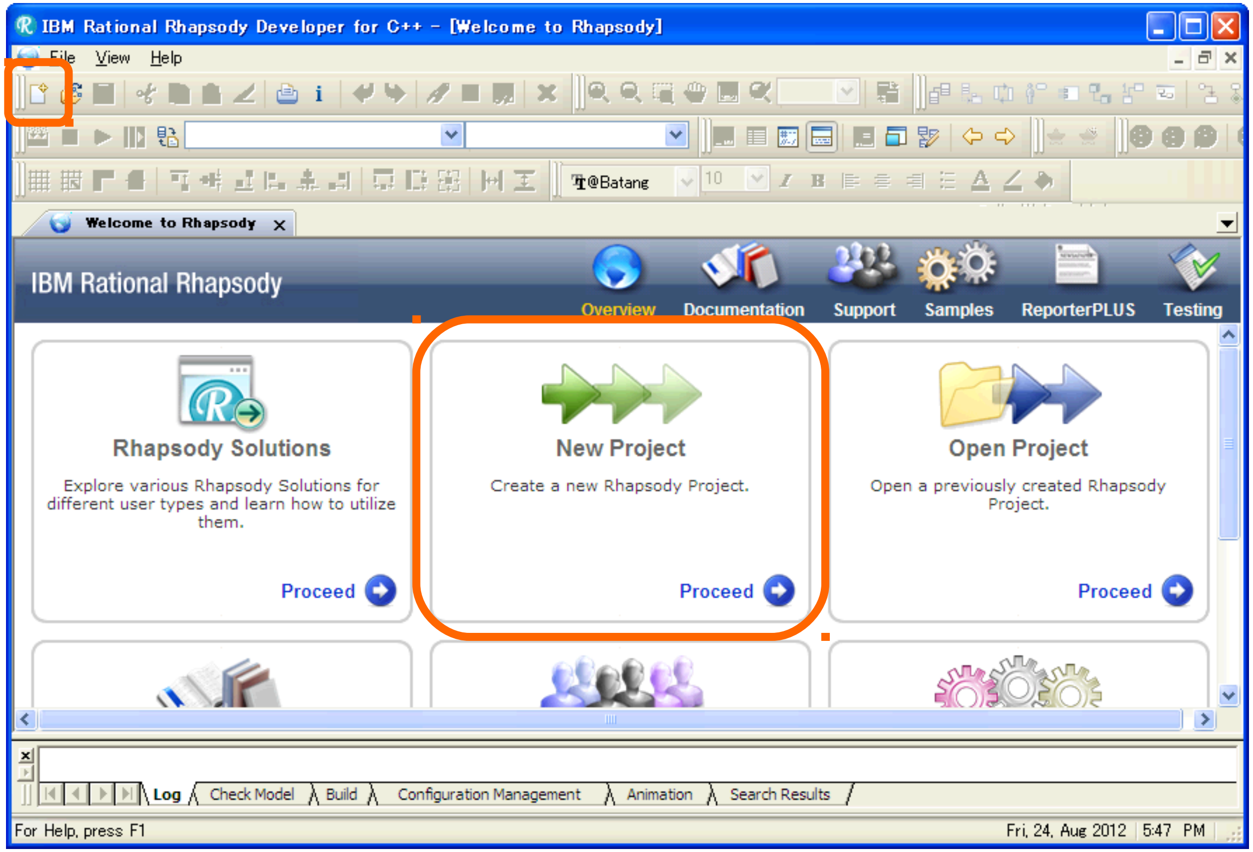
Simulation and code generation

☑ Inline parameters

☑ Signal storage reuse

Creating a new project

- Start Rhapsody.
- Perform one of the following operations:
  - Click “New Project” from the “Welcome to Rhapsody”.
  - Select “File”/”New” from the menu.
  - Click the “New” icon.
  - Press “Ctrl” key and “N” key simultaneously.



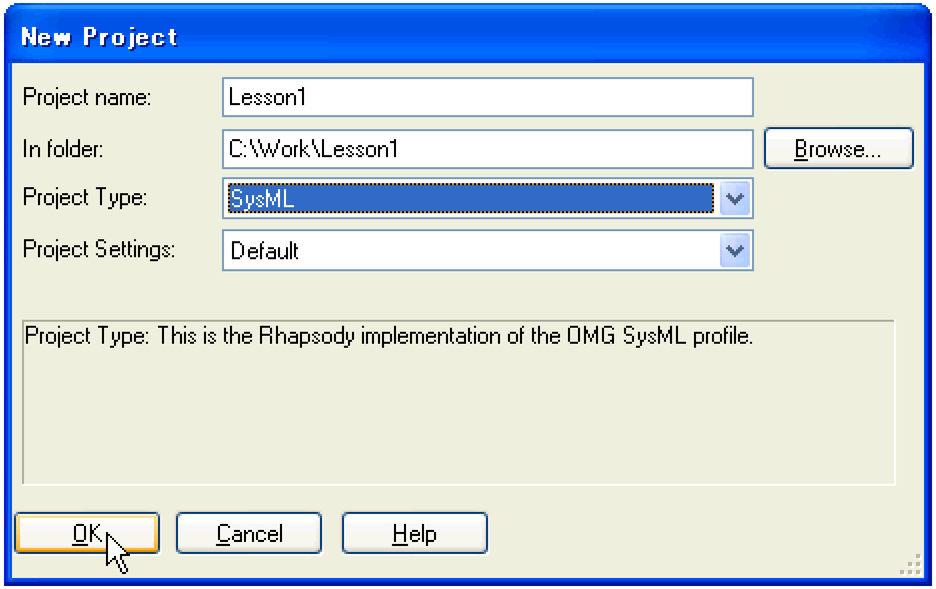
Explore various Rhapsody Solutions for different user types and learn how to utilize them.

Create a new Rhapsody Project.

Open a previously created Rhapsody Project.

“New Project” dialog

- Click “Browse…” button, and refer the C:\Work folder.
- Name the project Lesson1.
- Select SysML as the Project Type.
- Click “OK” button.



New Project

Project name: Lesson1

In folder: C:\Work\Lesson1

Project Type: SysML

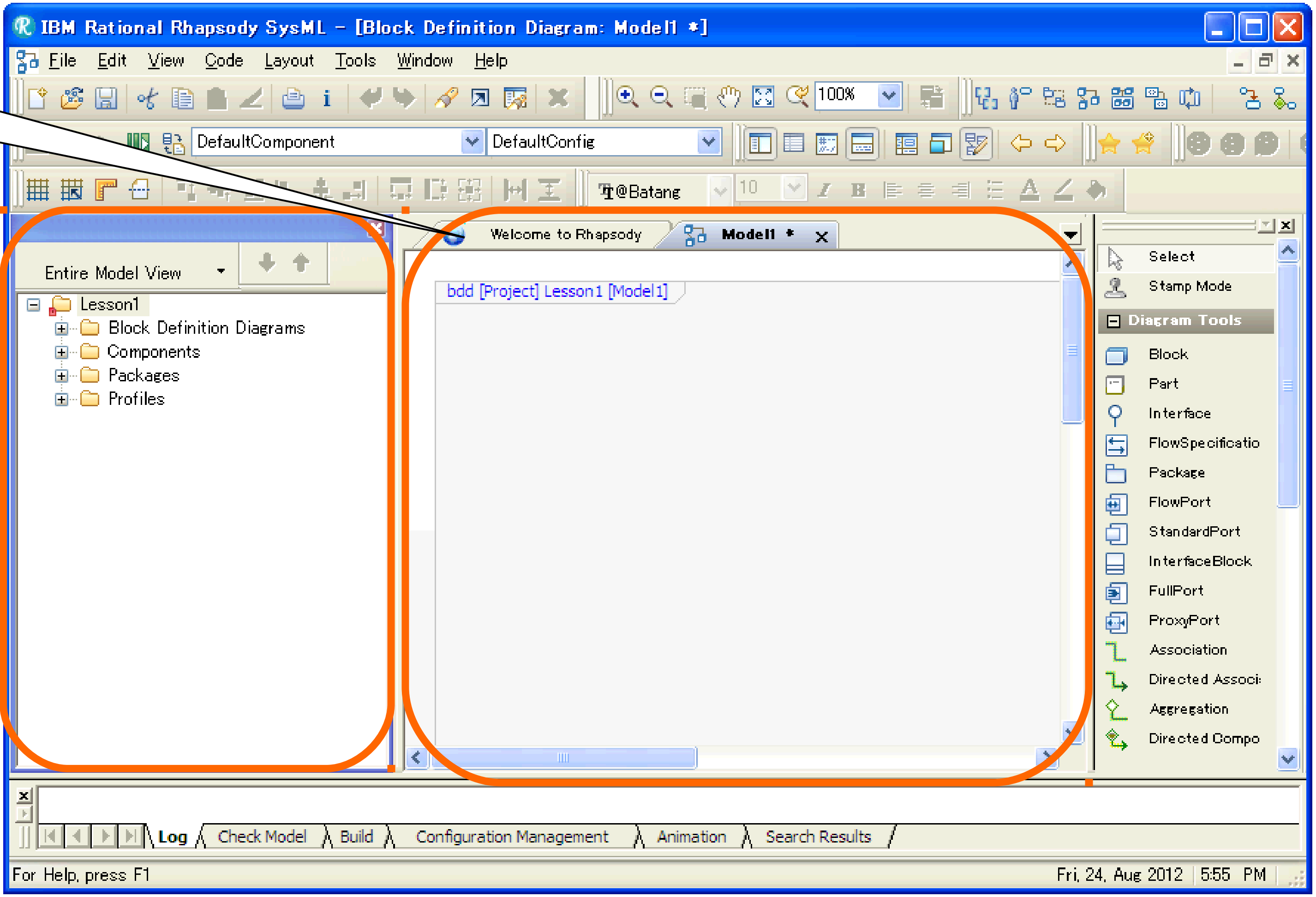
Project Settings: Default

Project Type: This is the Rhapsody implementation of the OMG SysML profile.

Workspace

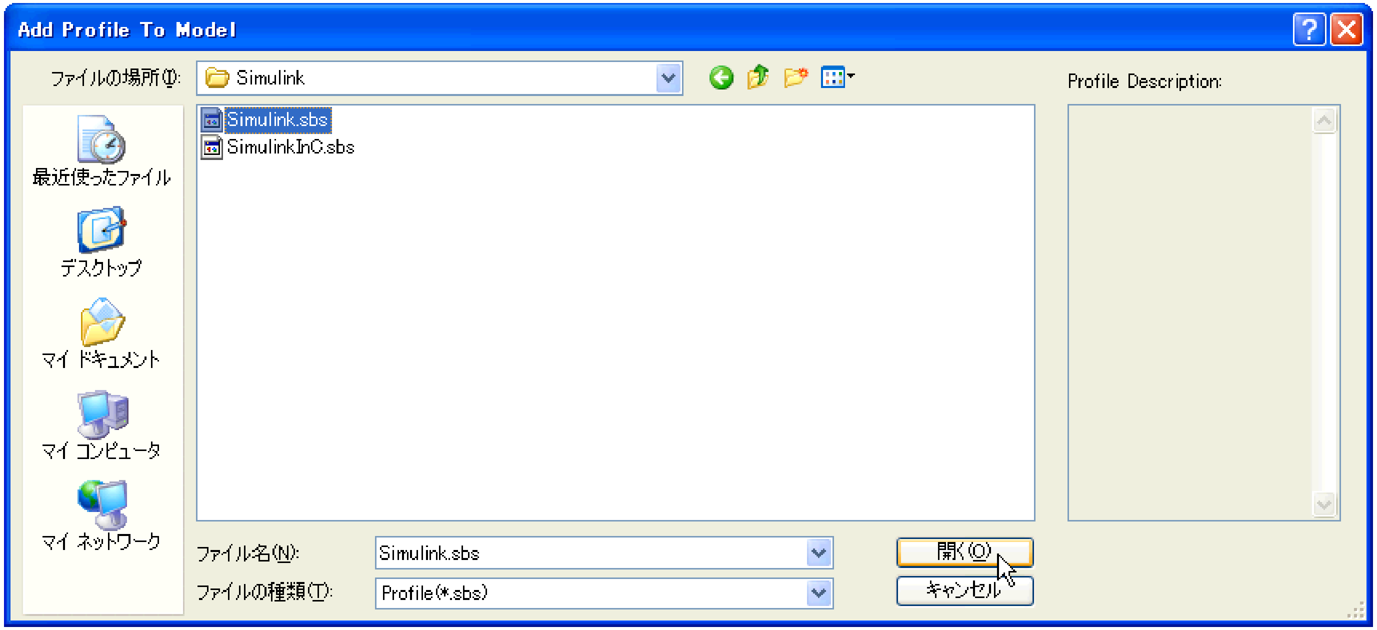
Area to edit diagrams

Browser: shows model elements hierarchically



Adding the Simulink profile

- Select the Lesson1 model on the browser, and select “File”/”Add Profile to Model…” from the menu.
- Select “Simulink.sbs”, and click the “Open” button.
  - You can find the file in [Rhapsody installed folder]\Share\Profiles\Simulink.
  - Note: If you are using Rhapsody Developer for C, you have to specify “SimulinkInC.sbs” instead of “Simulink.sbs”.
- You can see the Simulink profile on the browser.



Add Profile To Model

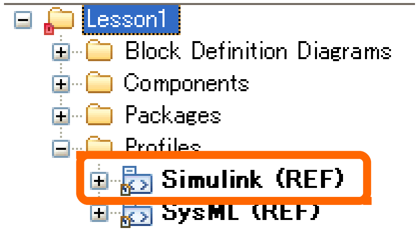
ファイルの場所(I): Simulink

Simulink.sbs
SimulinkInC.sbs

Profile Description:

ファイル名(N): Simulink.sbs

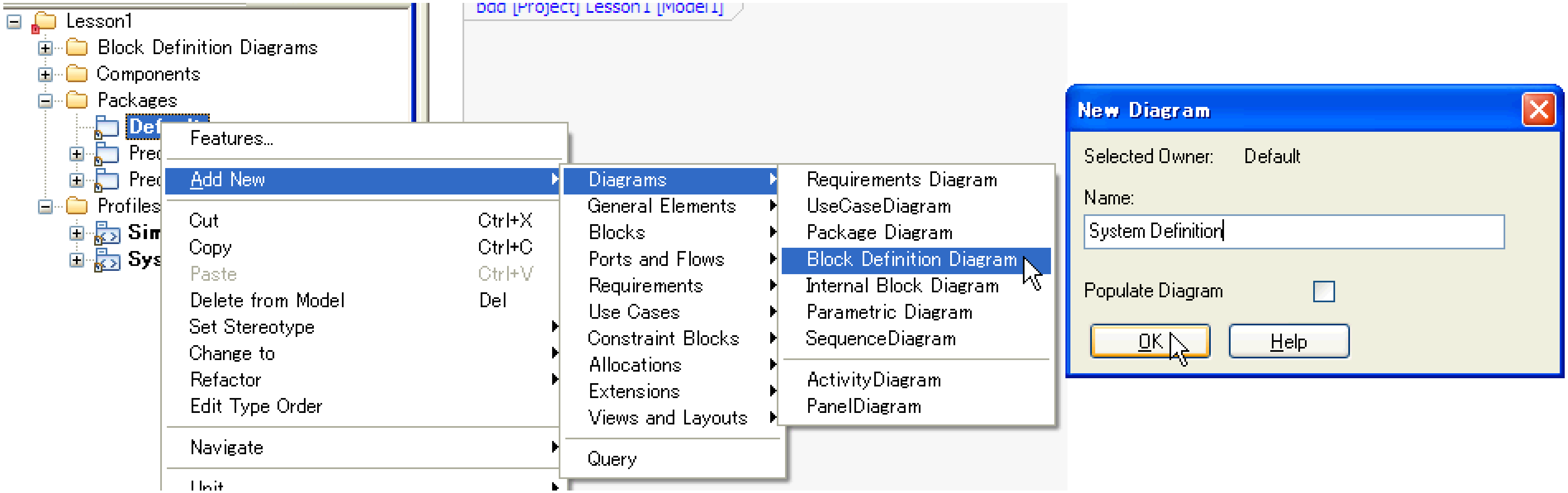
ファイルの種類(T): Profile(*.sbs)



```
Lesson1
  Block Definition Diagrams
  Components
  Packages
  Profiles
    Simulink (REF)
    SysML (REF)
```

Creating a Block Definition Diagram (BDD) (1/3)

- Right-click the “Default” package, and select “Add New”/”Diagrams”/”Block Definition Diagram”.
  - – BDD is used to define relations among blocks.
- Name the diagram “System Definition”.



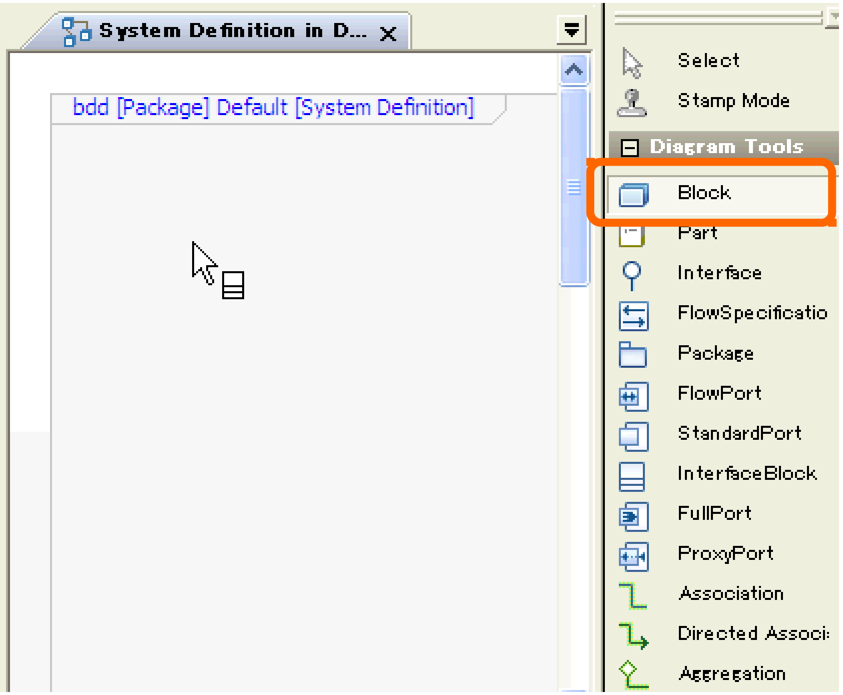
Selected Owner: Default

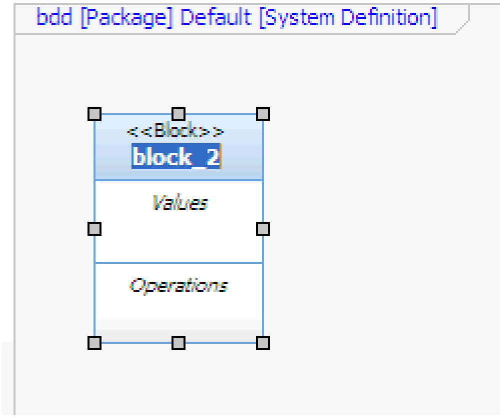
Name: System Definition

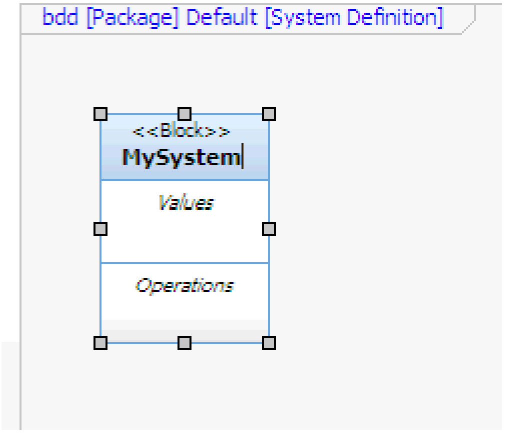
Populate Diagram □

Creating a Block Definition Diagram (BDD) (2/3)

- Select “block”, and click the diagram editing area.
  - A new block with tentative name is created.
- Select the name of the block, and enter “MySystem” from the keyboard.

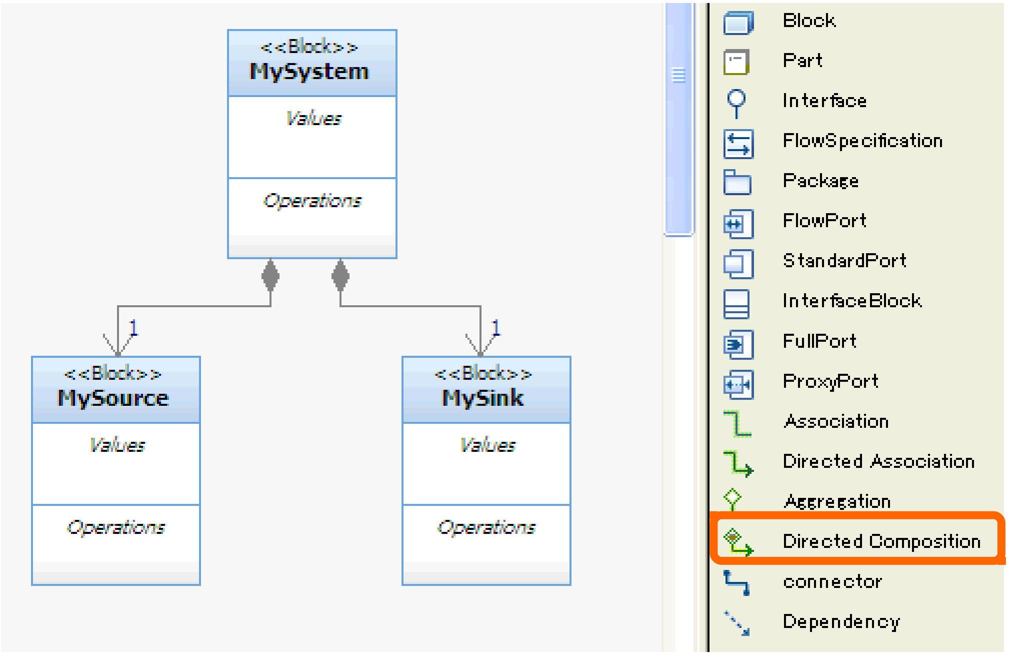






Creating a Block Definition Diagram (BDD) (3/3)

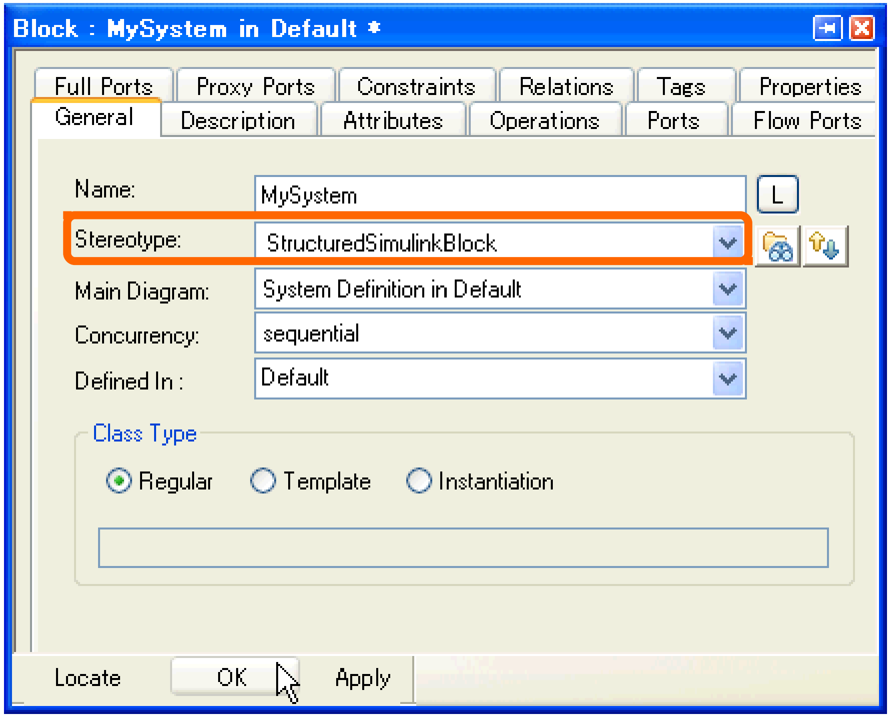
- Create additional two blocks, and names them MySource and MySink.
- Select "Directed Composition", and connect between blocks as below.
  - This diagram shows MySystem has MySource and MySink as parts.



Applying <<StructuredSimulinkBlock>> stereotype

- Perform one of the following operations:
  - Double-click the MySystem block on the BDD or on the browser.
  - Right-click the MySystem block on the BDD or on the browser, and select “Features…” from the pop-up menu.

- Select “StructuredSimulinkBlock” as the Stereotype, and click the “OK” button.
  - This stereotype means a Simulink model will be exported from this block.



Block : MySystem in Default *

Name: MySystem

Stereotype: StructuredSimulinkBlock

Main Diagram: System Definition in Default

Concurrency: sequential

Defined In : Default

Class Type

● Regular ○ Template ○ Instantiation

Tags of <<StructuredSimulinkBlock>> stereotype

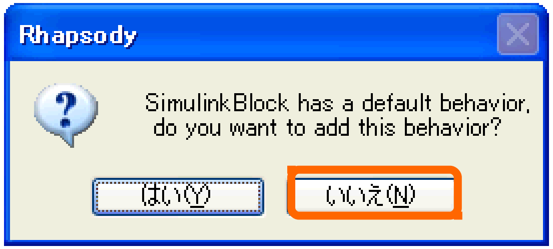
- Tags related to Simulink configuration parameters
  - StartTime: Start time.
  - StopTime: Stop time.
  - SolverType: Solver type (case-insensitive).
    - Variable-step, Fixed-step
  - SolverName: Solver name (case-insensitive).
    - VariableStepDiscrete, ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb, FixedStepDiscrete, ode5, ode4, ode3, ode2, ode1, ode14x
  - Parameters: To specify other configuration parameters, use this tag
    - ‘parameter name1’, ‘parameter value1’, ‘parameter name2’, ‘parameter name3’,…

- Other tags
  - SampleTime: Sample time for an S-Function (positive, 0, or -2). 0 is continuous time, and -2 is variable time.
  - DiagramForSimulink: Use this tag to specify IBD’s name for Simulink model export if this block has multiple IBDs.
  - Plots: Signal names to be plotted.
  - MatlabExePath: Not used in Rhapsody V8.0.

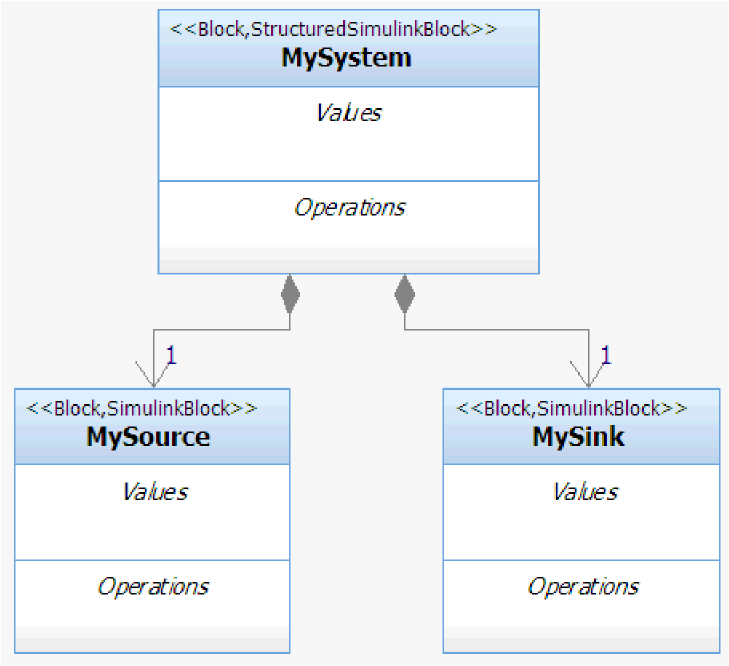
- Important note: The units of StartTime, StopTime, and SampleTime are seconds.

Applying <<SimulinkBlock>> stereotype

- Apply <<SimulinkBlock>> stereotype to MySource block and MySink block.
  - This stereotype means the behavior of this block is defined by a Simulink model.
  - Answer “No” for the dialog because we do not want to add behavior using SysMLUML.
- If you want to enlarge blocks, select blocks and press “Ctrl” key and “E” key simultaneously.

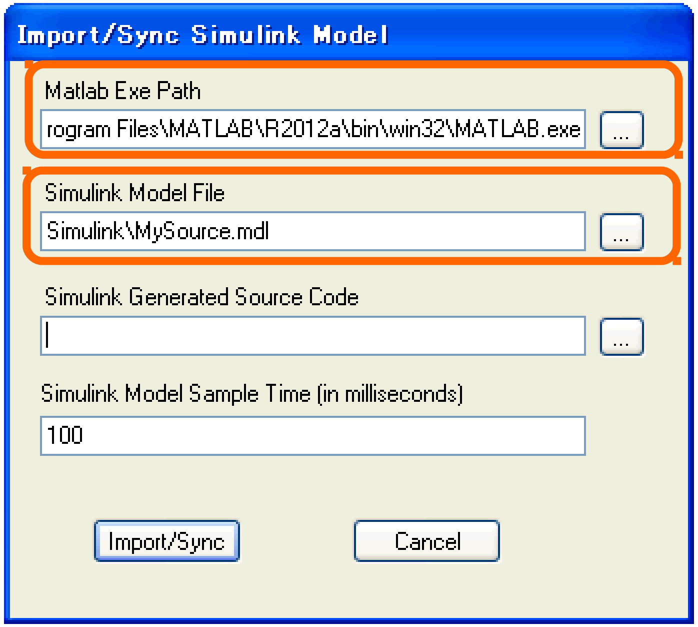


SimulinkBlock has a default behavior,
do you want to add this behavior?



Synchronizing blocks with Simulink models

- Right-click MySource block on the BDD or on the browser, and select “Import/Sync Simulink model” from the pop-up menu.
- Specify the MATLAB.exe (MATLAB are capital letters.) path.
  - Please do not specify the matlab.exe (matlab are small letters) path.
- Specify “Simulink\MySource.mdl” as Simulink Model File.
  - Ignore other fields.
- Click the “Import/Sync” button.
  - SysML flow port is generated from Simulink port.
- Synchronize MySink block with Simulink\MySink.mdl.

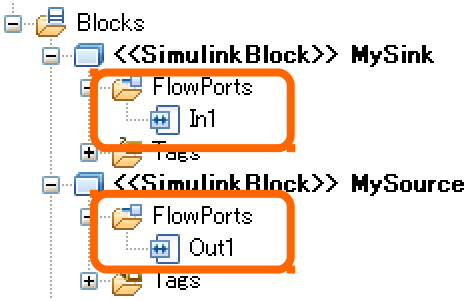

Import/Sync Simulink Model

Matlab Exe Path: rogram Files\MATLAB\R2012a\bin\win32\MATLAB.exe

Simulink Model File: Simulink\MySource.mdl

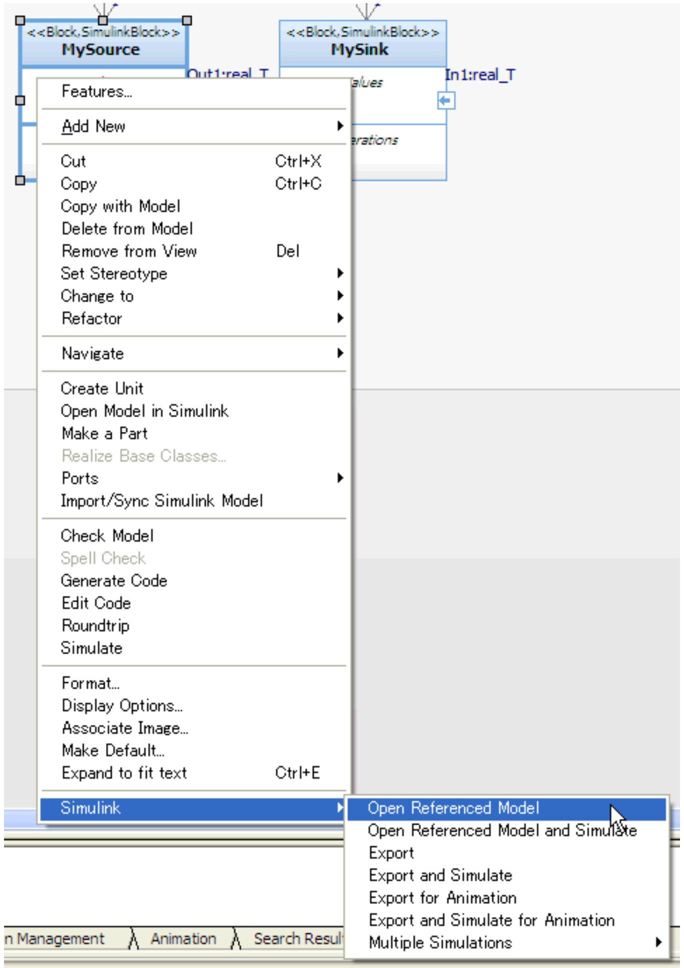
Simulink Generated Source Code:

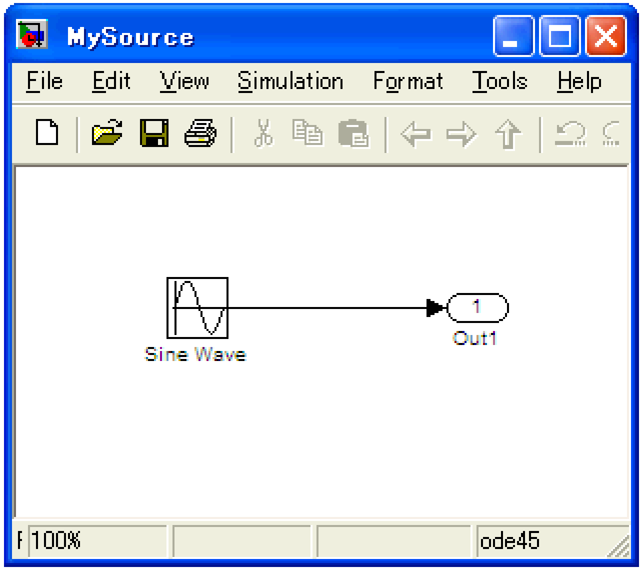
Simulink Model Sample Time (in milliseconds): 100


- Blocks
  - <<SimulinkBlock>> MySink
    - FlowPorts
      - In1
    - Tags
  - <<SimulinkBlock>> MySource
    - FlowPorts
      - Out1
    - Tags

Opening referenced Simulink model

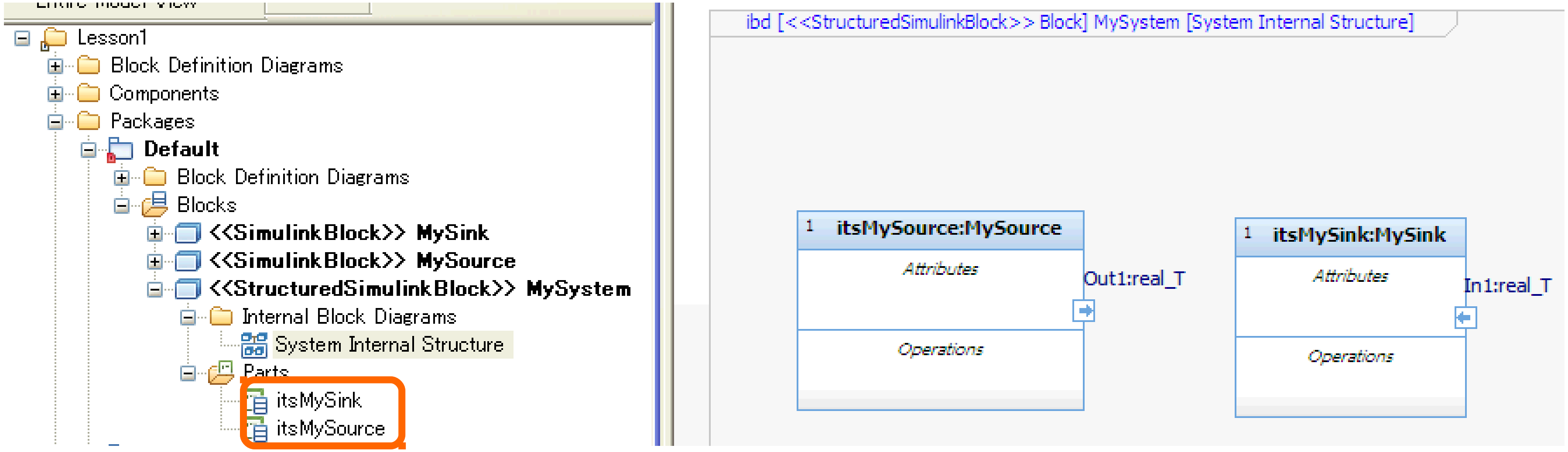
- Right-click MySource block, and select “Simulink”/”Open Referenced Model”.





Creating an Internal Block Diagram (IBD) (1/3)

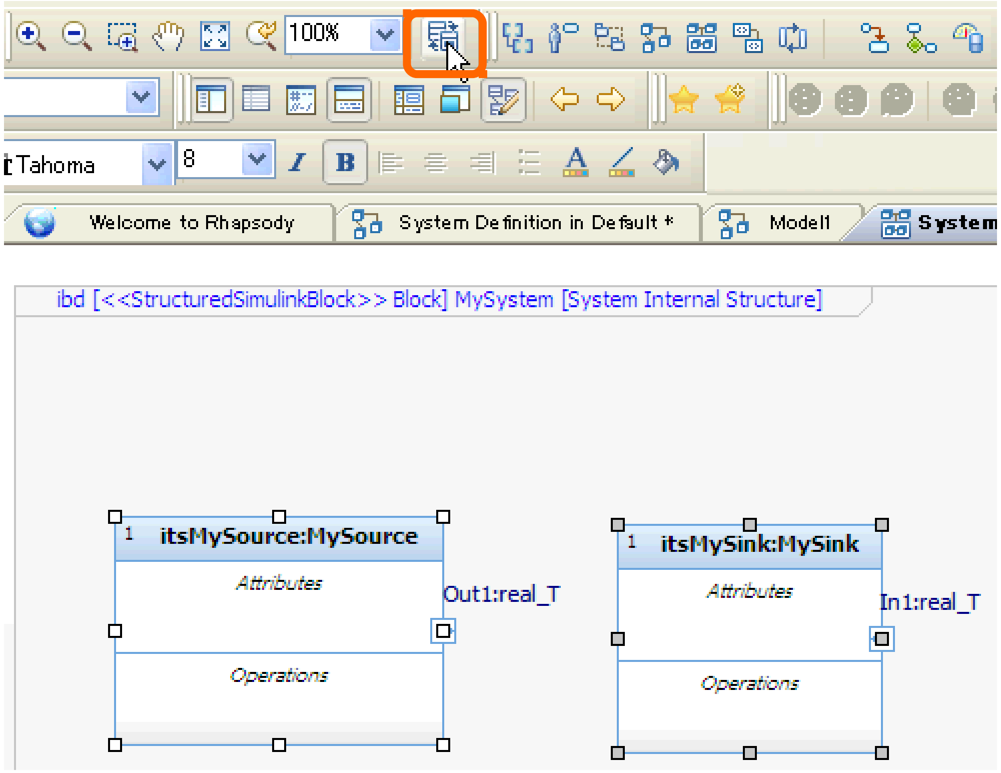
- Select the System block on the BDD or on the browser, and select “Add New”/”Internal Block Diagram” (“Add New”/”Diagrams”/”Internal Block Diagram”) from the pop-up menu.
  - Name the diagram “System Internal Structure”.
  - IBD defines the internal structure of a block.
- Drag the itsMySource part from the browser and drop it to the diagram.
- Perform the same operation for itsMySink.



Lesson1
Block Definition Diagrams
Components
Packages
Default
Block Definition Diagrams
Blocks
<<SimulinkBlock>> MySink
<<SimulinkBlock>> MySource
<<StructuredSimulinkBlock>> MySystem
Internal Block Diagrams
System Internal Structure
Parts
itsMySink
itsMySource
ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]
1 itsMySource:MySource
Attributes
Operations
Out1:real_T
1 itsMySink:MySink
Attributes
Operations
In1:real_T

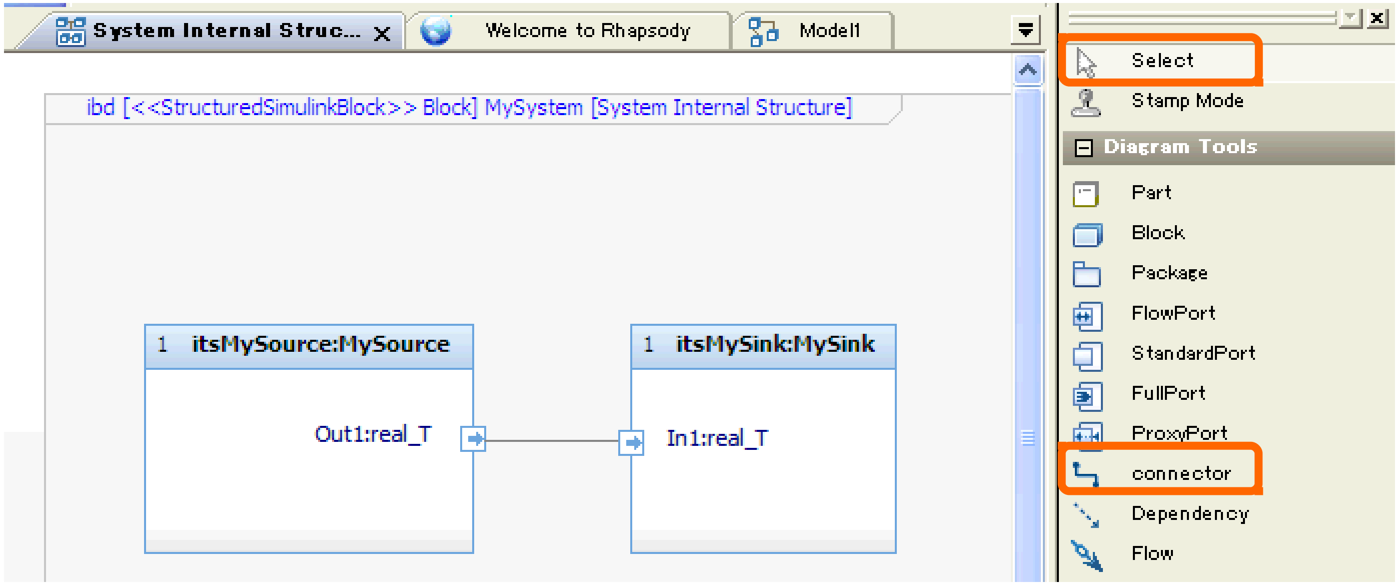
Creating an Internal Block Diagram (IBD) (2/3)

- Select itsMySource part and itsMySink part, and click “Specification/Structured View” icon.



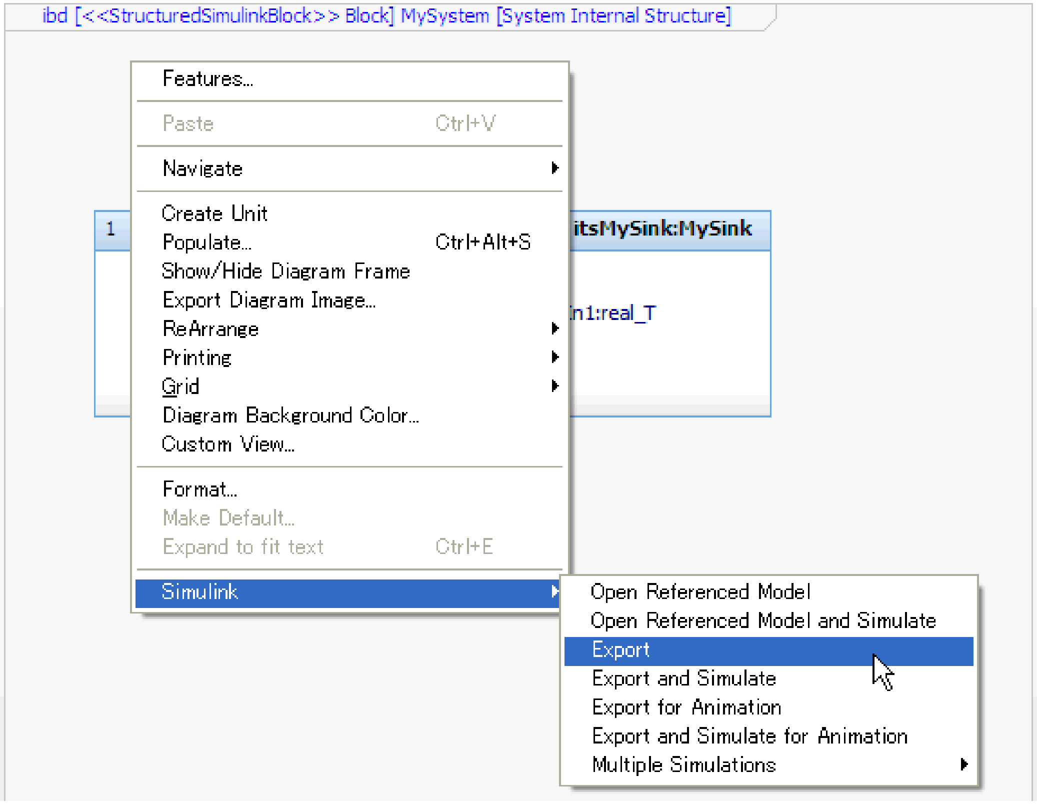
Creating an Internal Block Diagram (IBD) (3/3)

- Select “Select”, and change the positions of flow ports and labels.
  - Place the output flow port to right side of the block, and the input flow port to the left side of the block.
- Select “connector”, and connect between flow ports.
  - This means value change of Out1 flow port is propagated to In1 flow port.



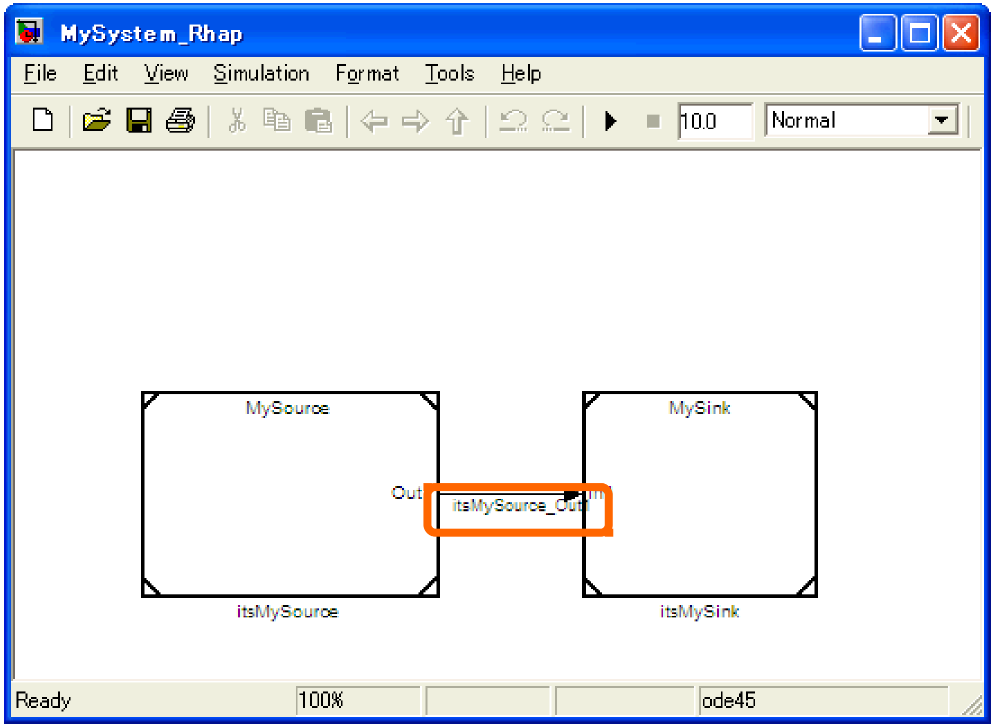
Exporting a Simulink model (1/2)

- Right-click on the System block’s IBD or the System block on the browser, and select “Simulink”/”Export” from the pop-up menu.
  - Note: Please wait. Do not change the active component and the active configuration until “Simulink Model Export/Update Done” is shown on the Rhapsody “Log” window.
  - MATLAB will be started automatically, and a Simulink model will be exported and displayed.



Exporting a Simulink model (2/2)

- Simulink model MySystem_Rhap.mdl is exported in C:\Work\Lesson1\MySystemComp\Simulink.
  - itsMySource part and itsMySink part refers MySource and MySink respectivelly.
  - The positions and sizes of blocks on the Simulink model are same as those of parts on the IBD.
  - Name of the generated signal is itsMySource_Out1.

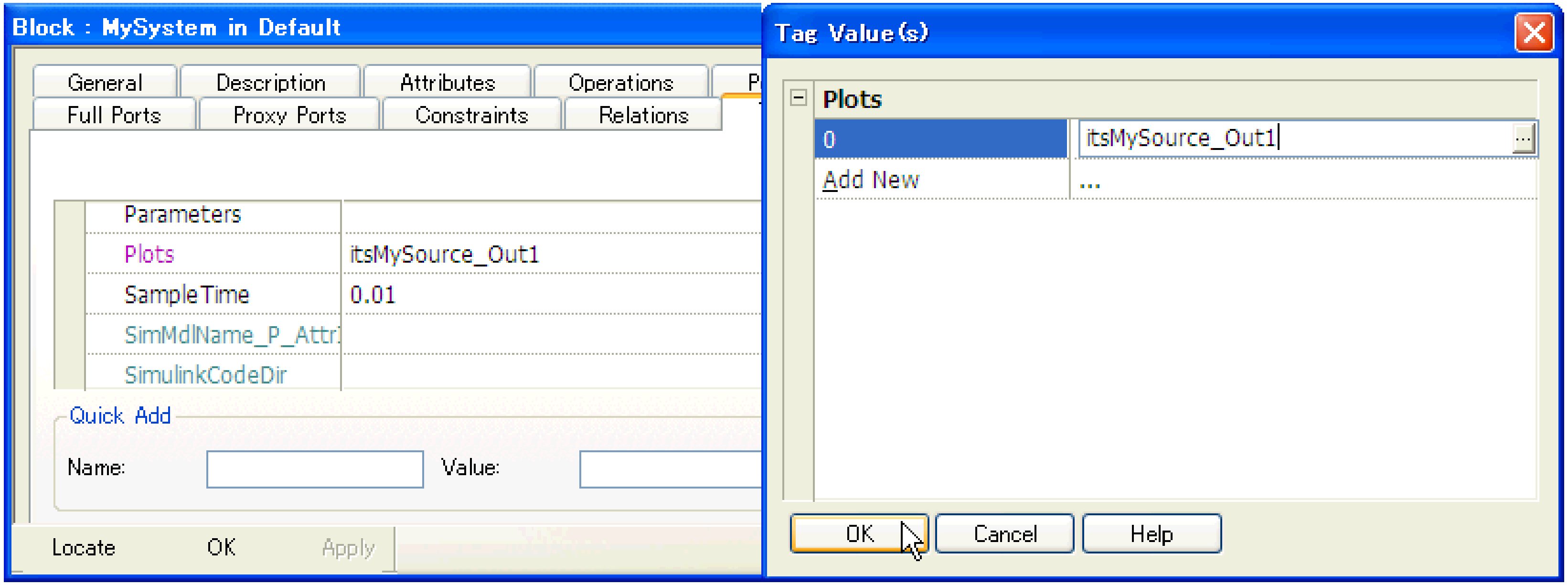


IBD to be used for Simulink model export

- A block may have multiple IBD.
- IBD to be used for Simulink model export is determined as follows:
  - If an IBD is selected, use it.
  - If a block has only one IBD, use it.
  - Otherwise, use an IBD that name is same as the DiagramForSimulink tag value of <<StructuredSimulinkBlock>> stereotype.

Setting a signal name to be plotted (1/2)

- Set the Plots tag value of MySystem block to itsMySource_Out1.
  - itsMySource_Out1 is a signal name defined in MySystem_Rhap Simulink model.



Block : MySystem in Default

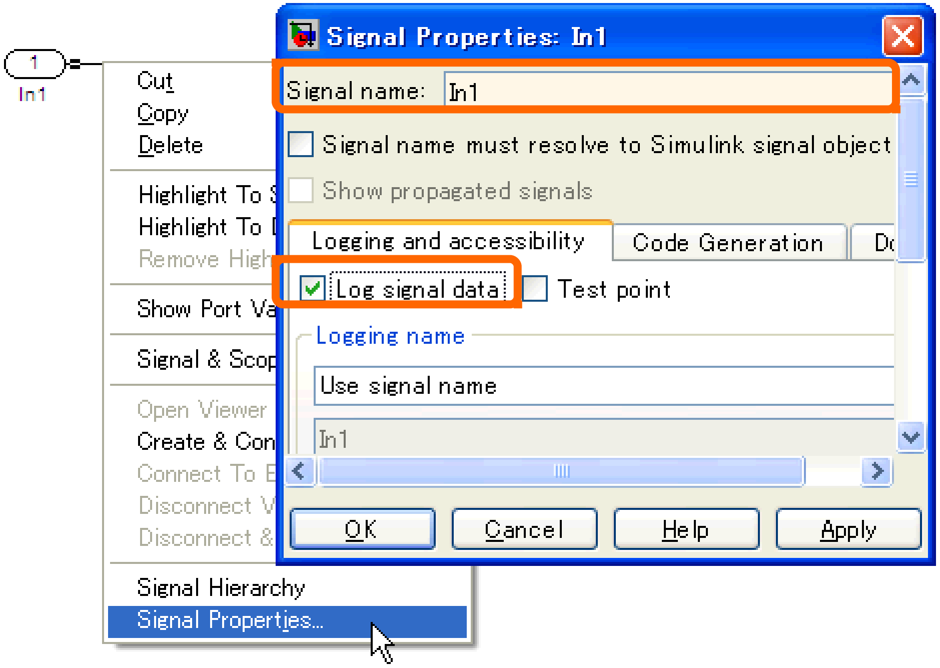
Parameters	
Plots	itsMySource_Out1
SampleTime	0.01
SimMdlName_P_Attr1	
SimulinkCodeDir	

Tag Value(s)

Plots	
0	itsMySource_Out1
Add New	...

Setting a signal name to be plotted (2/2)

- Note: We can specify signal names in referenced Simulink model if we use MATLAB/Simulink R2012a (?) or later.
  - *PartName.SimulinkSignalName* such as itsMySink.In1
  - Right-click a signal on the referenced Simulink model window, open Signal Properties dialog, check the “Log signal data”, and specify the signal name.



Signal Properties: In1

Signal name: In1

□ Signal name must resolve to Simulink signal object

□ Show propagated signals

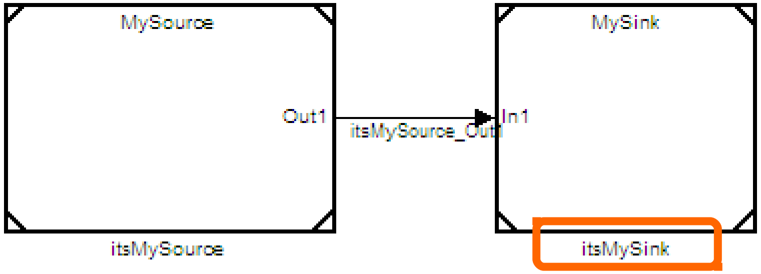
Logging and accessibility | Code Generation | Dc

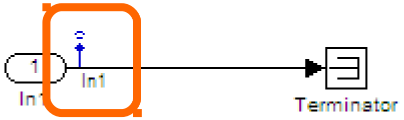
☑ Log signal data □ Test point

Logging name

Use signal name

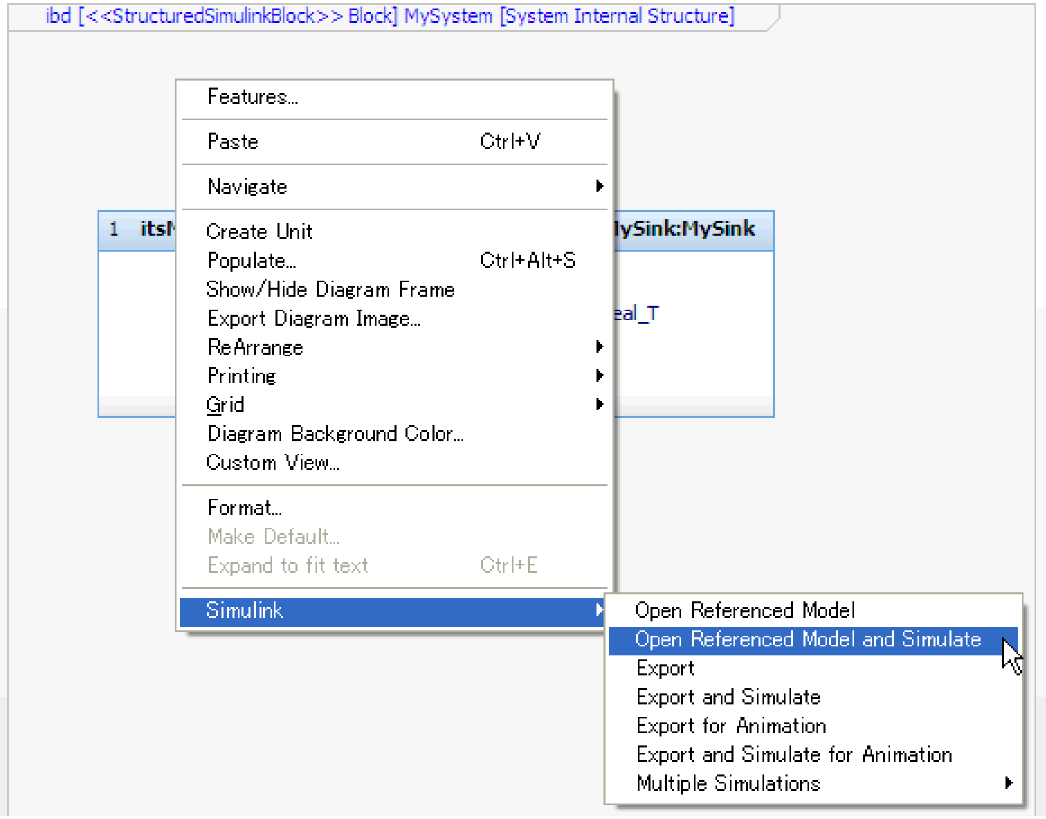
In1

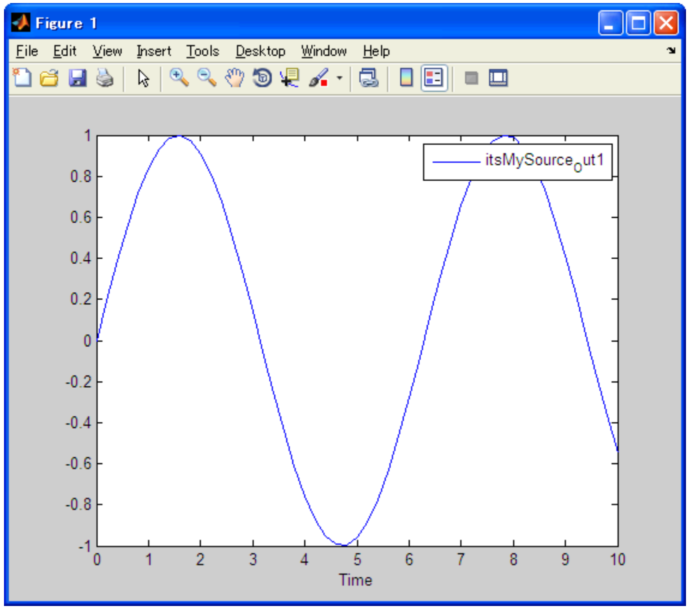




Executing simulation

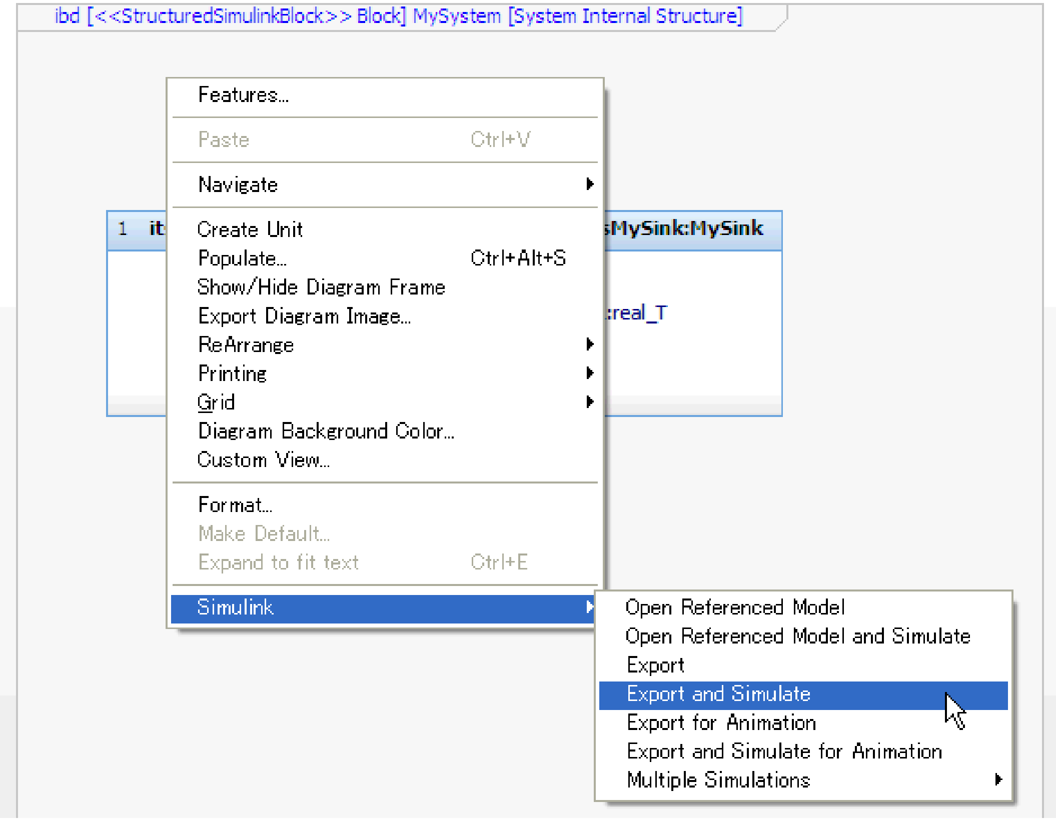
- Right-click on the System block’s IBD or the System block on the browser, and select “Simulink”/”Open Referenced Model and Simulate” from the pop-up menu.
  - Figure of the specified signal is shown.





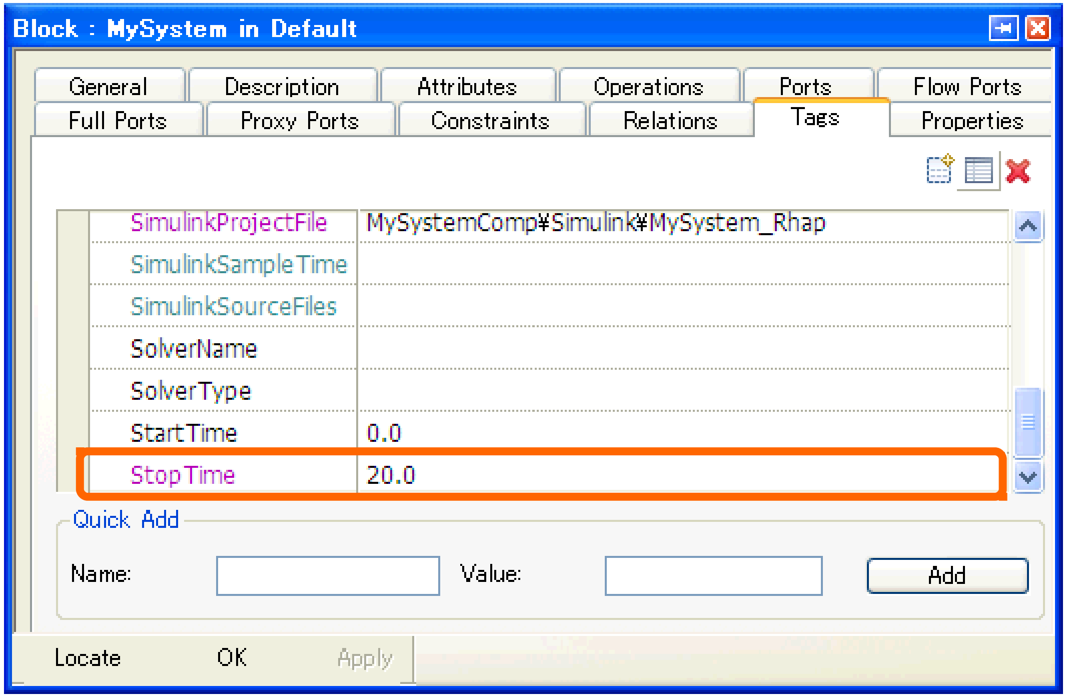
Executing simulation automatically

- If you select “Simulink”/”Export and Simulate” instead of “Simulink”/”Export”, the simulation will be started automatically after the model export.


ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]

Changing configuration parameters

- Double-click MySystem block, and change configuration parameters.



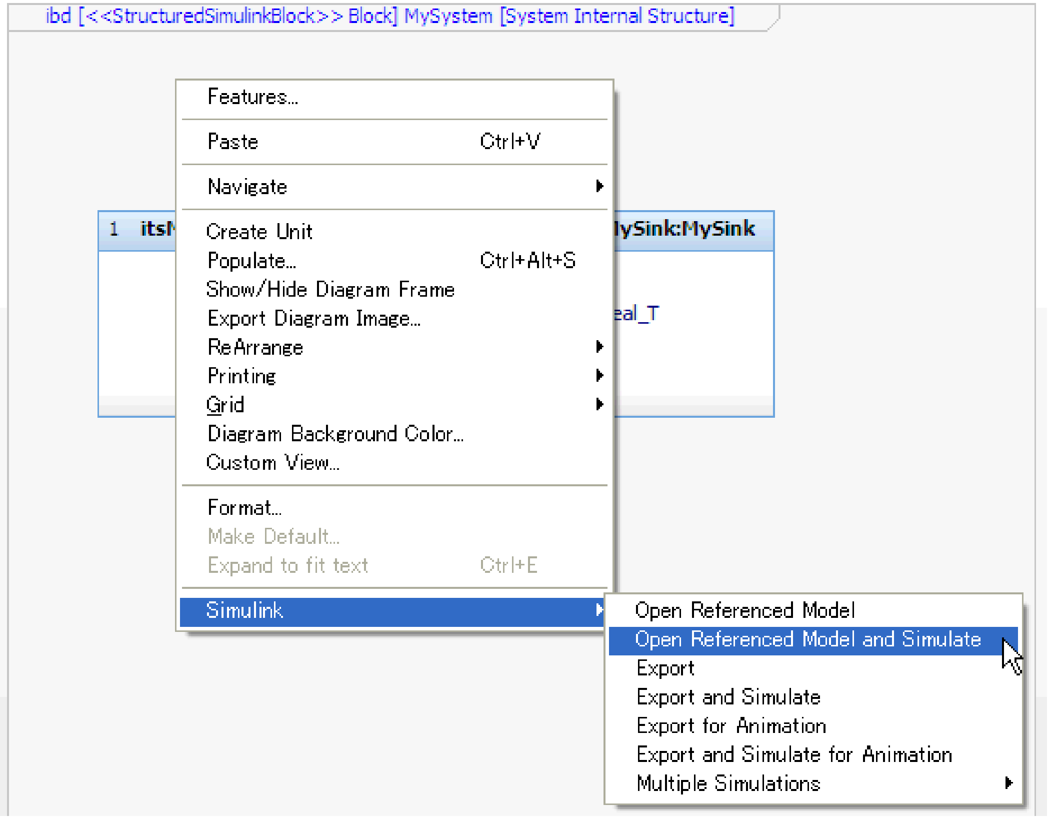
SimulinkProjectFile	MySystemComp¥Simulink¥MySystem_Rhap
SimulinkSampleTime	
SimulinkSourceFiles	
SolverName	
SolverType	
StartTime	0.0
StopTime	20.0

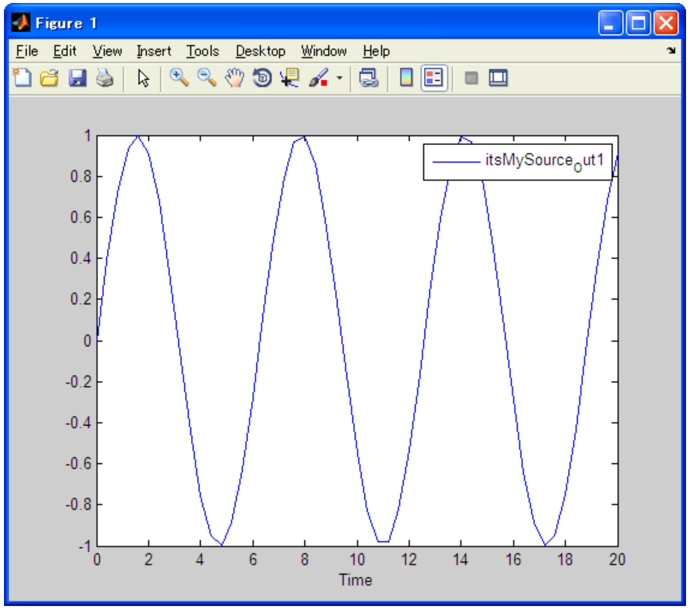
Name:

Value:

Executing simulation

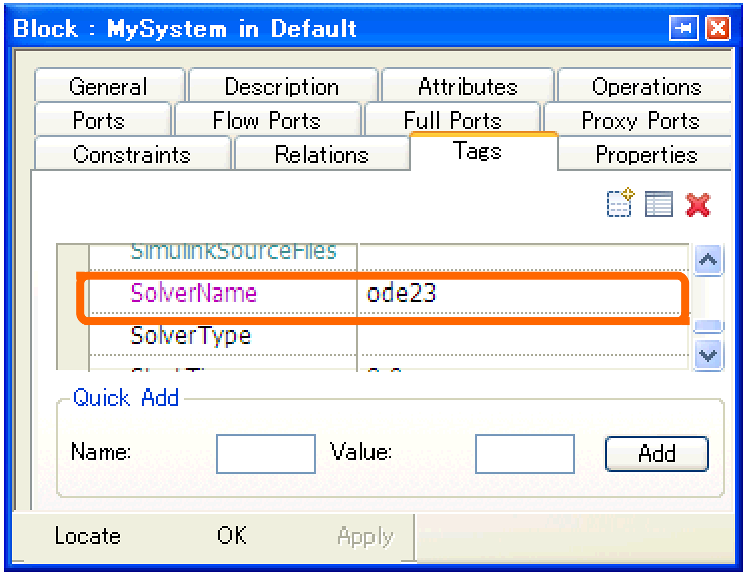
- Right-click on the System block’s IBD or the System block on the browser, and select “Simulink”/”Open Referenced Model and Simulate” from the pop-up menu.
  - Changed configuration parameters are used to execute simulation.



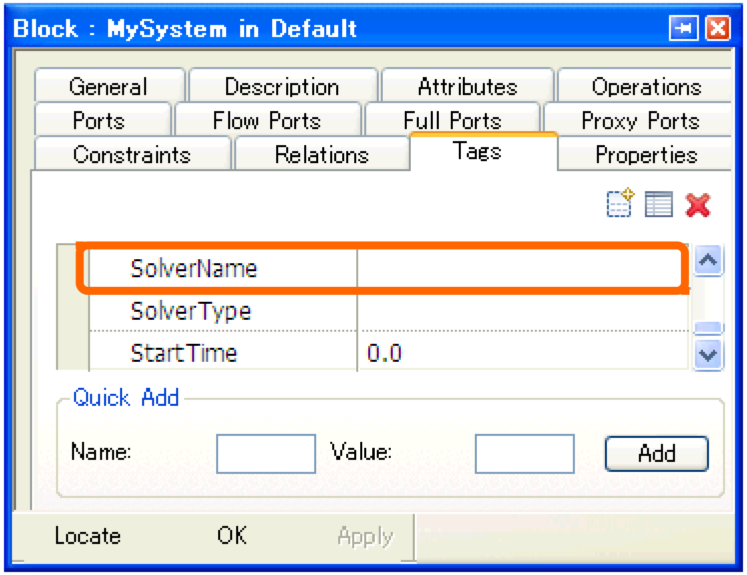


Note: Clearing configuration parameters

- If you clear configuration parameters, perform “Export and Simulate” instead of “Open Referenced Model and Simulate”, or perform “Export” before “Open Referenced Model and Simulate”.



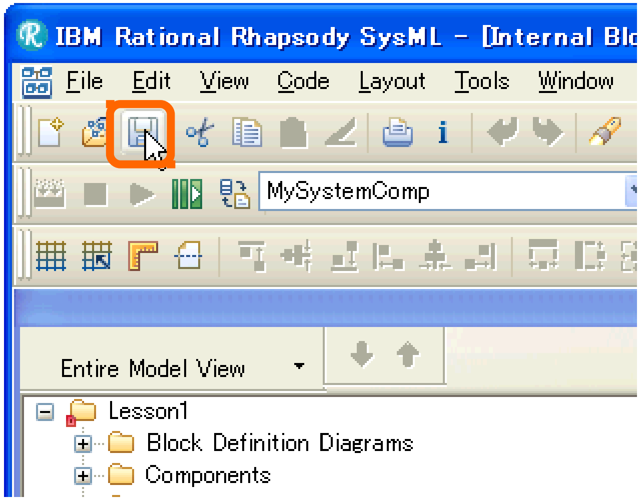
SimulinkSourceFiles	
SolverName	ode23
SolverType	



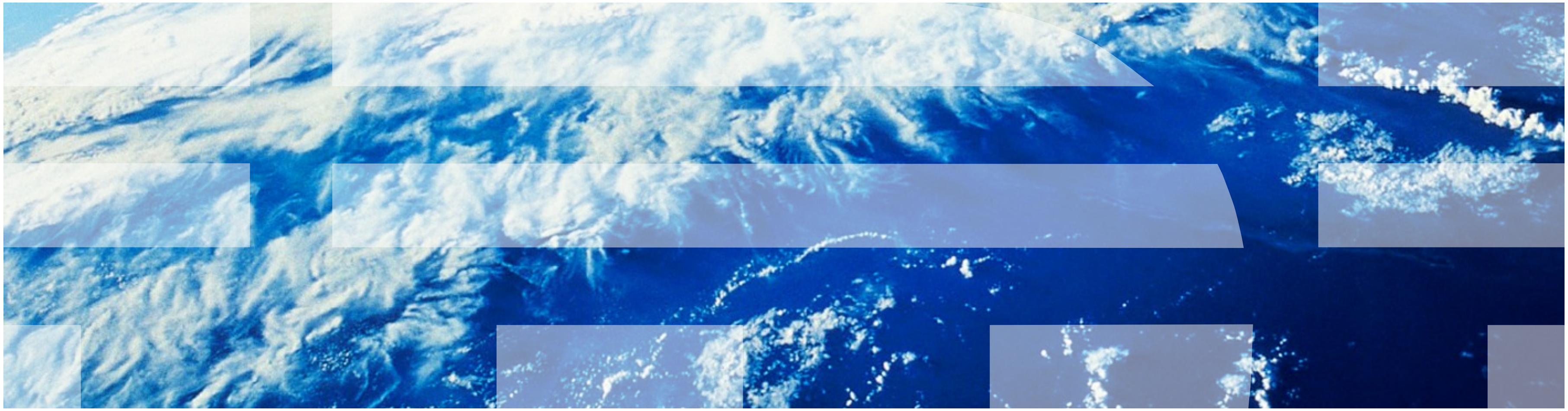
SolverName	
SolverType	
StartTime	0.0

Saving the project

- Perform one of the following operations:
  – Select “File”/”Save” from the menu.
  – Click the “Save” icon.
  – Press “Ctrl” key and “S” key simultaneously.



Lesson 2: Exporting templates of Simulink sub-models

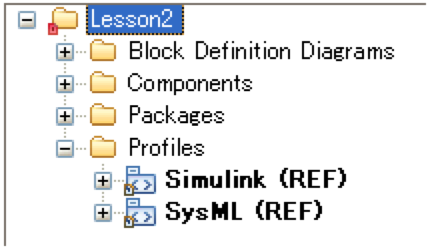


Contents of this lesson

- Create a model containing Simulink sub-models.
  - Define attributes to be converted to MATLAB variables.
- Export templates of Simulink sub-models.
- Edit Simulink sub-models.
  - Use MATLAB variables in Simulink model.
- Export a Simulink model, and simulate it.
- Change the values of MATLAB variables.
- Open the Simulink model, and simulate it using the changed values.
- Use multiple instances of Simulink sub-models.
- Simulate multiple times using a CSV file

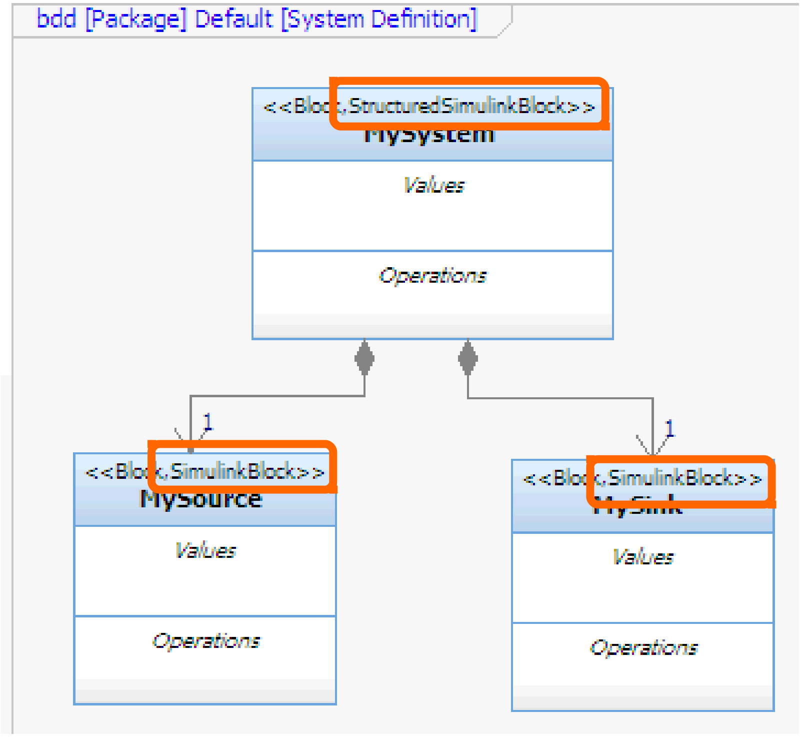
Preparations

- Start Rhapsody.
- Create a SysML project Lesson2 in C:\Work\.
- Add the Simulink profile to the model.



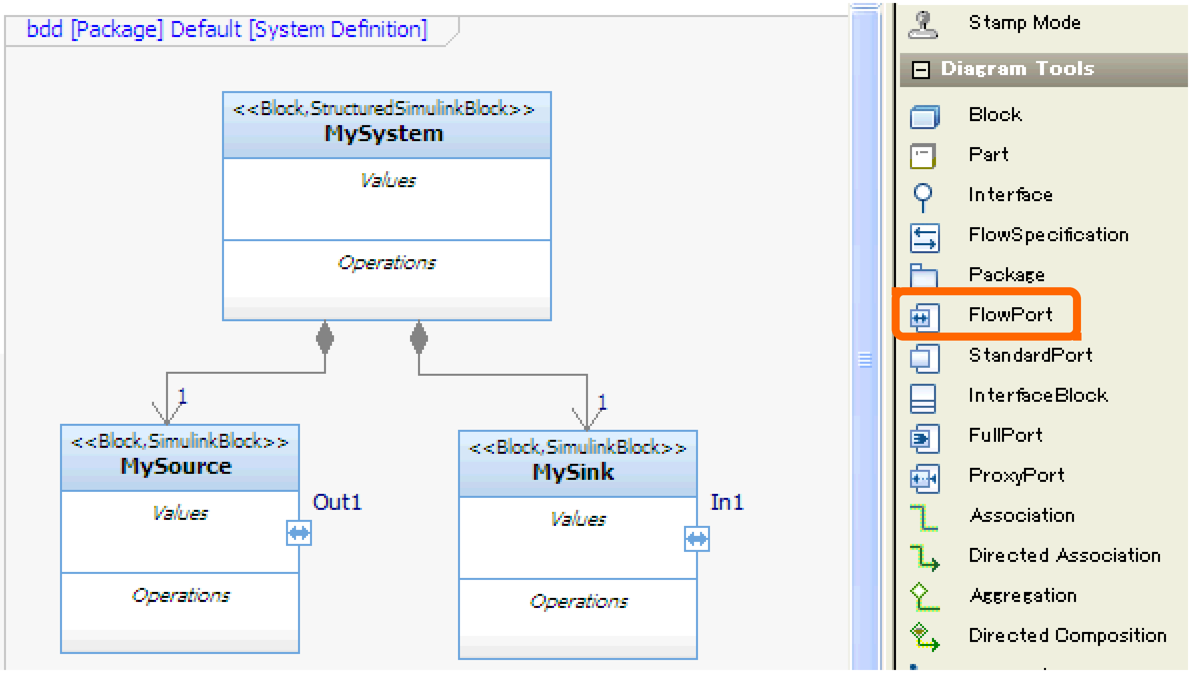
Creating a Block Definition Diagram (BDD)

- Create a BDD under the Default package and place MySystem, MySource, and MySink blocks.
  - Apply <<StructuredSimulinkBlock>> to MySystem block, and <<SimulinkBlock>> stereotypes to MySource and MySink blocks.



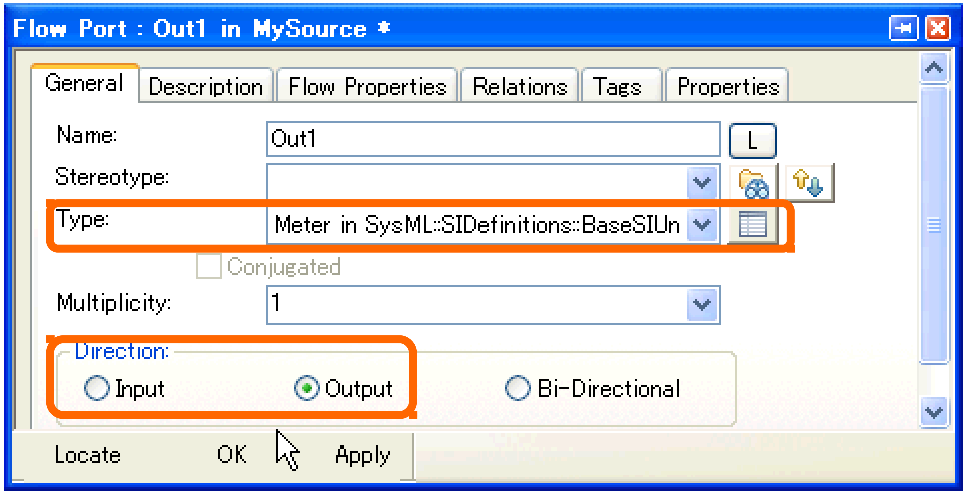
Adding flow ports

- Select “FlowPort”, click on the edge of MySource block, and name it Out1.
- Select “FlowPort”, click on the edge of MySink block, and name it In1.



Setting flow ports’ type and direction

- Set the type of Out1 and In1 to Meter.
- Set the direction of Out1 to Output.
- Set the direction of In1 to Inout


Flow Port : Out1 in MySource *

Name: Out1

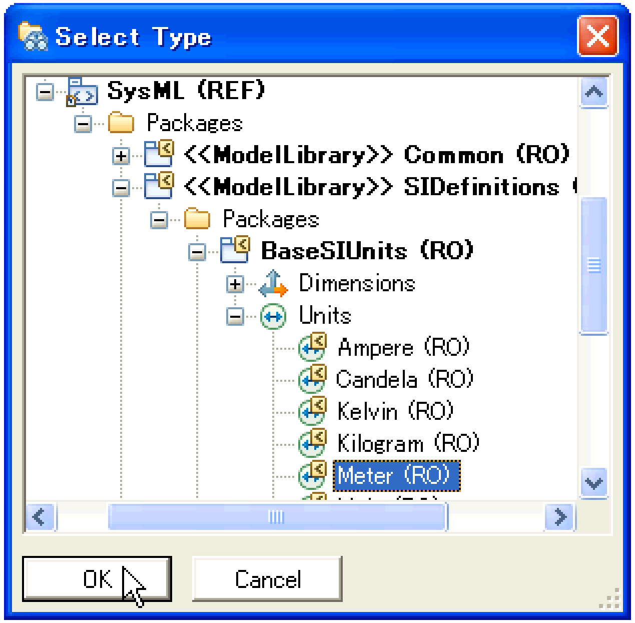
Stereotype:

Type: Meter in SysML::SIDefinitions::BaseSIUn

□ Conjugated

Multiplicity: 1

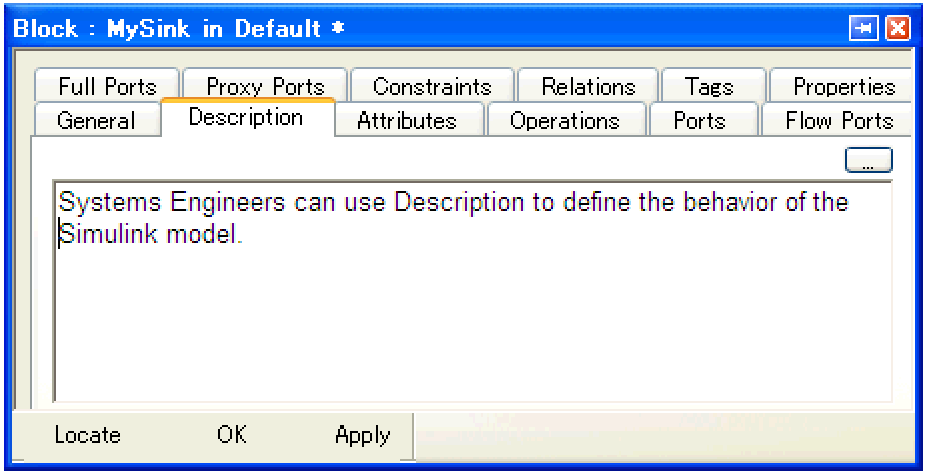
Direction: ○ Input ● Output ○ Bi-Directional


Select Type

```
SysML (REF)
  Packages
    <<ModelLibrary>> Common (RO)
    <<ModelLibrary>> SIDefinitions (
      Packages
        BaseSIUnits (RO)
          Dimensions
          Units
            Ampere (RO)
            Candela (RO)
            Kelvin (RO)
            Kilogram (RO)
            Meter (RO)
```

Setting description

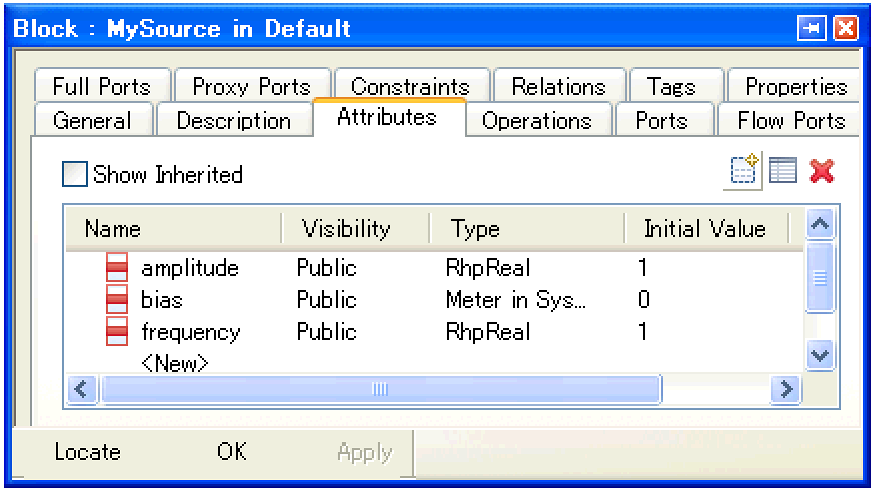
- Double-click MySource block, and enter Description.



Systems Engineers can use Description to define the behavior of the Simulink model.

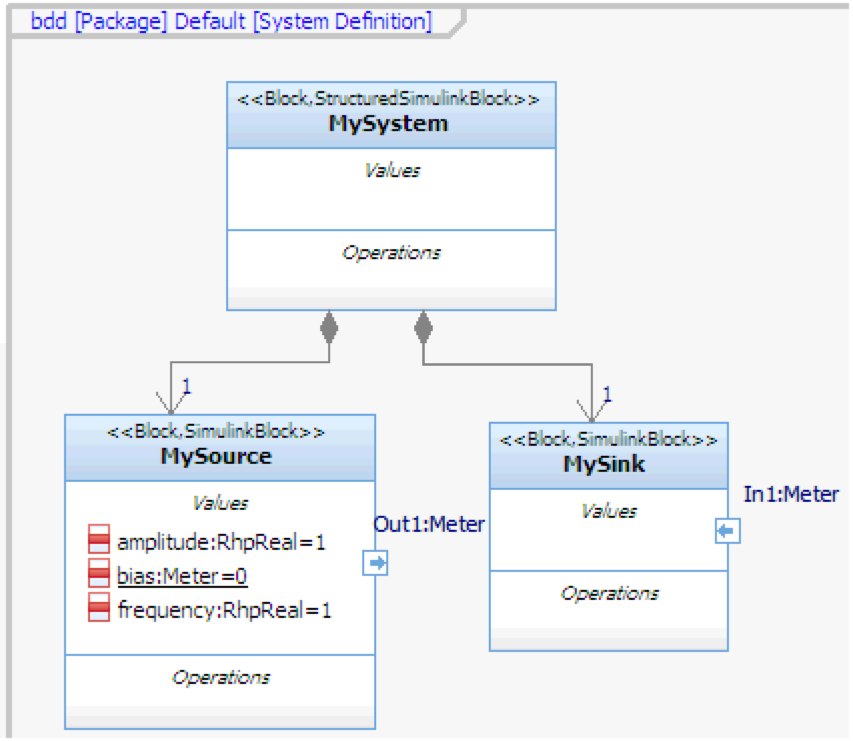
Adding attributes to MySource block

- Add attributes to MySource block
  – bias – static, will be converted to a MATLAB variable
  – amplitude, frequency – dynamic, will be converted to MATLAB variables and model arguments



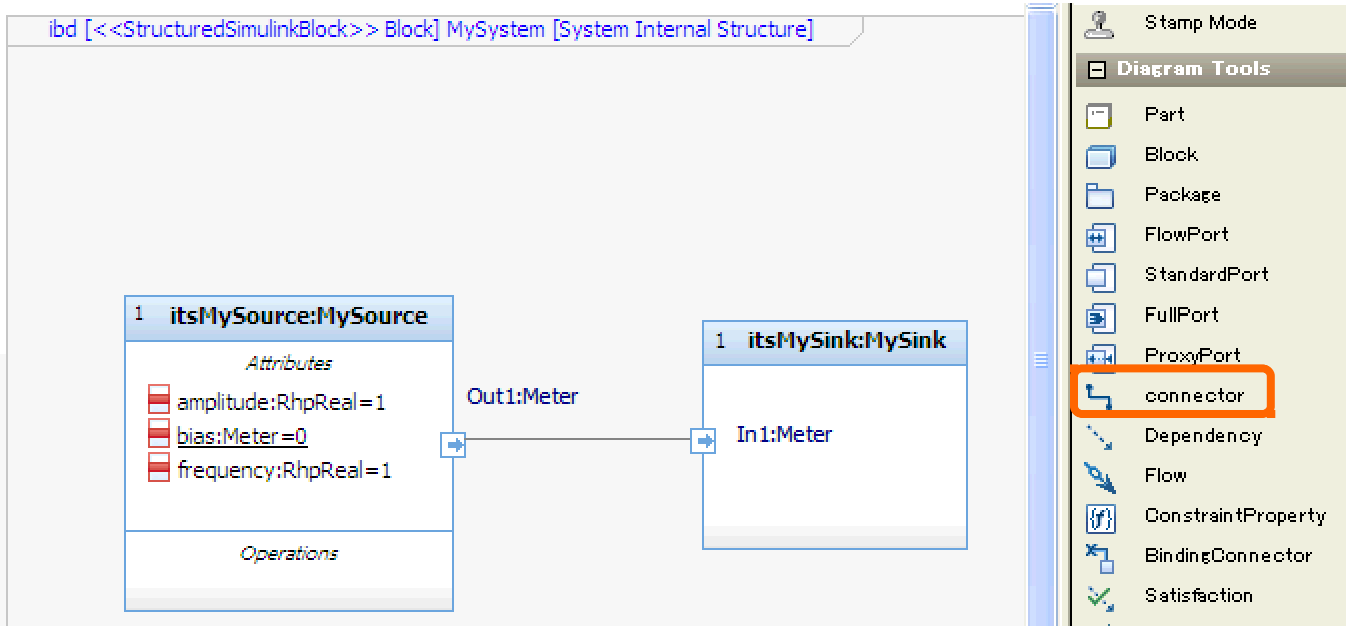
□ Show Inherited

Name	Visibility	Type	Initial Value
amplitude	Public	RhpReal	1
bias	Public	Meter in Sys...	0
frequency	Public	RhpReal	1
<New>			



Creating an IBD of MySystem block

- Create an IBD of MySystem block as below.

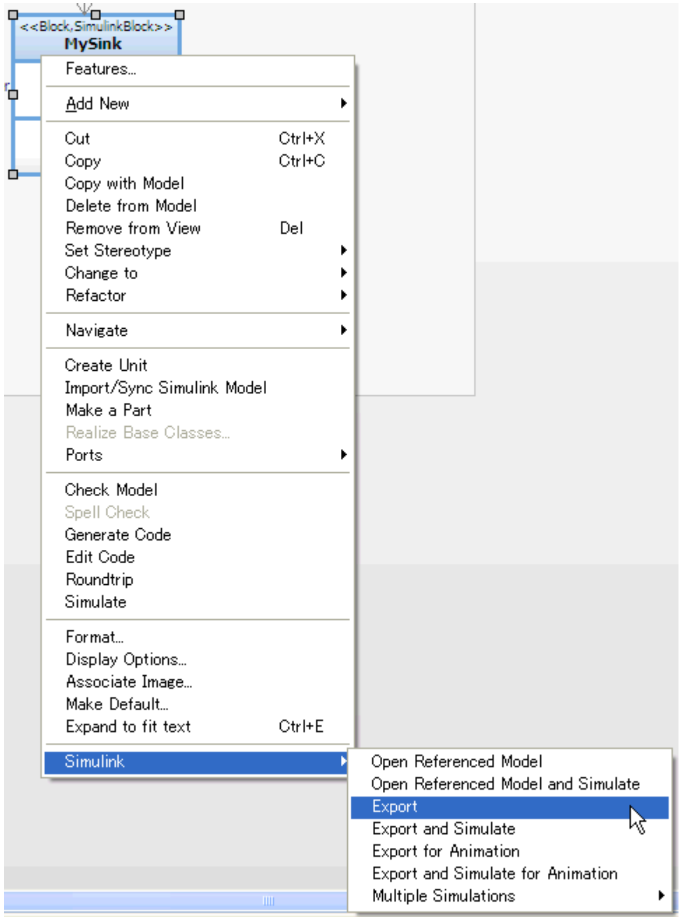


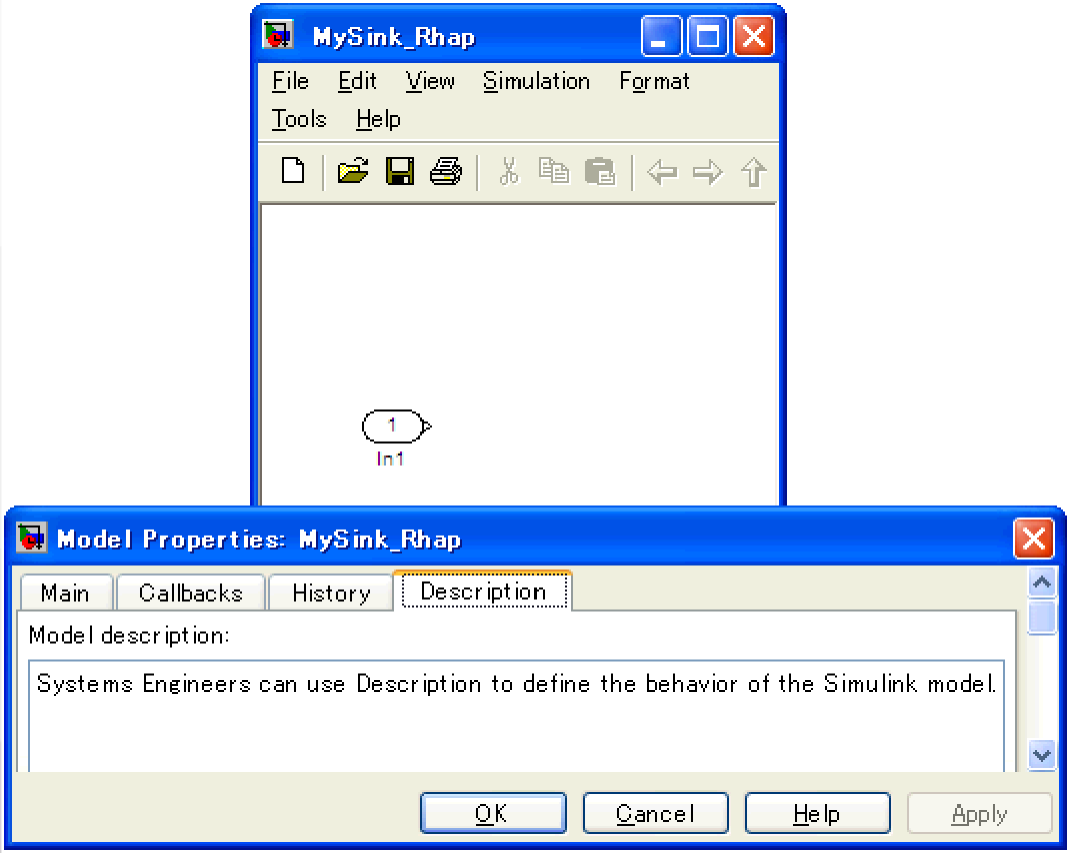
ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]

Exporting a Simulink model template from MySink block

- Export a Simulink model template from MySink block.
  - Note: Please wait. Do not change the active component and the active configuration until “Simulink Model Export/Update Done” is shown on the Rhapsody “Log” window.

- Simulink model MySink_Rhap.mdl is exported in C:\Work\Lesson2\Simulink.





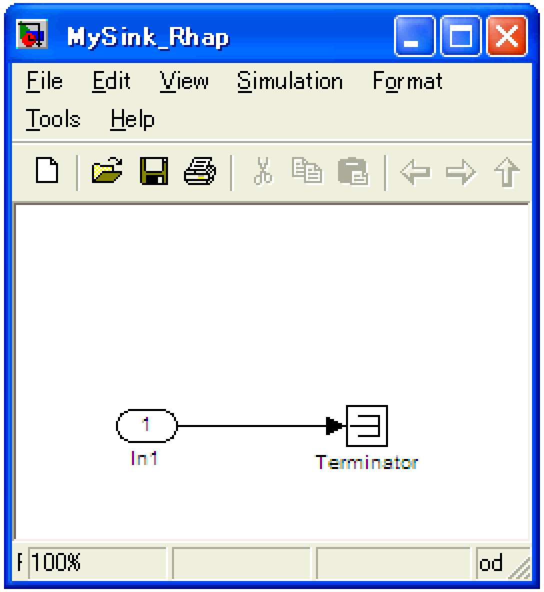
MySink_Rhap
File Edit View Simulation Format
Tools Help
1
In1
Model Properties: MySink_Rhap
Main Callbacks History Description
OK Cancel Help Apply

Model description:

Systems Engineers can use Description to define the behavior of the Simulink model.

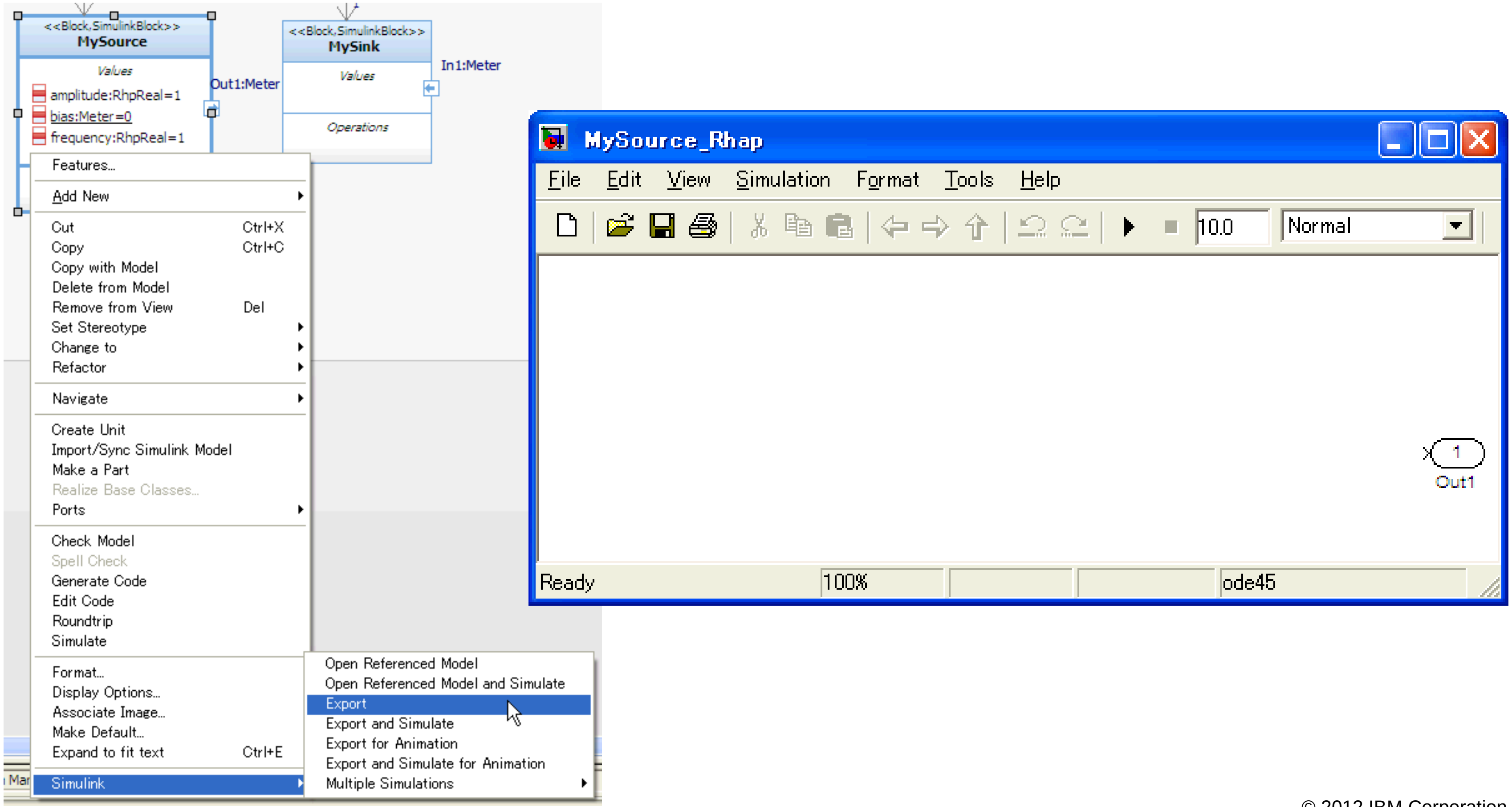
Editing MySink_Rhap Simulink sub-model

- Edit MySink_Rhap.mdl as below, and save it.



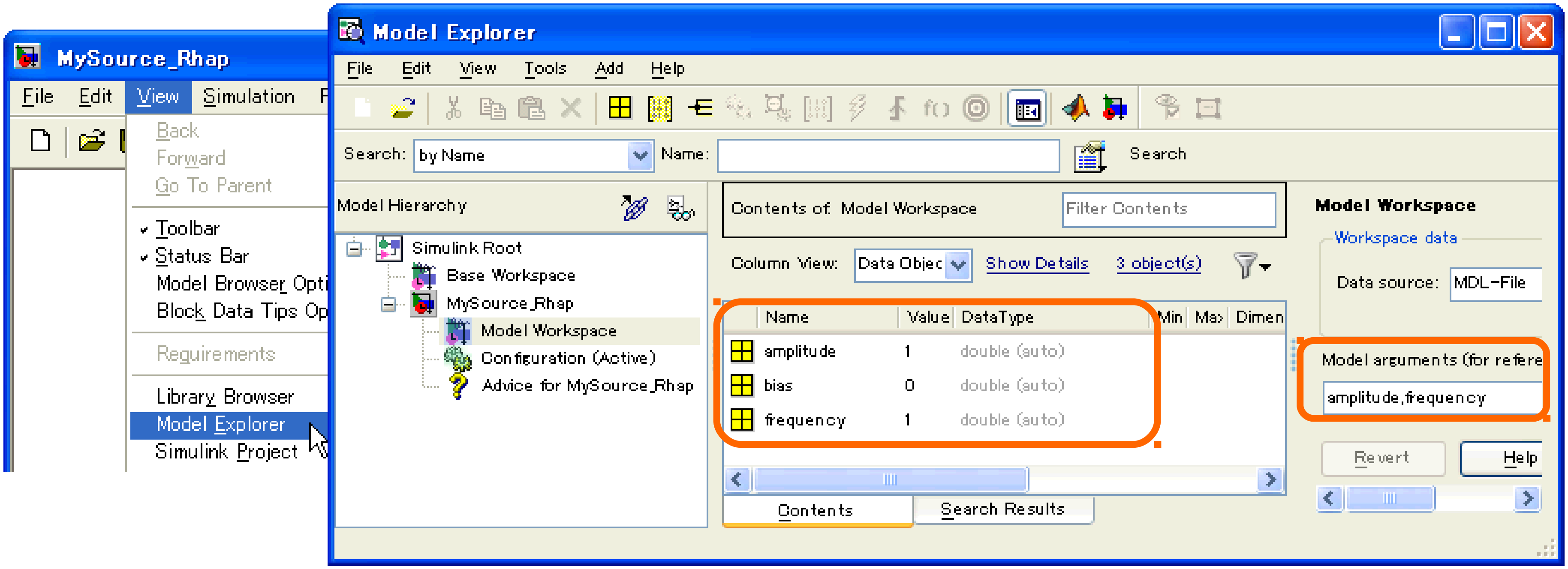
Exporting a Simulink model template from MySource block

- Export a Simulink model template from MySoruce block.
- Simulink model MySink_Rhap.mdl is exported in C:\Work\Lesson2\Simulink.



Checking MATLAB variables and model arguments

- Open Simulink Model Explorer
  - Three MATLAB variables (amplitude, bias, and frequency) are defined in Model Workspace of MySource_Rhap.
  - Two MATLAB variables (amplitude and frequency) are defined as model arguments.
  - bias is not defined as a model argument because it is a static attribute.



Search: by Name

Name:

Contents of: Model Workspace

Column View: Data Objec

Name	Value	DataType	Min	Max	Dimen
amplitude	1	double (auto)			
bias	0	double (auto)			
frequency	1	double (auto)			

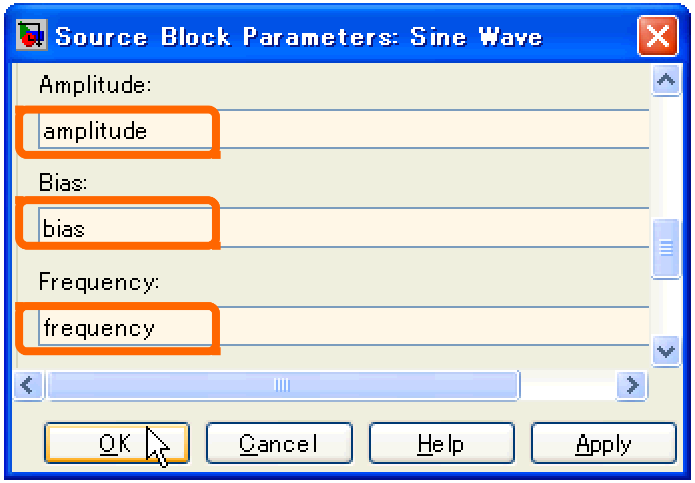
Data source: MDL-File

Model arguments (for refere

amplitude,frequency

Editing MySource_Rhap Simulink sub-model

- Edit MySink_Rhap.mdl as below, and save it.
  - We can use MATLAB variables in this model.

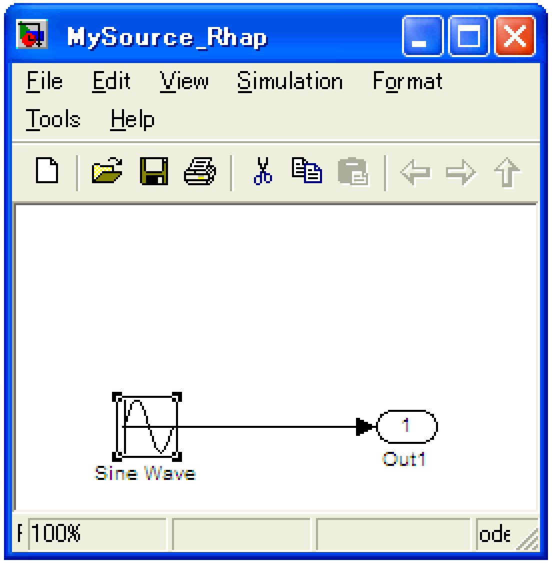


Source Block Parameters: Sine Wave

Amplitude:
amplitude

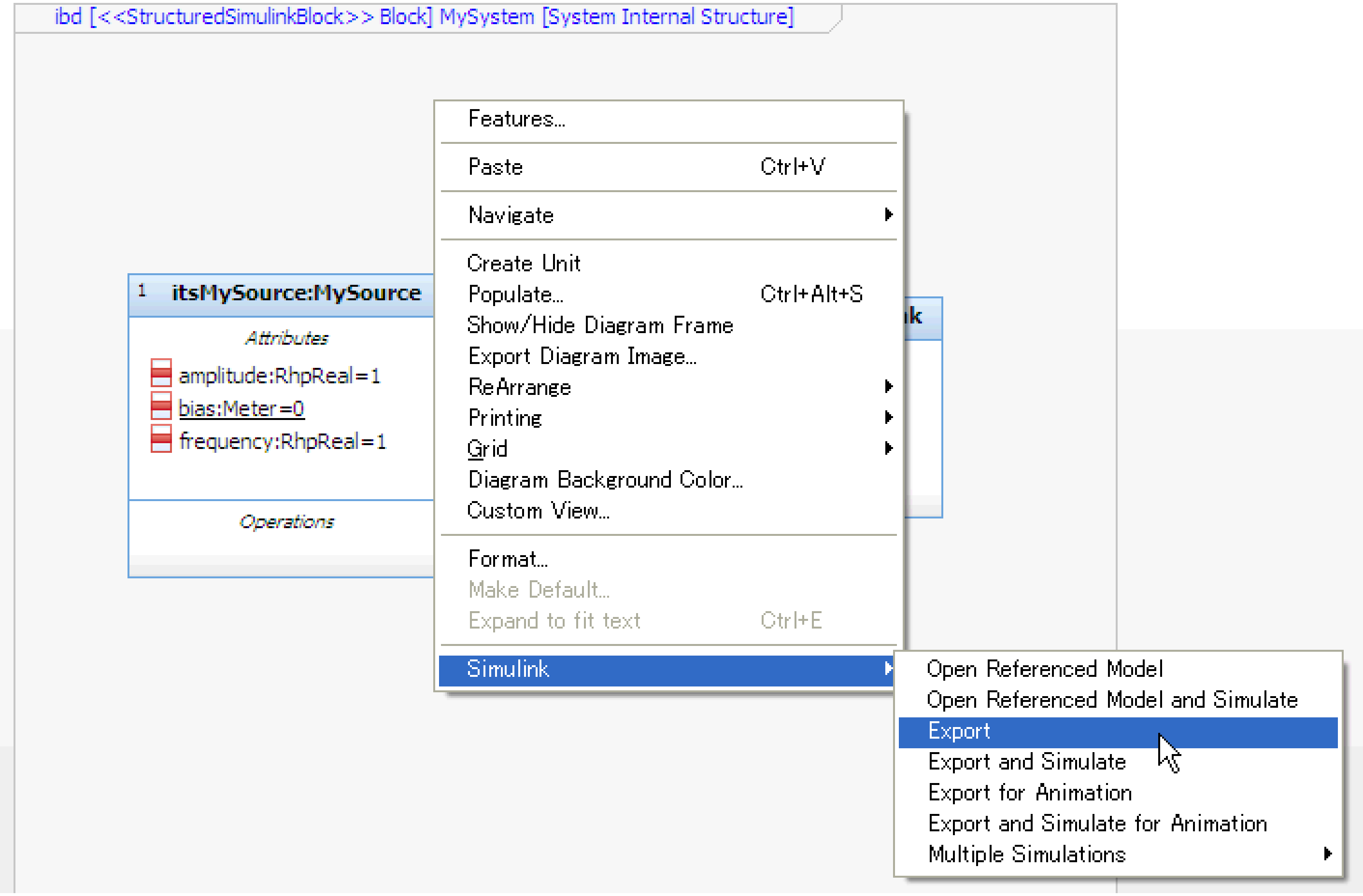
Bias:
bias

Frequency:
frequency



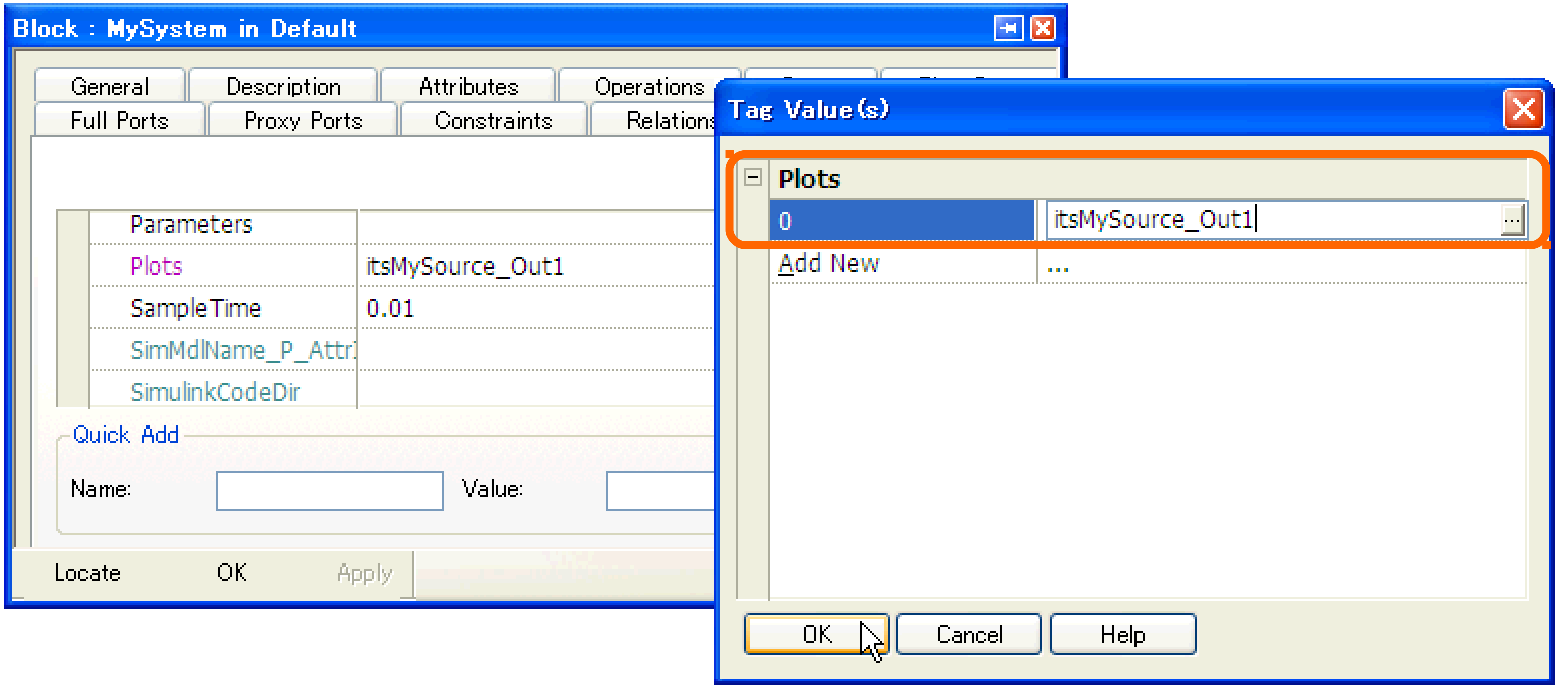
Exporting a Simulink model from MySystem block

- Export a Simulink model from MySystem block.



Setting a signal name to be plotted

- Set the Plots tag value of MySystem block to itsMySource_Out1.



Block : MySystem in Default

Parameters	
Plots	itsMySource_Out1
SampleTime	0.01
SimMdlName_P_Attr	
SimulinkCodeDir	

Quick Add

Name:

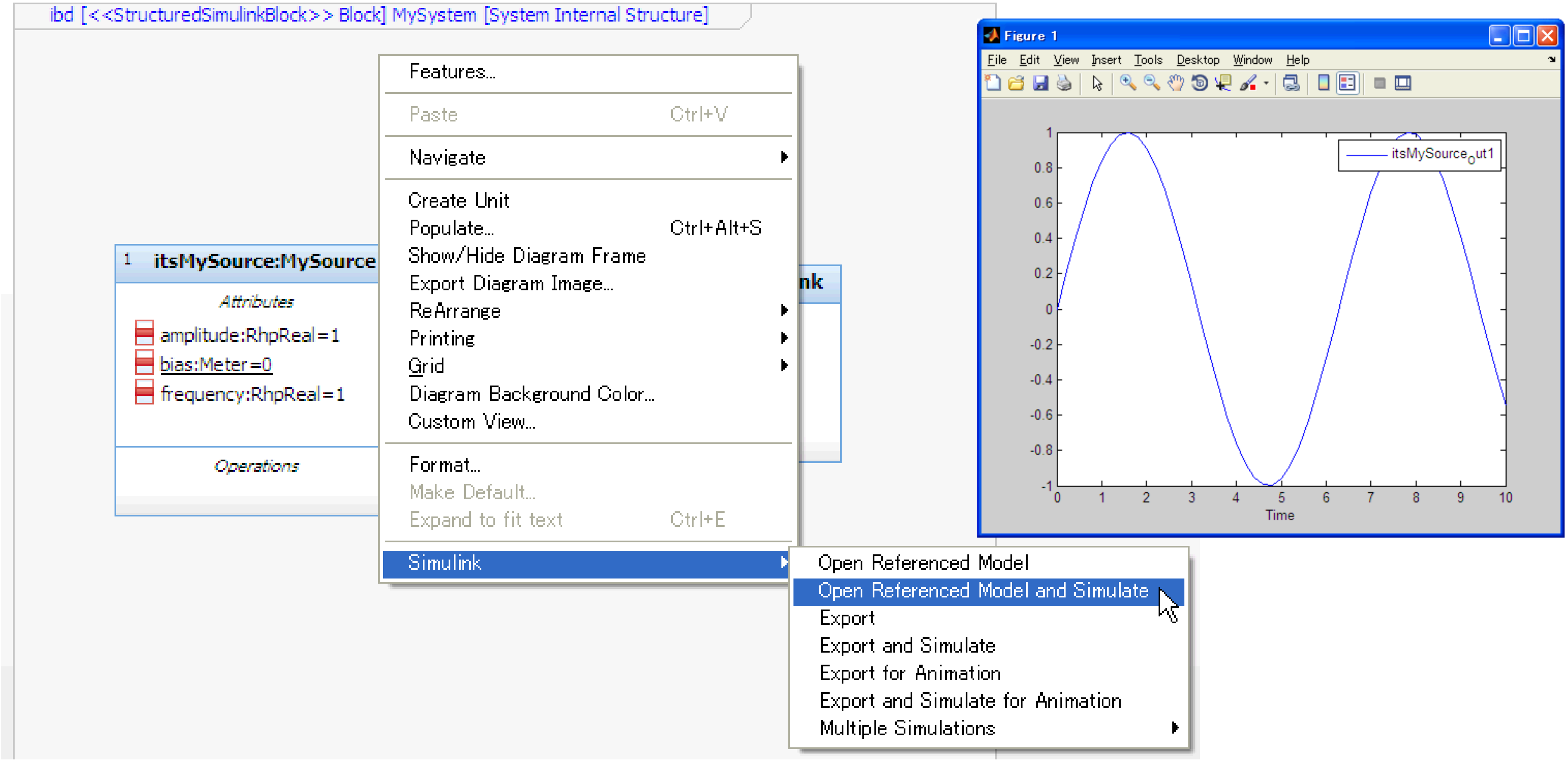
Value:

Tag Value(s)

Plots	
0	itsMySource_Out1
Add New	...

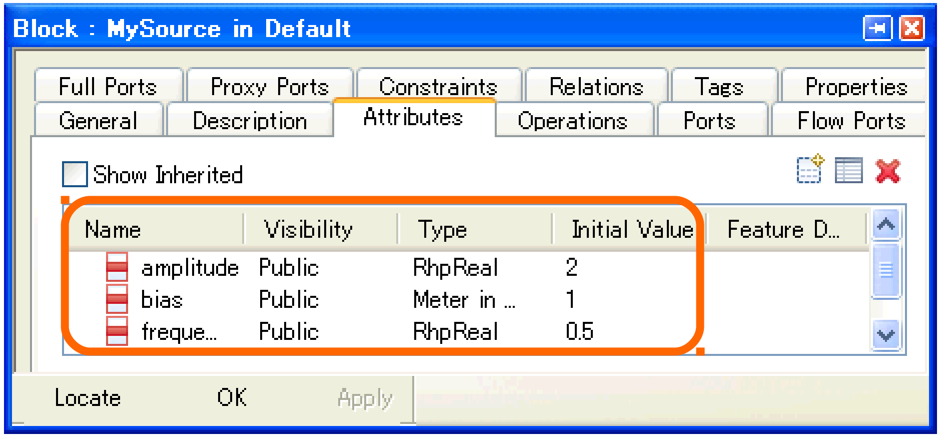
Executing simulation

- Right-click on the MySystem block’s IBD or the MySystem block on the browser, and select “Simulink”/”Open Referenced Model and Simulate” from the pop-up menu.
  - Figure of the specified signal is shown.

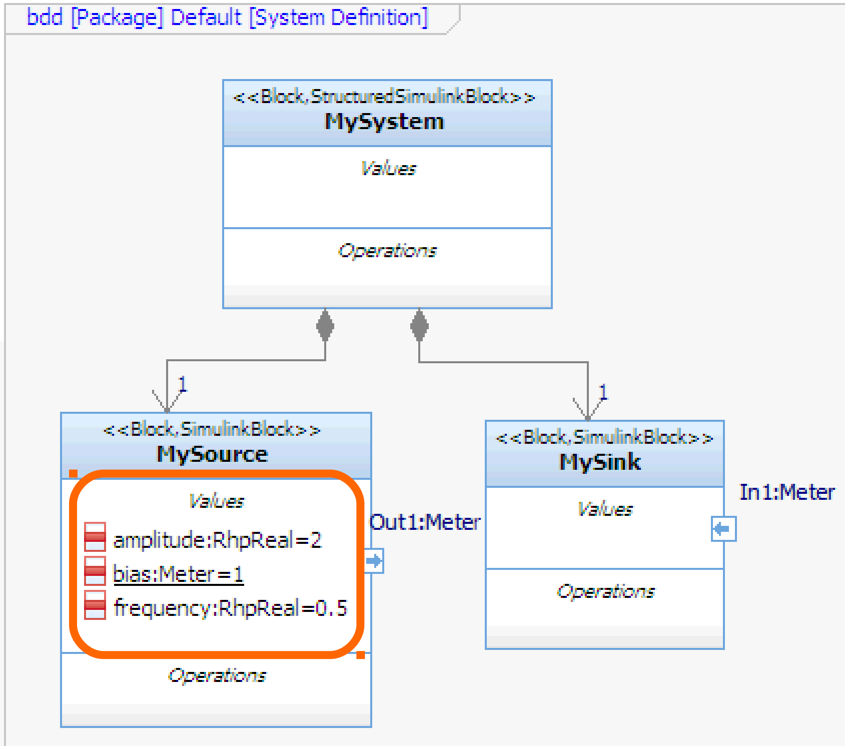


Changing values of MATLAB variables in Rhapsody

- Change values of MATLAB variables in Rhapsody.

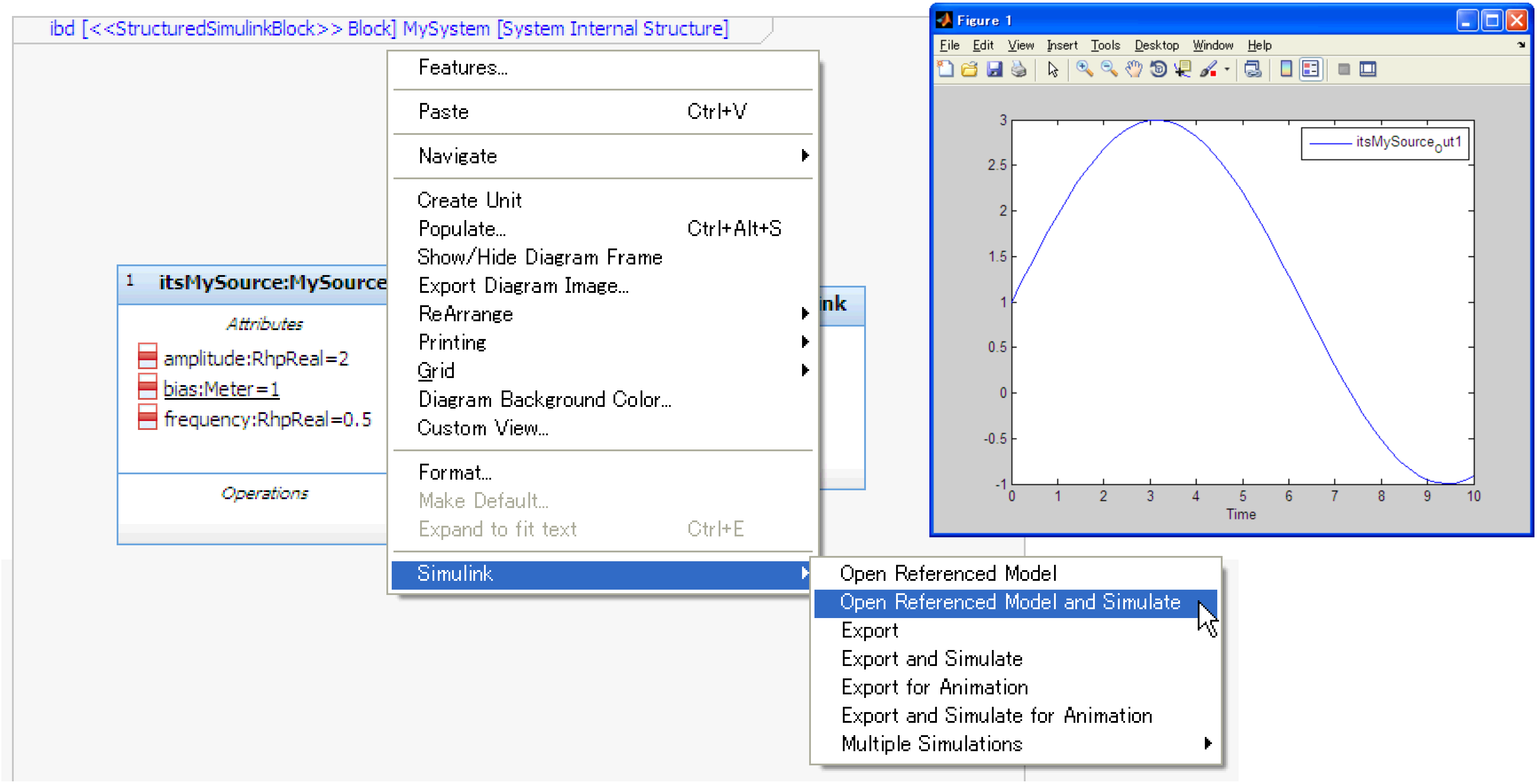


Name	Visibility	Type	Initial Value	Feature D...
amplitude	Public	RhpReal	2	
bias	Public	Meter in ...	1	
freque...	Public	RhpReal	0.5	



Executing simulation

- Changed values of MATLAB variables are used.



ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]

1 itsMySource:MySource

Attributes

amplitude:RhpReal=2

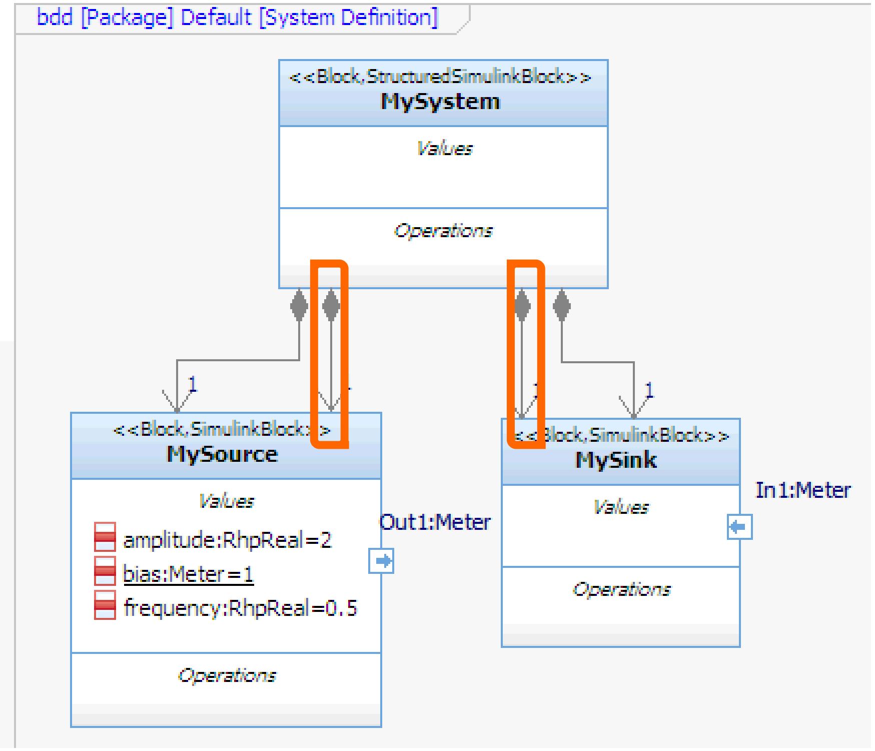
bias:Meter=1

frequency:RhpReal=0.5

Operations

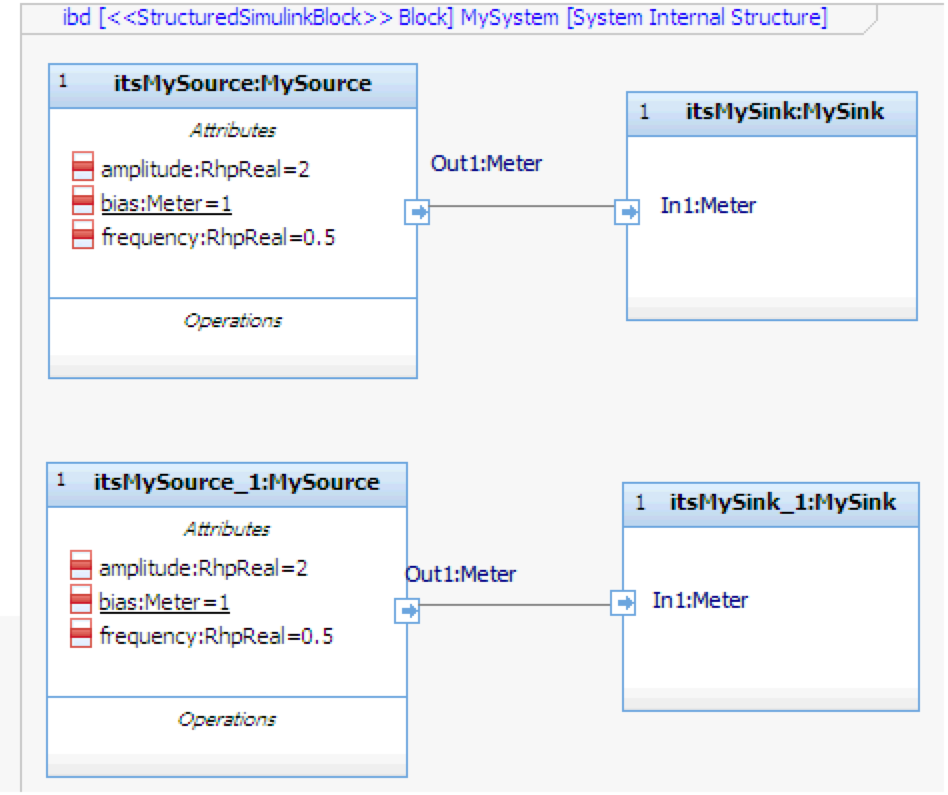
Changing the BDD

- Change the System Definition BDD as below to use multiple instances of Simulink sub-models:



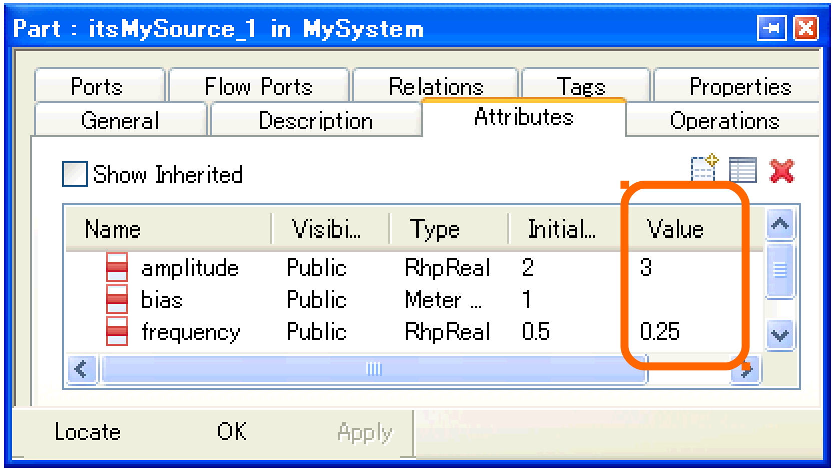
Changing the IBD

- Change the MySystem block's IBD as below:

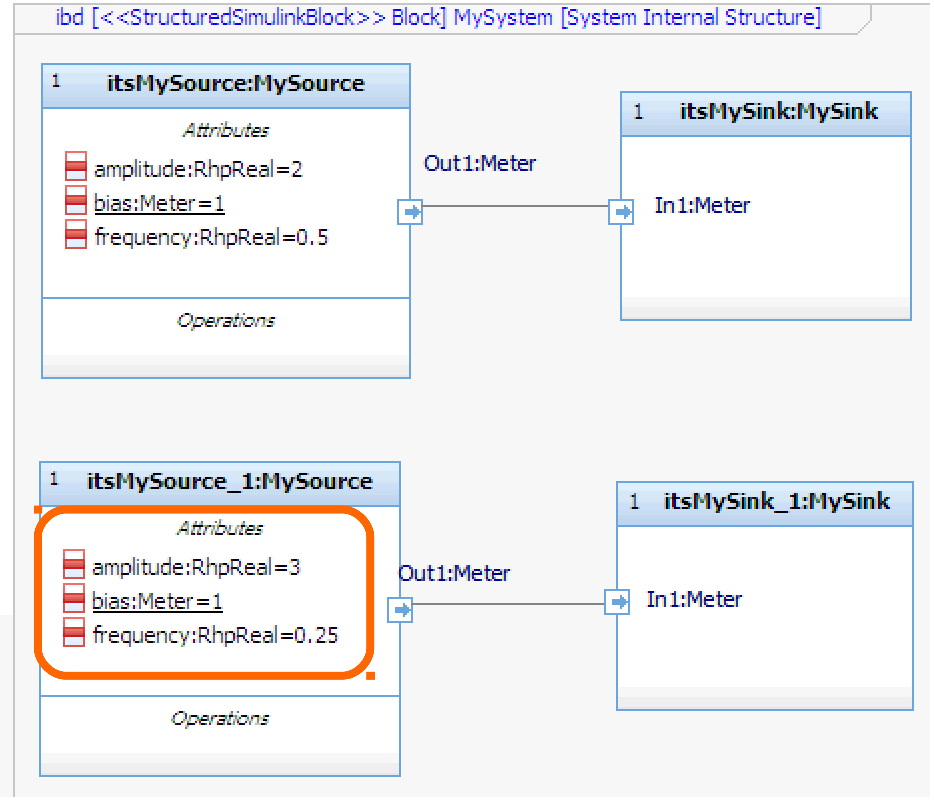


Changing the values of part attributes

- Double-click the itsMySource_1 part, and change the values of amplitude and frequency.
  - – Do not change the value of bias, because bias is static.



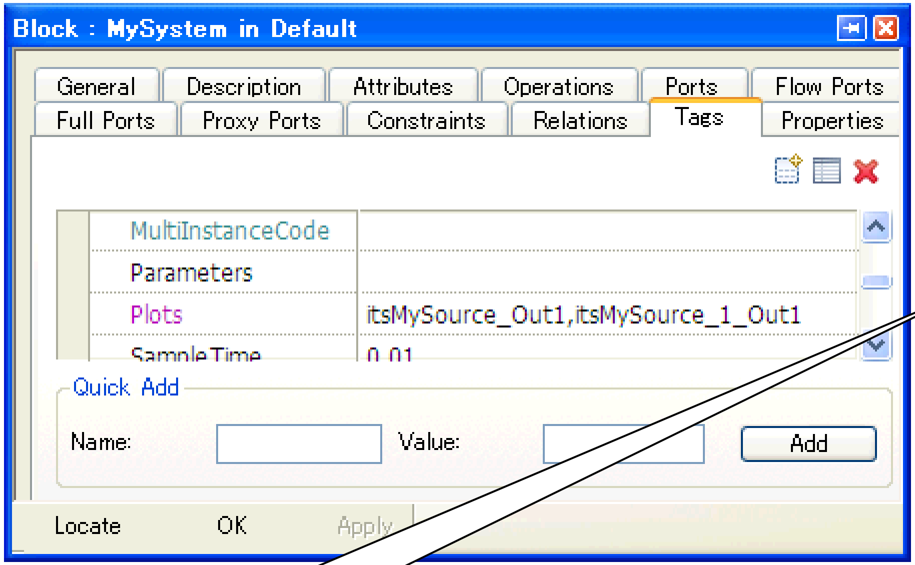
Name	Visibi...	Type	Initial...	Value
amplitude	Public	RhpReal	2	3
bias	Public	Meter ...	1	
frequency	Public	RhpReal	0.5	0.25



Adding a signal name to be plotted

- Add itsMySource_1_Out1 to the Plots tag value of MySystem block.

Two figures



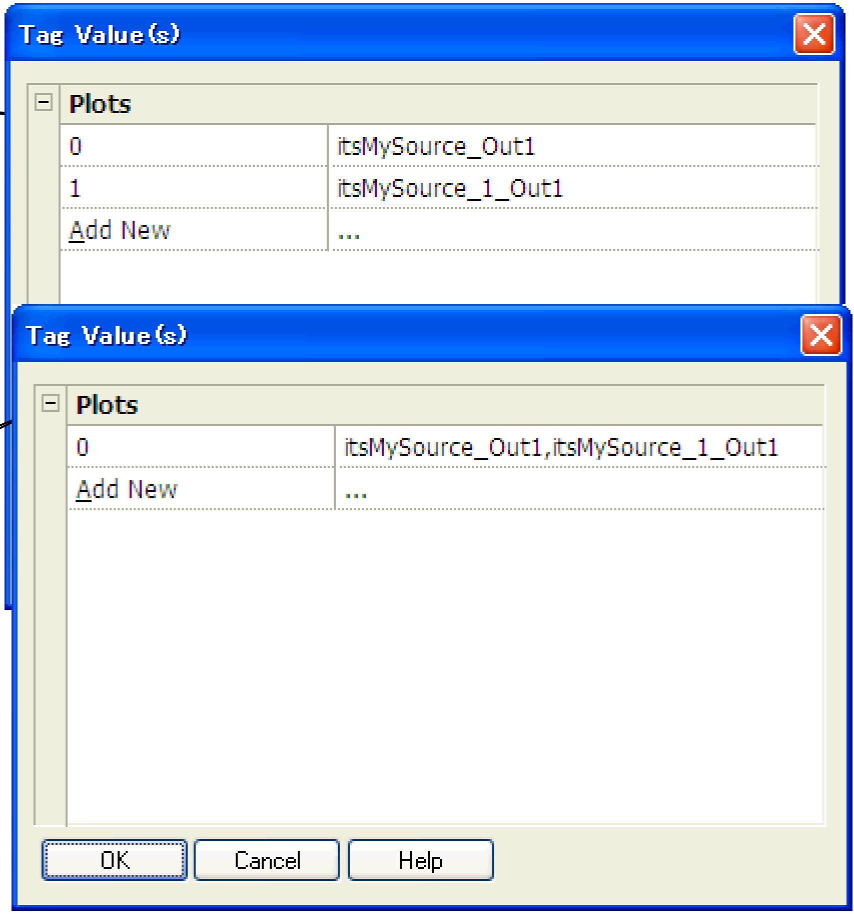
Block : MySystem in Default

MultiInstanceCode	
Parameters	
Plots	itsMySource_Out1,itsMySource_1_Out1
SampleTime	0.01

Quick Add

Name: Value:

One figure



Tag Value(s)

Plots

0	itsMySource_Out1
1	itsMySource_1_Out1
Add New	...

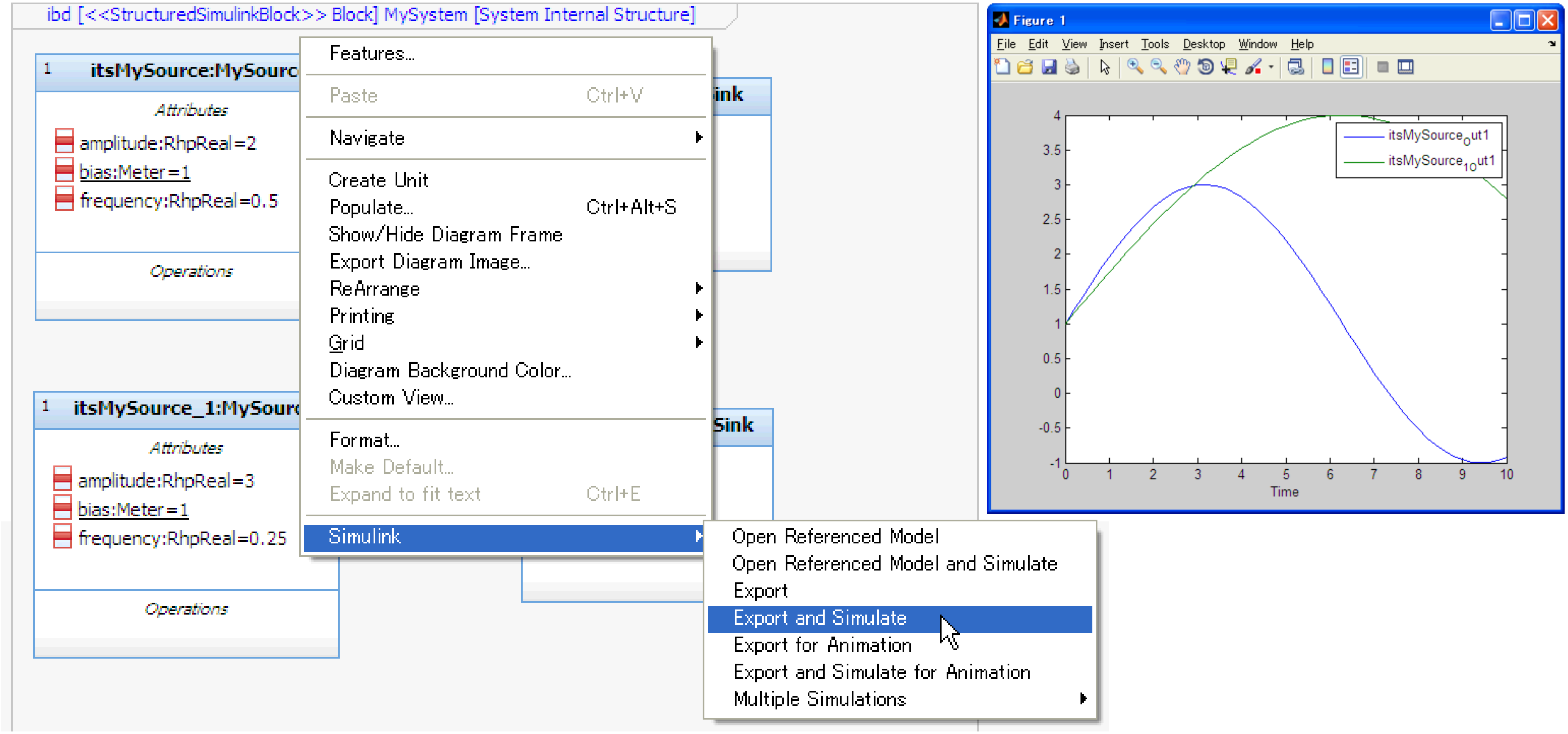
Tag Value(s)

Plots

0	itsMySource_Out1,itsMySource_1_Out1
Add New	...

Executing simulation

- Each instance of MySource_Rhap uses different values of MATLAB variables.



ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]

1 itsMySource:MySourc

Attributes

amplitude:RhpReal=2

bias:Meter=1

frequency:RhpReal=0.5

Operations

1 itsMySource_1:MySourc

Attributes

amplitude:RhpReal=3

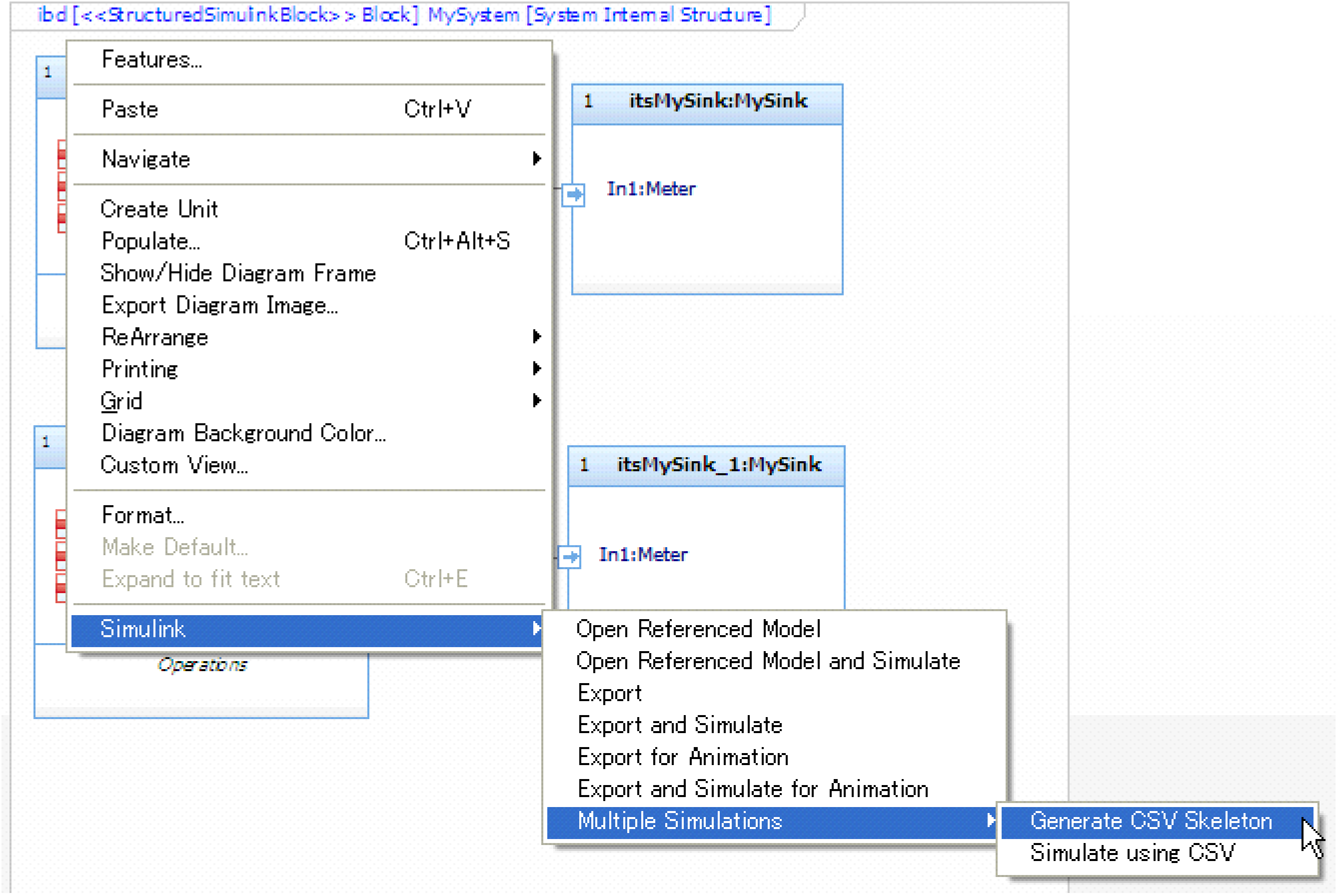
bias:Meter=1

frequency:RhpReal=0.25

Operations

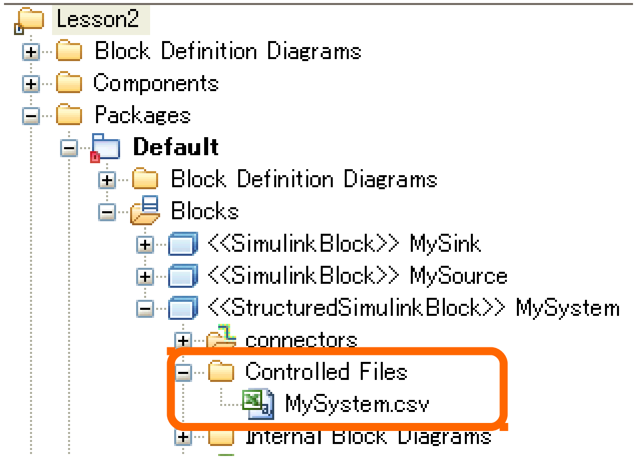
Generating CSV skeleton file

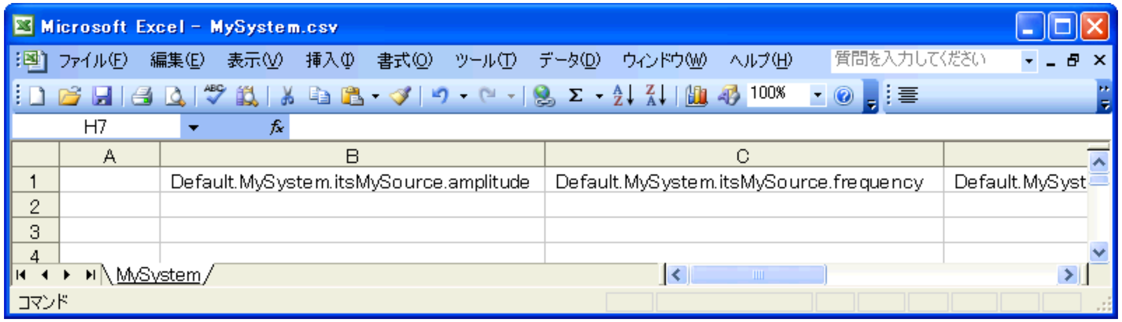
- We can simulate multiple times by changing the values of MATLAB variables if we use MATLAB/Simulink R2011b (?) or later.
- Right-click on the MySystem block’s IBD or the MySystem block on the browser, and select “Simulink”/”Multiple Simulations”/”Generate CSV Skeleton” from the pop-up menu.



Generated CSV skeleton file

- A CSV file is generated as a controlled file of MySystem block.
- Generated CSV skeleton contains the following columns:
  - Default.MySystem.itsMySource.amplitude
  - Default.MySystem.itsMySource.frequency
  - Default.MySystem.itsMySource_1.amplitude
  - Default.MySystem.itsMySource_1.frequency
  - Default.MySource.amplitude
  - Default.MySource.bias
  - Default.MySource.frequency




Microsoft Excel - MySystem.csv
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)
質問を入力してください
100%
H7
MySystem
コマンド

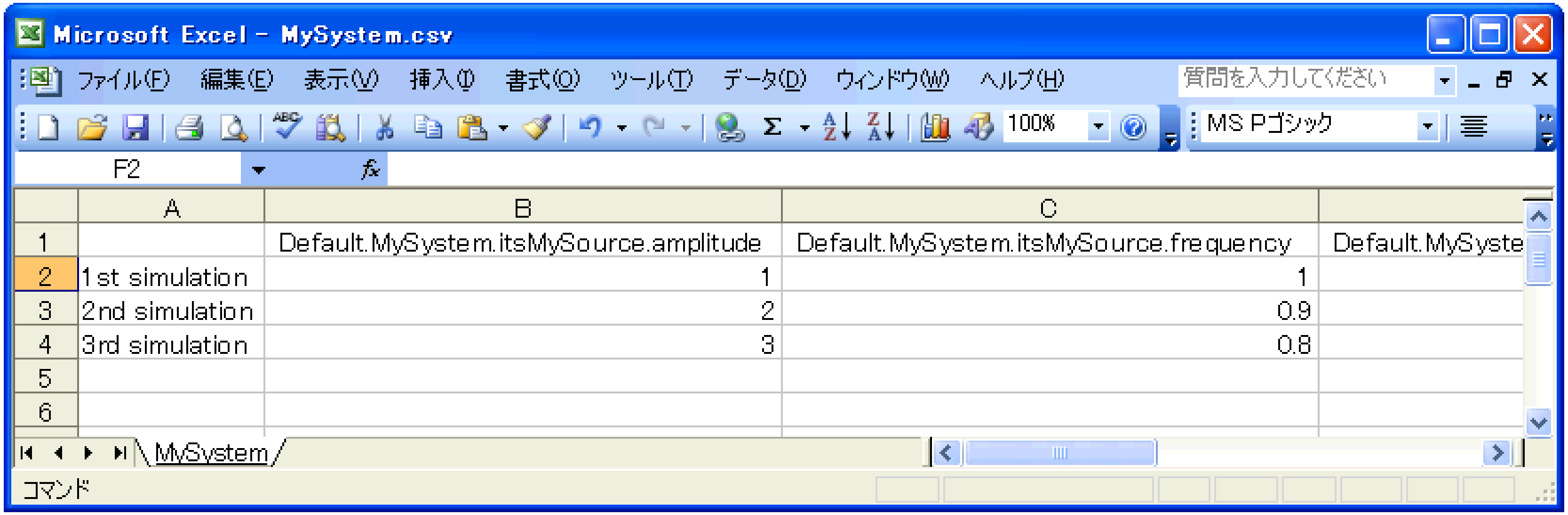
	A	B	C	
1		Default.MySystem.itsMySource.amplitude	Default.MySystem.itsMySource.frequency	Default.MySyst
2				
3				
4				

Values to be used for simulation

- Part’s attribute value:
  - If Part attribute’s value is specified in the CSV file, the value will be used.
  - Else if Part attribute’s value is specified in the Rhapsody model, the value will be used.
  - Else if Block attribute’s initial value is specified in the CSV file, the value will be used.
  - Else if Block attribute’s initial value is specified in the Rhapsody model, the value will be used.
  - Else, 0 will be used.
- Block’s attribute value:
  - If Block attribute’s initial value is specified in the CSV file, the value will be used.
  - Else if Block attribute’s initial value is specified in the Rhapsody model, the value will be used.
  - Else, 0 will be used.

Changing the CSV file

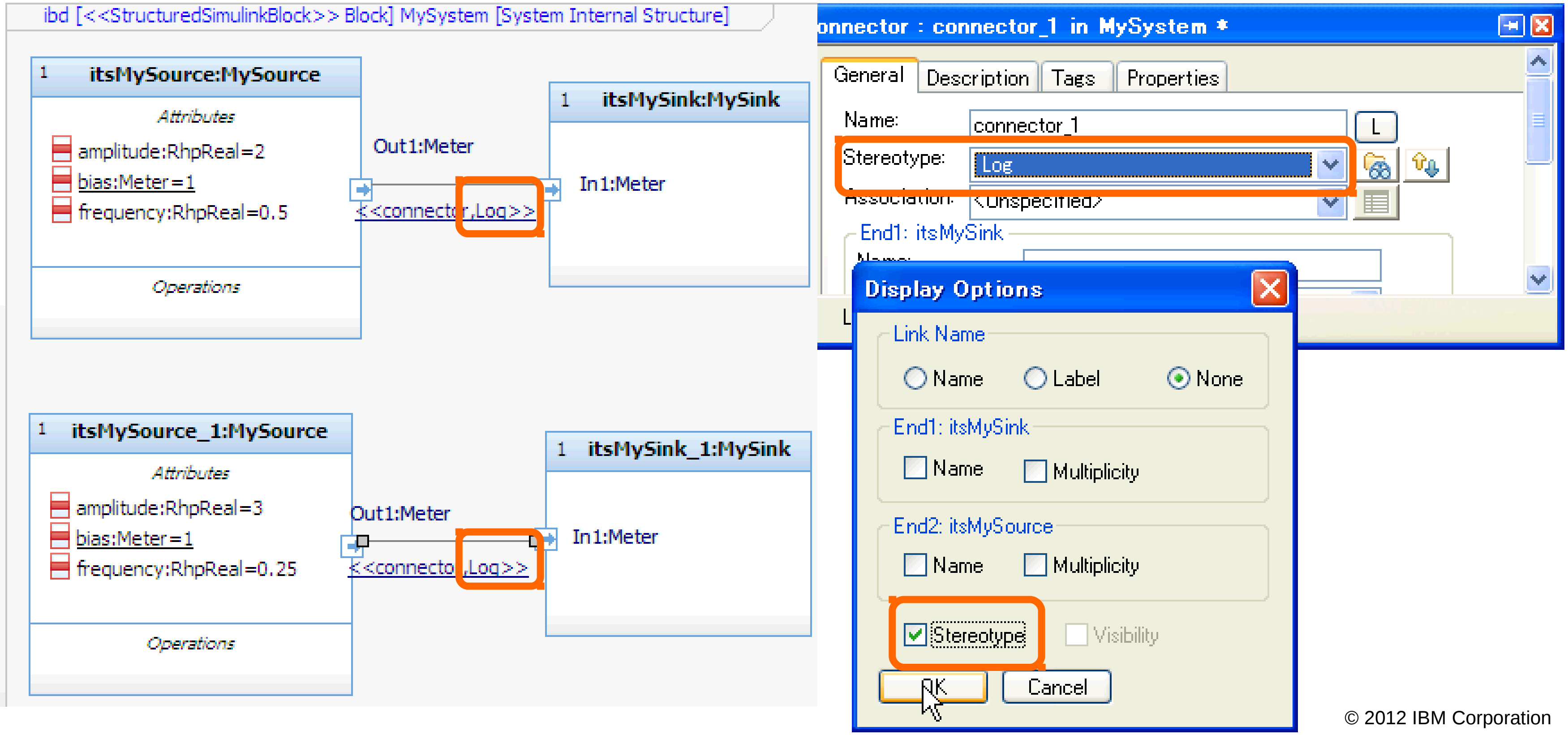
- First column is Simulation name
- Each simulation is specified in each row


Microsoft Excel - MySystem.csv
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)
質問を入力してください
100%
MS Pゴシック
F2
fx
MySystem
コマンド

	A	B	C	
1		Default.MySystem.itsMySource.amplitude	Default.MySystem.itsMySource.frequency	Default.MySyste
2	1st simulation	1	1	
3	2nd simulation	2	0.9	
4	3rd simulation	3	0.8	
5				
6				

Applying <<Log>> stereotype and export

- Apply <<Log>> stereotype to connectors to be logged.
  - We can see stereotypes by changing Display Options.
- Export a Simulink model from MySystem block.



onnector : connector_1 in MySystem *

General | Description | Tags | Properties

Name: connector_1

Stereotype: Log

Association: <Unspecified>

End1: itsMySink

Display Options

Link Name

○ Name ○ Label ● None

End1: itsMySink

□ Name □ Multiplicity

End2: itsMySource

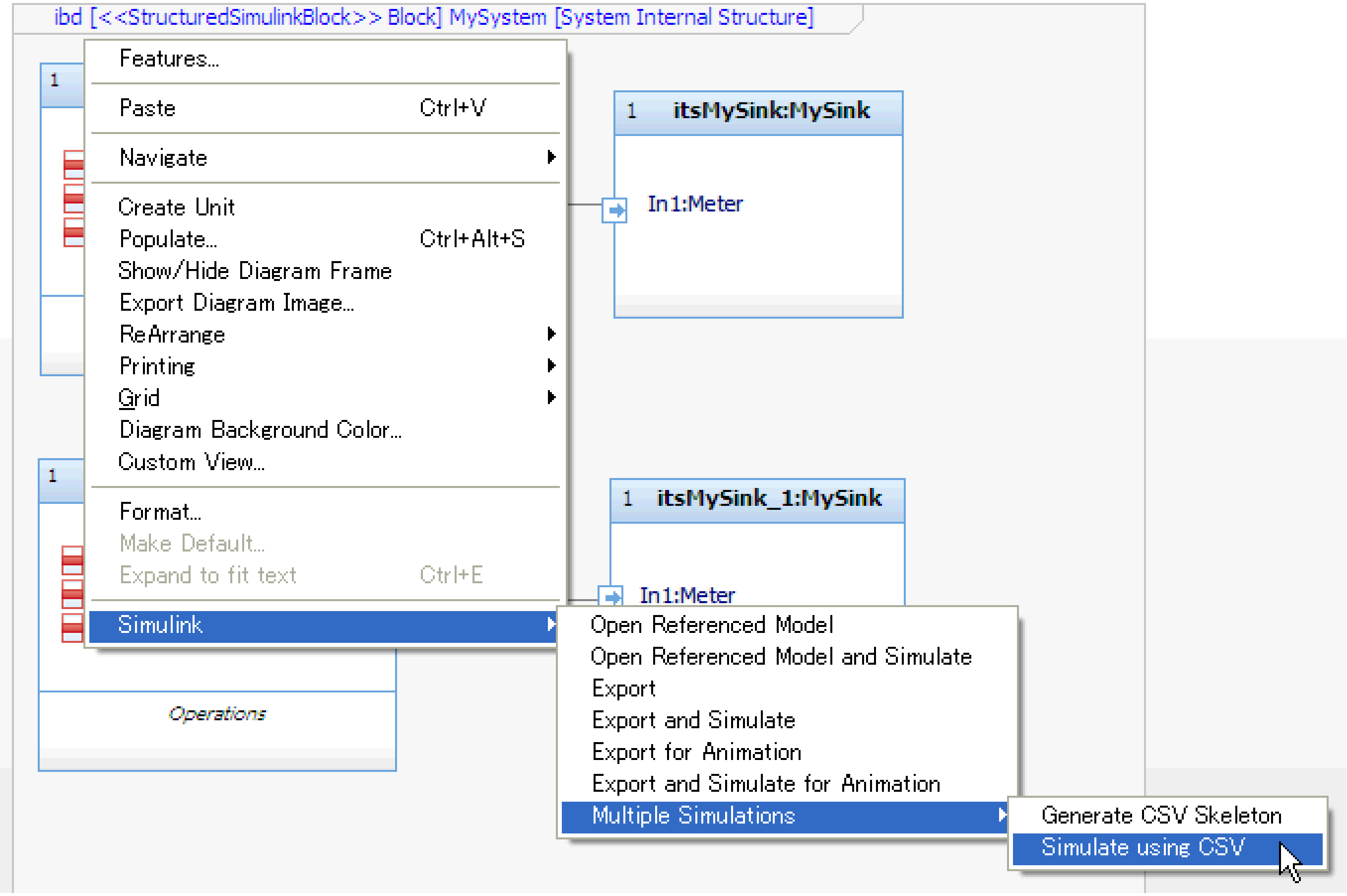
□ Name □ Multiplicity

☑ Stereotype □ Visibility

OK Cancel

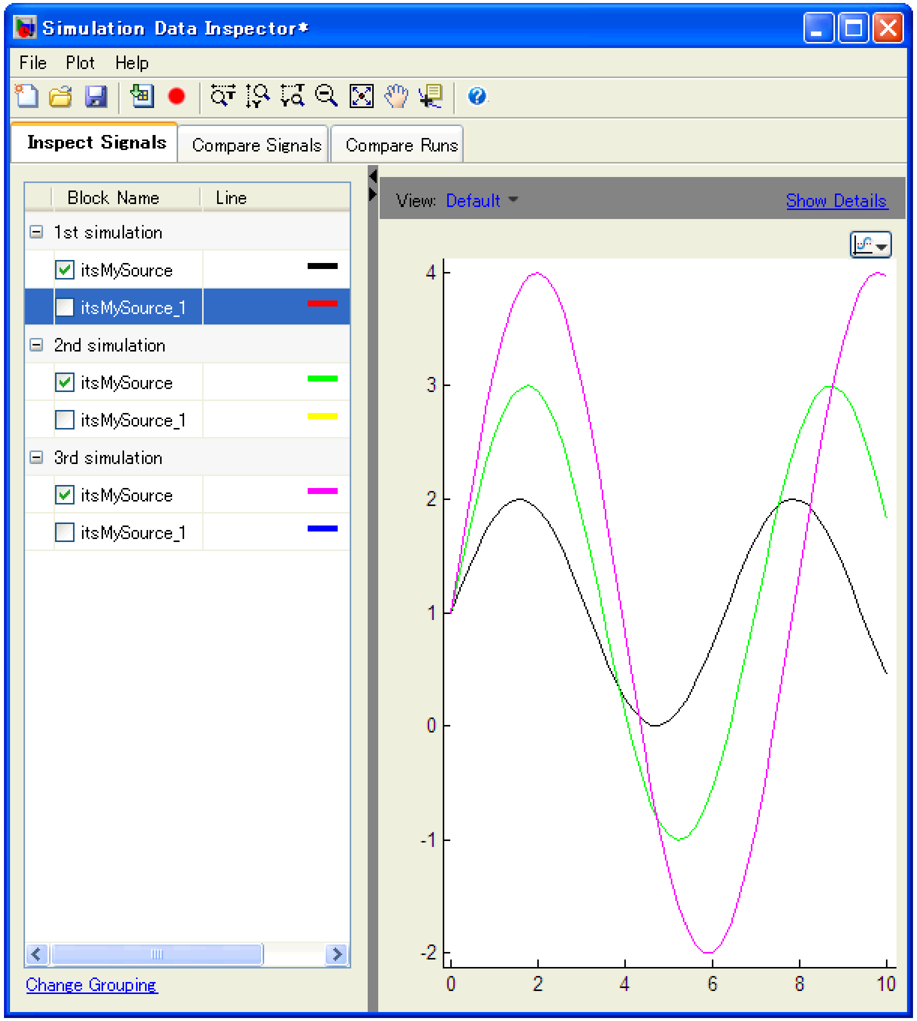
Simulating using CSV

- Right-click on the MySystem block's IBD or the MySystem block on the browser, and select "Simulink"/"Multiple Simulations"/"Simulate using CSV" from the pop-up menu.

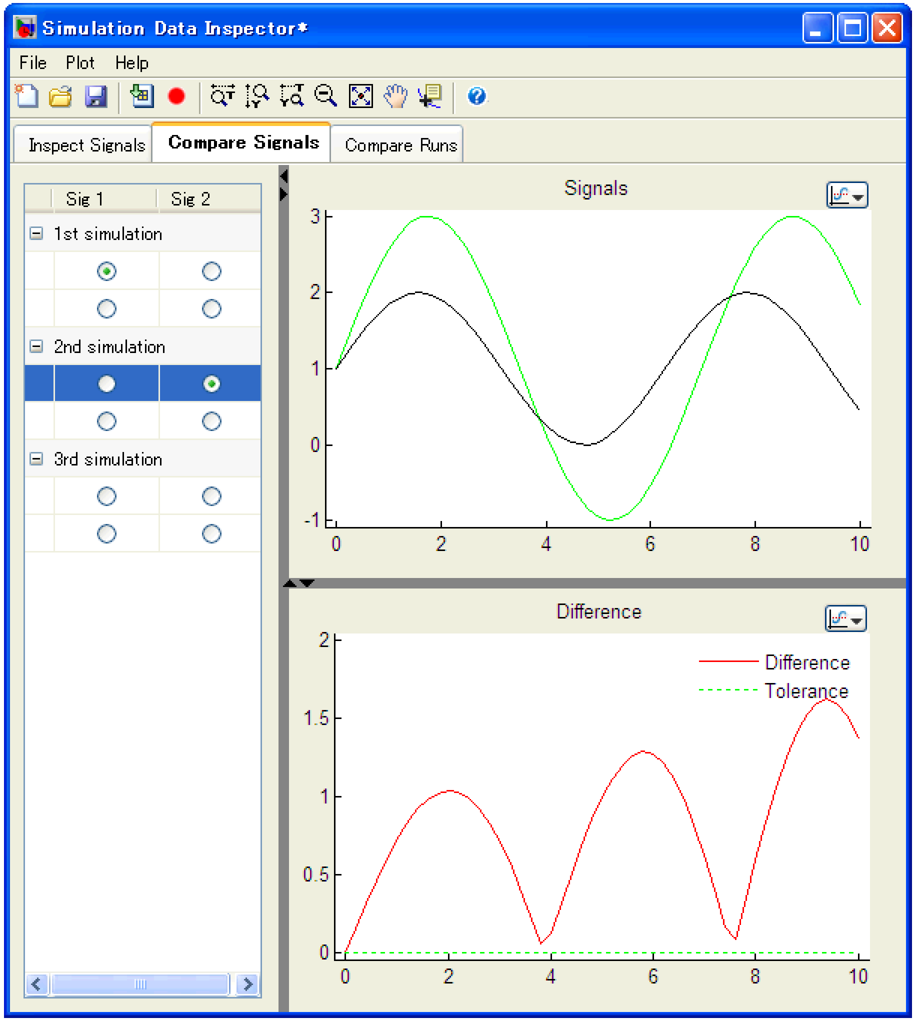


Using Simulink Data Inspector

- Perform “Inspect Signals” and “Compare Signals.”



Block Name	Line
⊟ 1st simulation	
☑ itsMySource	⬛
☐ itsMySource_1	🟥
⊟ 2nd simulation	
☑ itsMySource	🟩
☐ itsMySource_1	🟨
⊟ 3rd simulation	
☑ itsMySource	🟪
☐ itsMySource_1	🟦

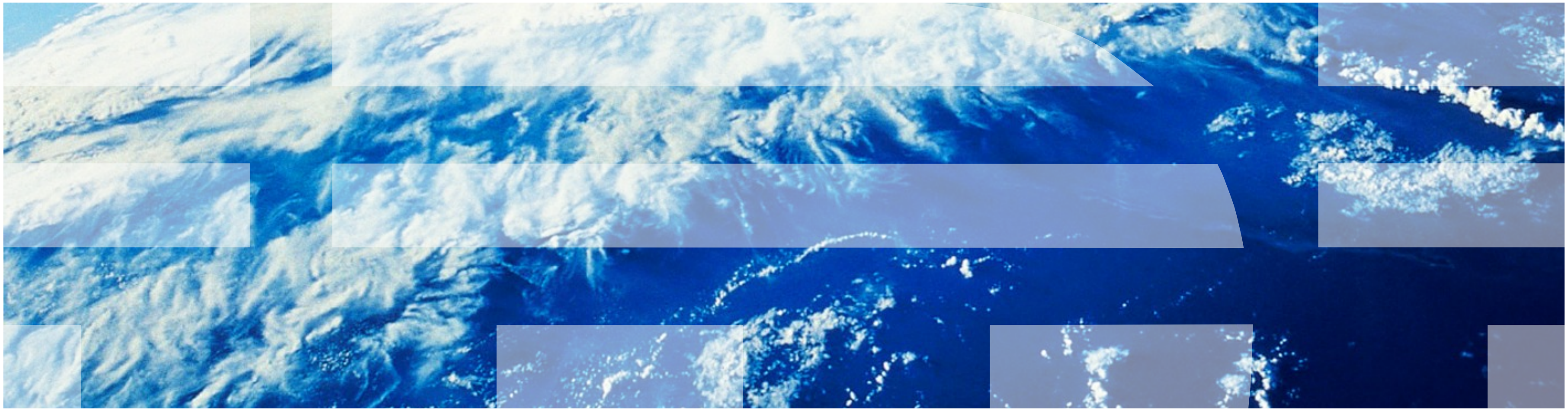


Sig 1	Sig 2
⊟ 1st simulation	
●	○
○	○
⊟ 2nd simulation	
○	●
○	○
⊟ 3rd simulation	
○	○
○	○

Saving the project

- Save the project.

Lesson 3: Using a bus object and an enumeration

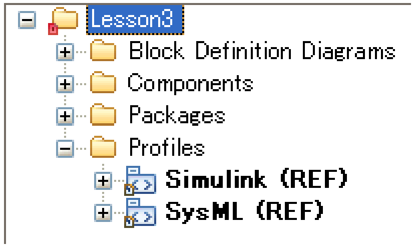


Contents of this lesson

- Create a model containing Simulink sub-models, a bus object, and an enumeration.
- Export templates of Simulink sub-models.
- Edit Simulink sub-models.
- Export a Simulink model, and simulate it.

Preparations

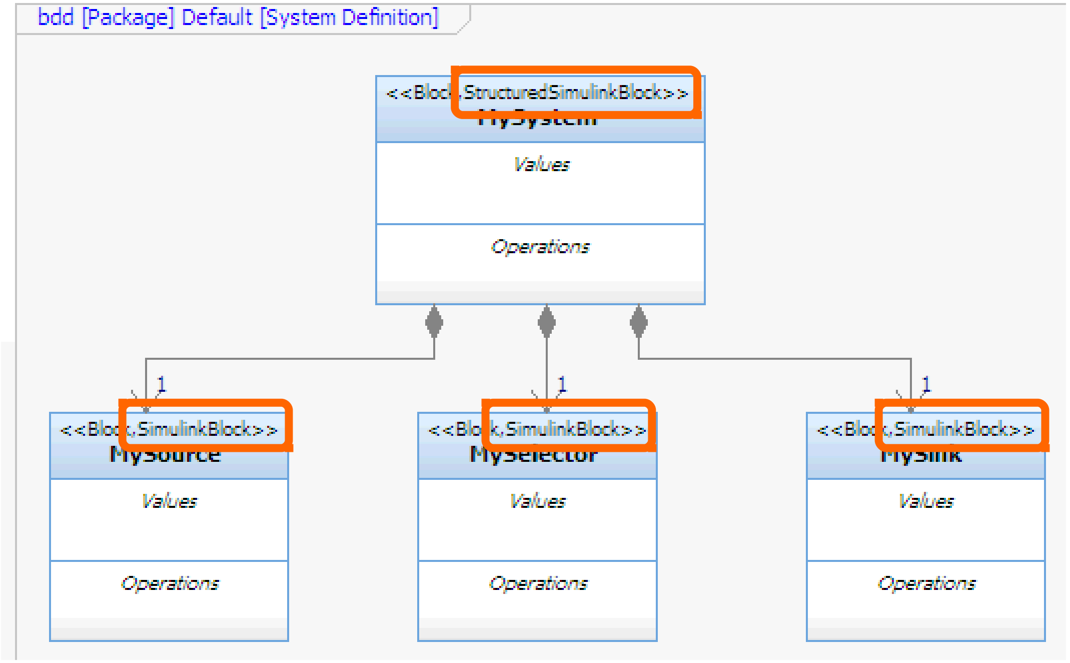
- Start Rhapsody
- Create a SysML project Lesson3 in C:\Work\.
- Add the Simulink profile to the model.



- Lesson3
  - Block Definition Diagrams
  - Components
  - Packages
  - Profiles
    - **Simulink (REF)**
    - **SysML (REF)**

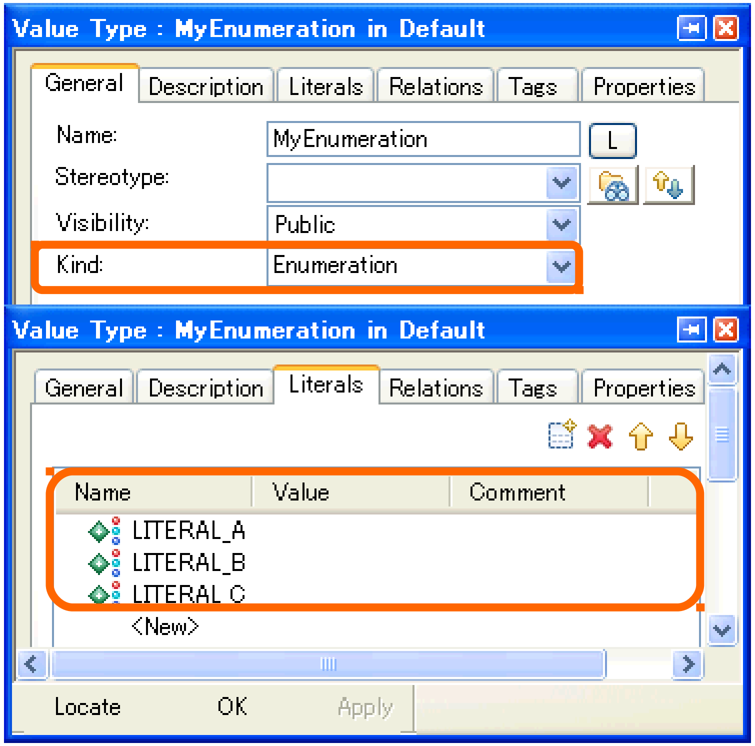
Creating a Block Definition Diagram (BDD)

- Create a BDD under the Default package and place MySystem, MySource, MySelector, and MySink blocks.
  - Apply <<StrucgturedSimulinkBlock>> to MySystem block, and <<SimulinkBlock>> stereotypes to MySource, MySelector, and MySink blocks.



Adding an enumeration

- Add a Value Type MyEnmeration, and change the Kind to Enumeration.
- Add three literals, LITERAL_A, LITERAL_B, and LITERAL_C.

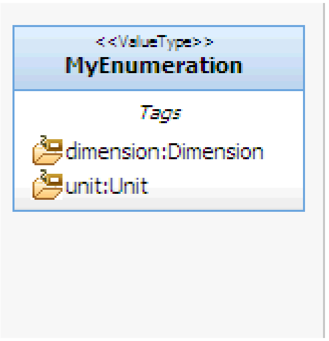


Value Type : MyEnumeration in Default

Name:	MyEnumeration
Stereotype:	
Visibility:	Public
Kind:	Enumeration

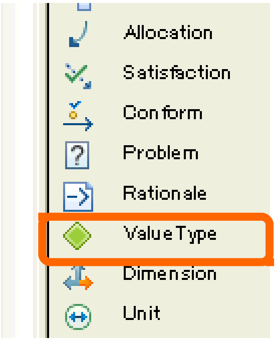
Value Type : MyEnumeration in Default

Name	Value	Comment
LITERAL_A		
LITERAL_B		
LITERAL_C		
<New>		



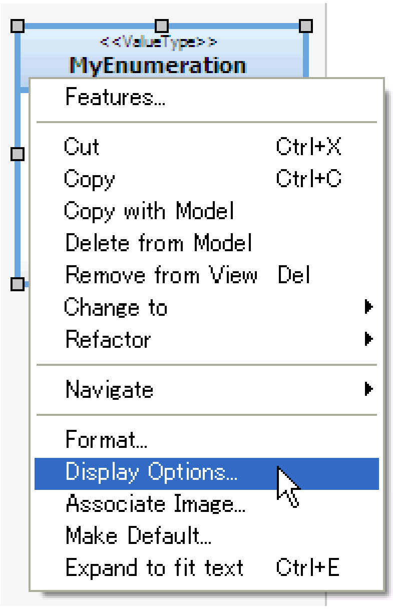
<<ValueType>>
MyEnumeration

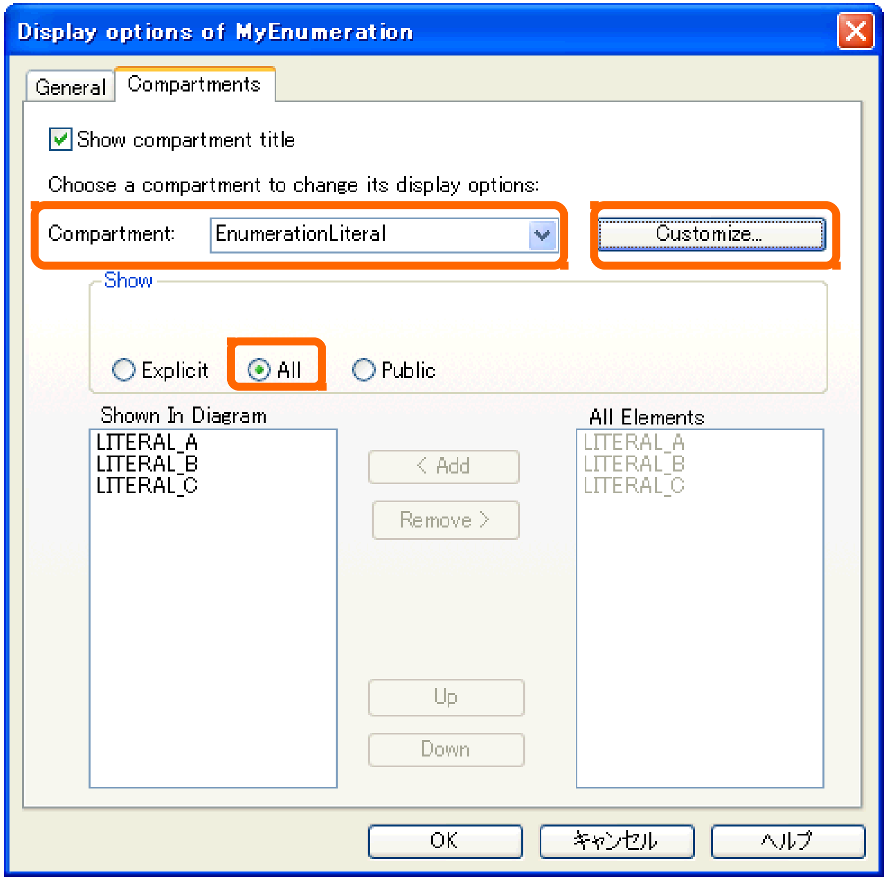
Tags
dimension:Dimension
unit:Unit



Changing the enumeration’s display options

- Right-click on the MyEnumeration Value Type, and select “Display Options …”.





Display options of MyEnumeration

General | Compartments

☑ Show compartment title

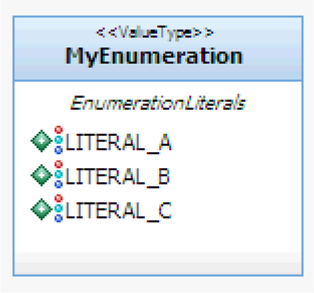
Choose a compartment to change its display options:

Compartment: EnumerationLiteral

Show: ○ Explicit ● All ○ Public

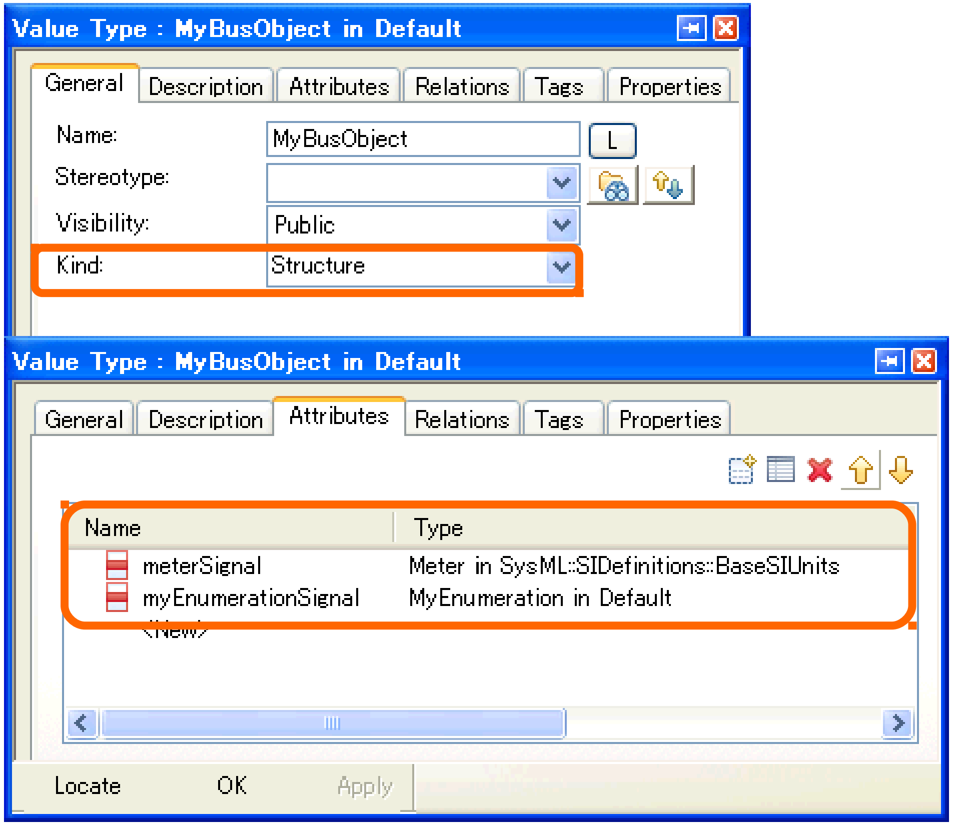
Shown In Diagram: LITERAL_A, LITERAL_B, LITERAL_C

All Elements: LITERAL_A, LITERAL_B, LITERAL_C



Adding a bus object

- Add a Value Type MyBusObject, and change the Kind to Enumeration.
- Add two attributes, meterSignal and myEnumerationSignal



Value Type : MyBusObject in Default

Name: MyBusObject

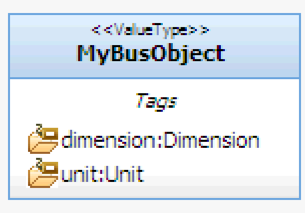
Stereotype:

Visibility: Public

Kind: Structure

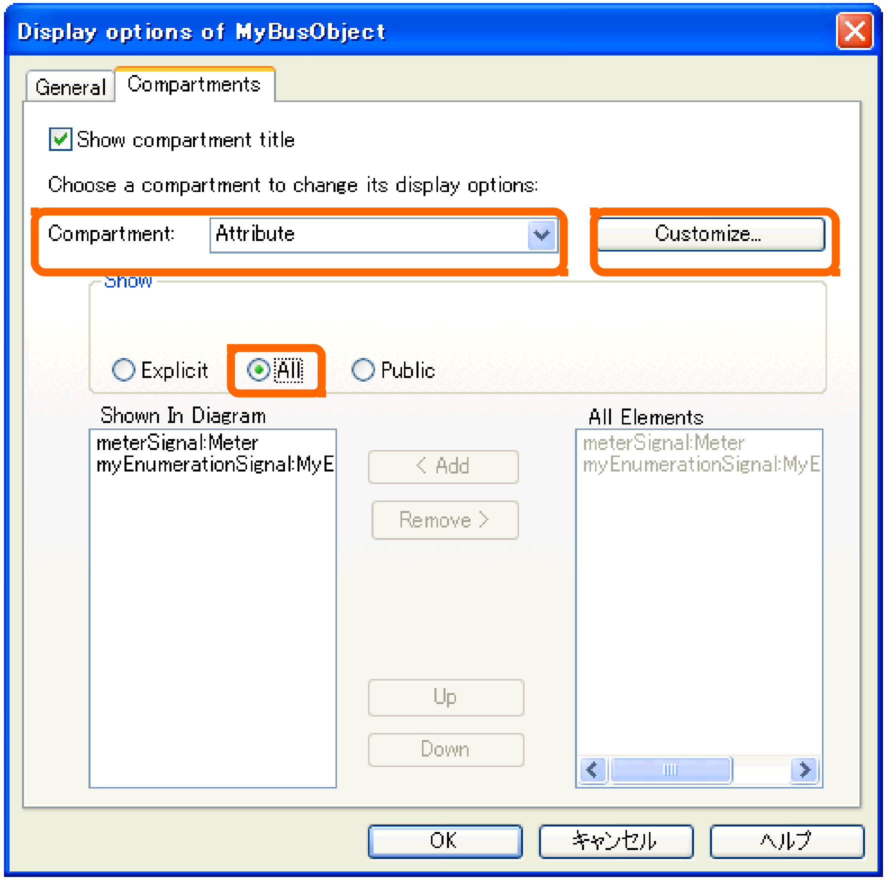
Value Type : MyBusObject in Default

Name	Type
meterSignal	Meter in SysML::SIDefinitions::BaseSIUnits
myEnumerationSignal	MyEnumeration in Default
<New>	



Changing the bus object's display options

- Right-click on the MyBusObject Value Type, and select "Display Options …".



Display options of MyBusObject

General | Compartments

☑ Show compartment title

Choose a compartment to change its display options:

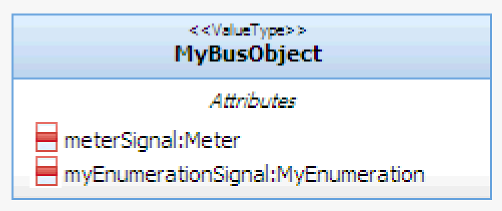
Compartment: Attribute

Customize...

Show: ○ Explicit ● All ○ Public

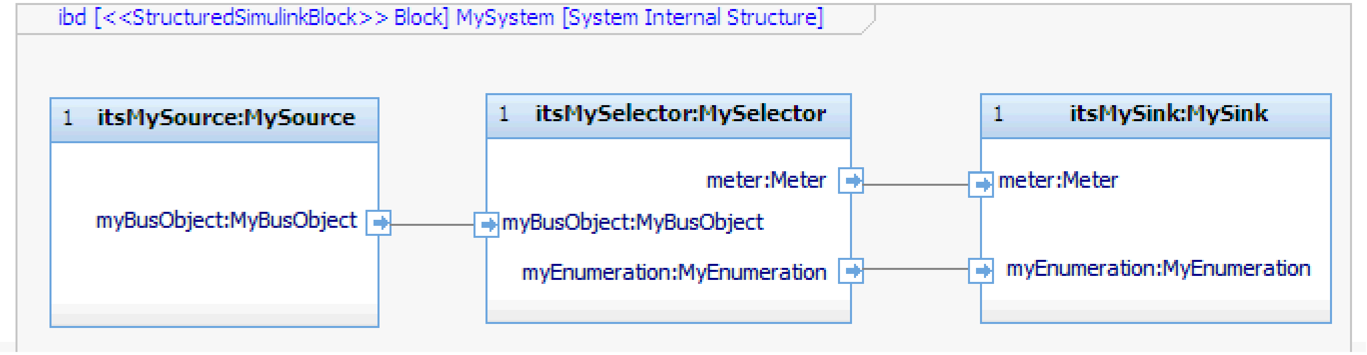
Shown In Diagram:
meterSignal:Meter
myEnumerationSignal:MyE

All Elements:
meterSignal:Meter
myEnumerationSignal:MyE

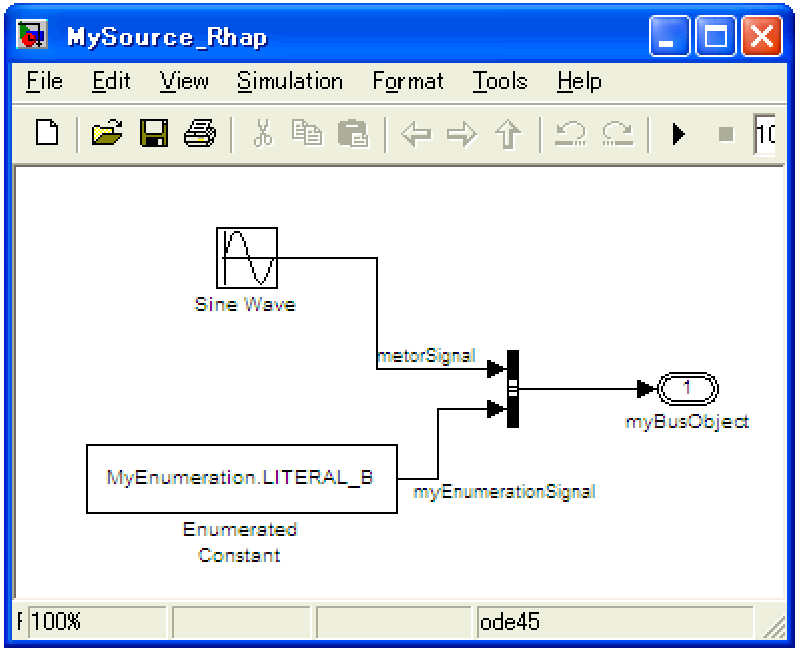


Creating an IBD of MySystem block

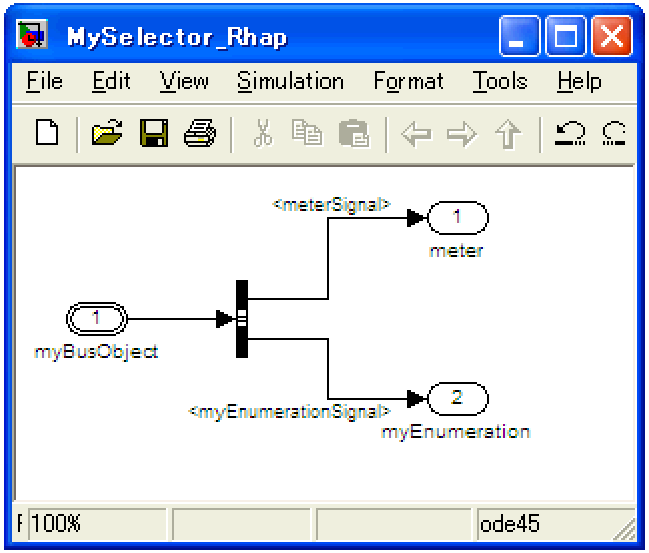
- Create an IBD of MySystem block as below.


ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]
1 itsMySource:MySource
myBusObject:MyBusObject
1 itsMySelector:MySelector
meter:Meter
myBusObject:MyBusObject
myEnumeration:MyEnumeration
1 itsMySink:MySink
meter:Meter
myEnumeration:MyEnumeration

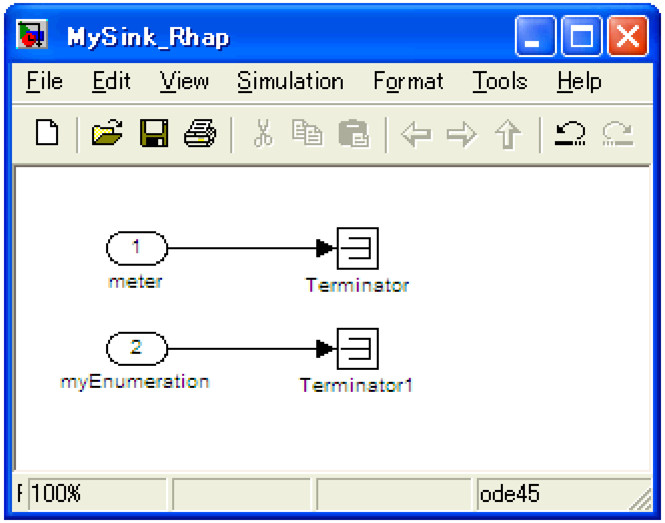
Exporting a Simulink model template from MySource block, and editing it



Exporting a Simulink model template from MySelector block, and editing it



Exporting a Simulink model template from MySink block, and editing it

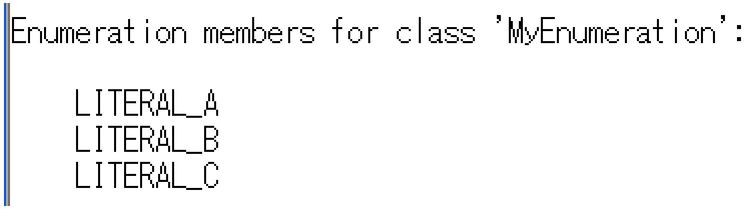


Checking the generated enumeration

- Open MyEnumeration.m in C:\Work\Lesson3\Simulink folder using a text editor.

```
%
% MATLAB commands (type definition) generated by Rhapsody Control System Designer
% Model Name: Lesson3
% Enumeration Name: MyEnumeration
% Generated Date: Wed Sep 05 15:42:41 JST 2012
%
classdef MyEnumeration < Simulink.IntEnumType
        enumeration
                LITERAL_A(0)
                LITERAL_B(1)
                LITERAL_C(2)
        end
end
```

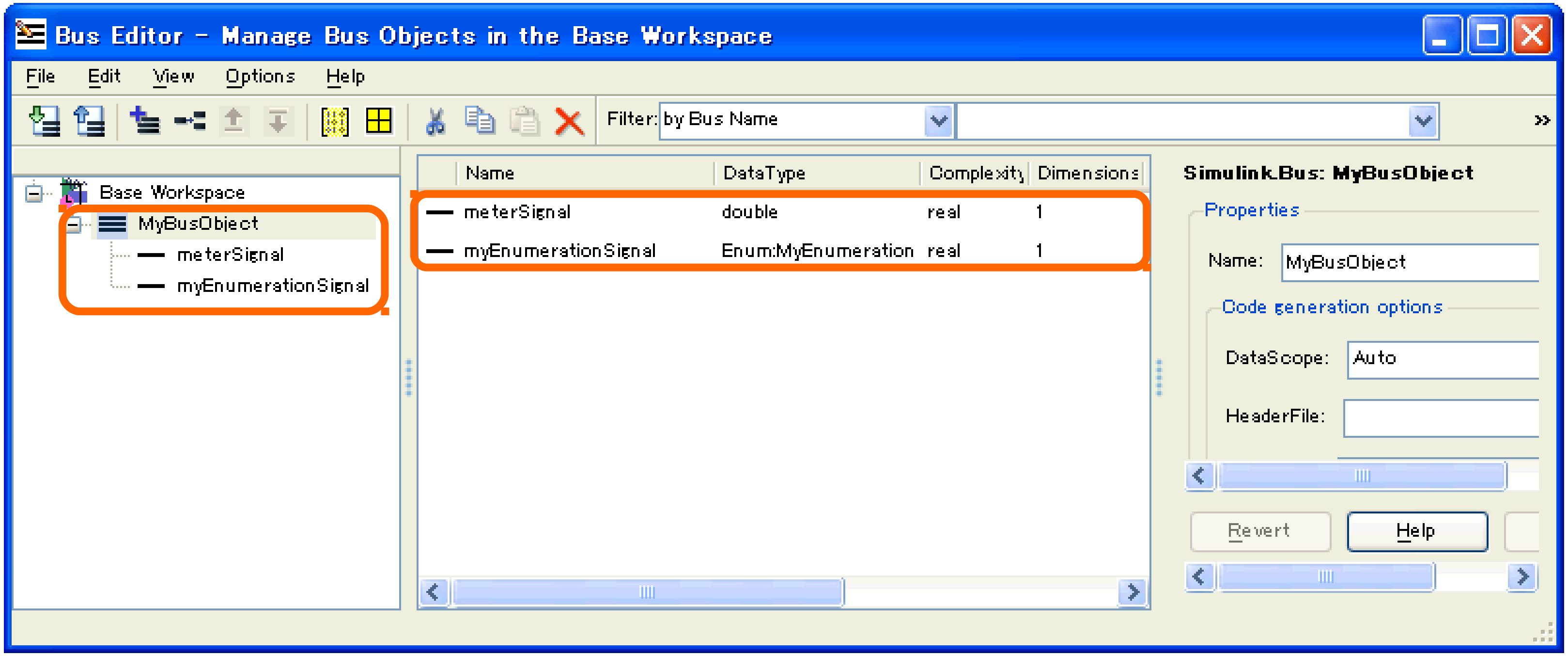
- Type “enumeration MyEnumeration” on the MATLAB window.
  – You can see the enumeration members.



```
Enumeration members for class 'MyEnumeration':

    LITERAL_A
    LITERAL_B
    LITERAL_C
```

Checking the generated bus object

- Open Simulink Bus Editor
  - If you cannot see the MyBusObject on the Bus Editor, execute MyBusObject_typeScript.m on the MATLAB window.



Bus Editor - Manage Bus Objects in the Base Workspace

Filter: by Bus Name

Base Workspace
- MyBusObject
  - meterSignal
  - myEnumerationSignal

Name	DataType	Complexity	Dimensions
meterSignal	double	real	1
myEnumerationSignal	Enum:MyEnumeration	real	1

Simulink.Bus: MyBusObject

Properties

Name: MyBusObject

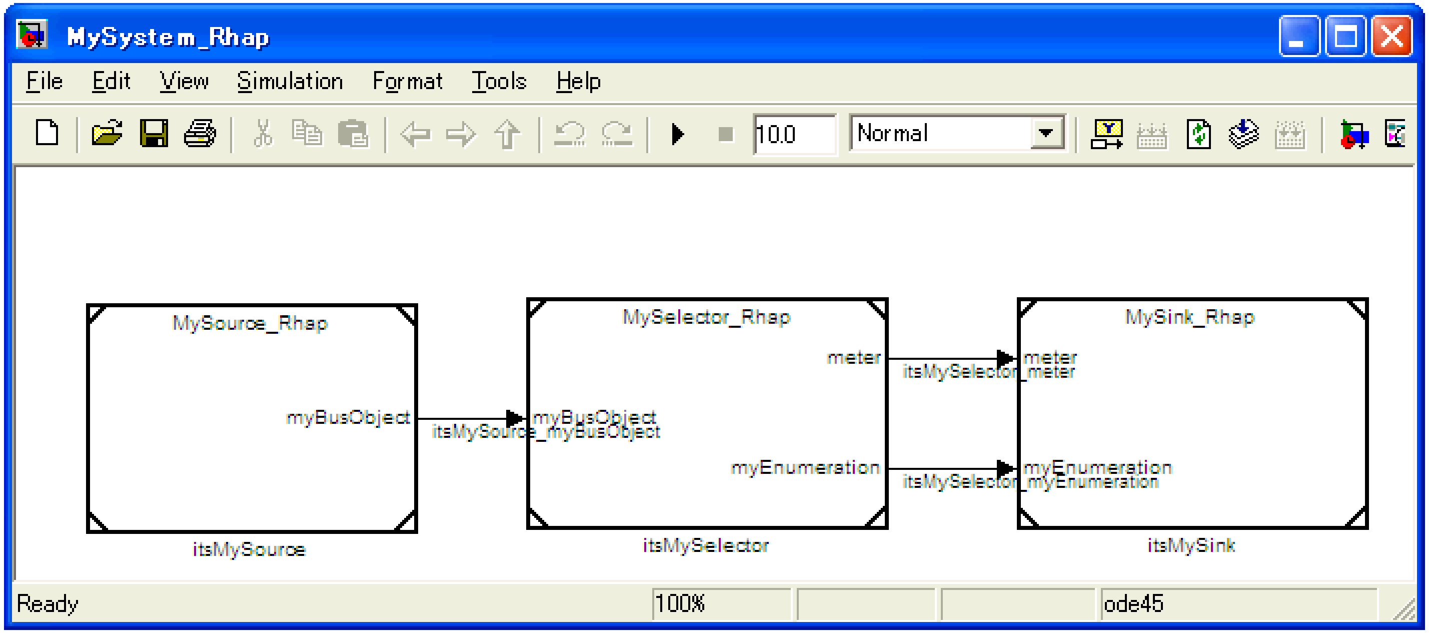
Code generation options

DataScope: Auto

HeaderFile:

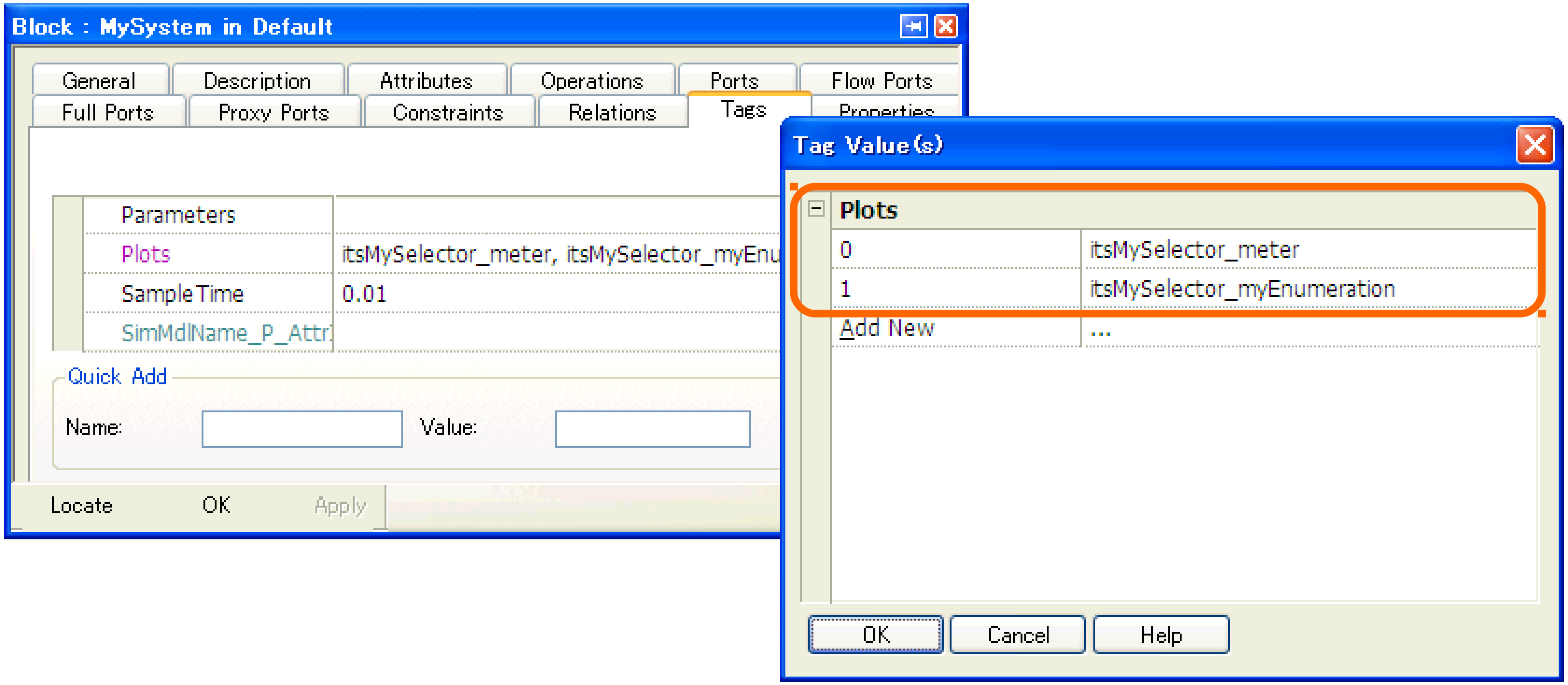
Exporting a Simulink model from MySystem block

- Exported Simulink model.



Setting signal names to be plotted

- Set the Plots tag value of MySystem block.



Parameters	
Plots	itsMySelector_meter, itsMySelector_myEnu
SampleTime	0.01
SimMdlName_P_Attr	

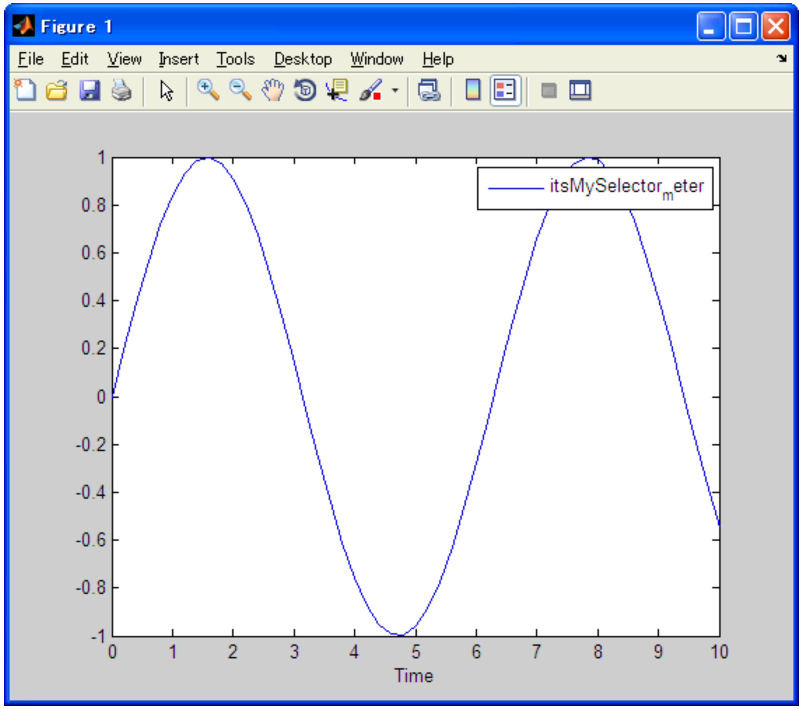
Name:

Value:

Plots	
0	itsMySelector_meter
1	itsMySelector_myEnumeration
Add New	...

Executing simulation

- Perform “Open Referenced Model and Simulate”.
  - You can see the figures.
  - Note: Enumeration is plotted by value not literal.



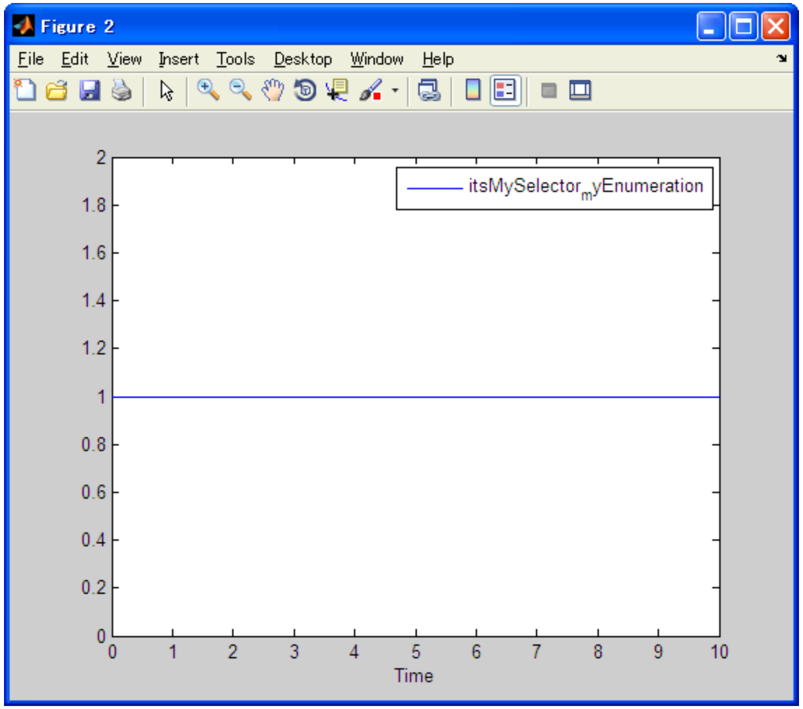
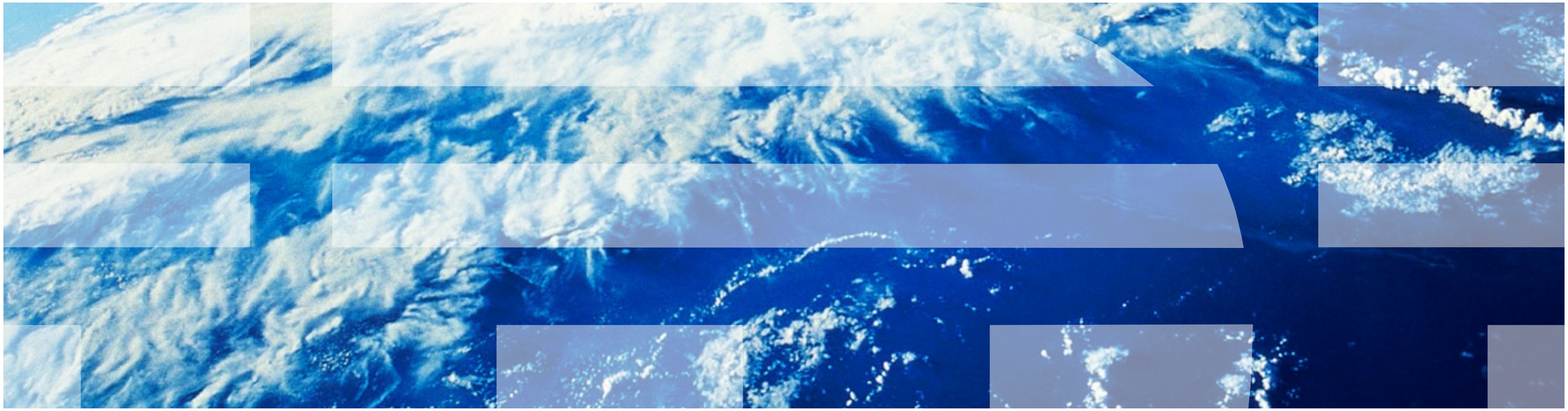


Figure 2
File Edit View Insert Tools Desktop Window Help
itsMySelector$_m$yEnumeration
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 1 2 3 4 5 6 7 8 9 10
Time

Saving the project

- Save the project.

Lesson 4: Exporting a three layer model

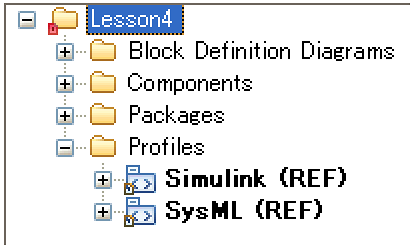


Contents of this lesson

- Create a three layer model.
- Define a super class to define a MATLAB variable used in multiple Simulink model.

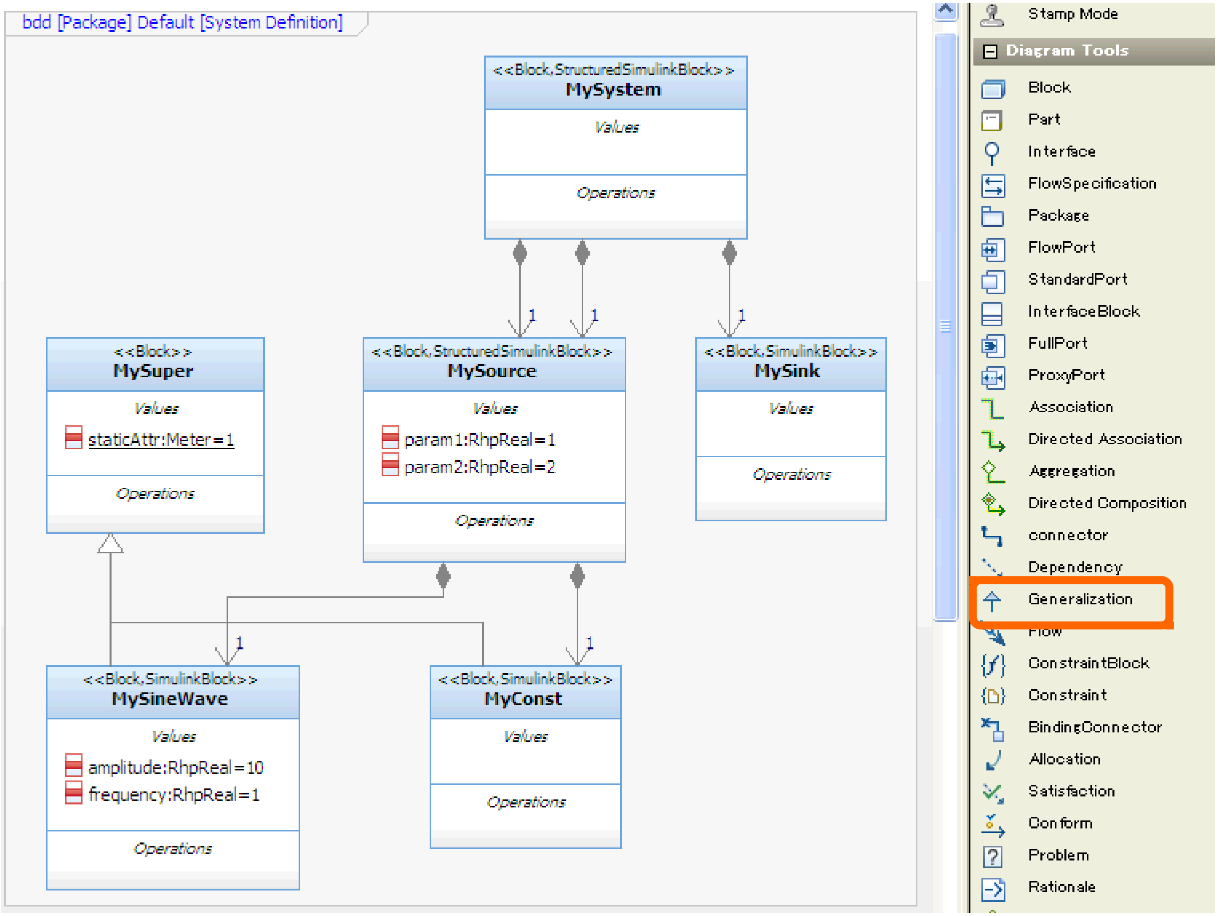
Preparations

- Start Rhapsody.
- Create a SysML project Lesson4 in C:\Work\.
- Add the Simulink profile to the model.

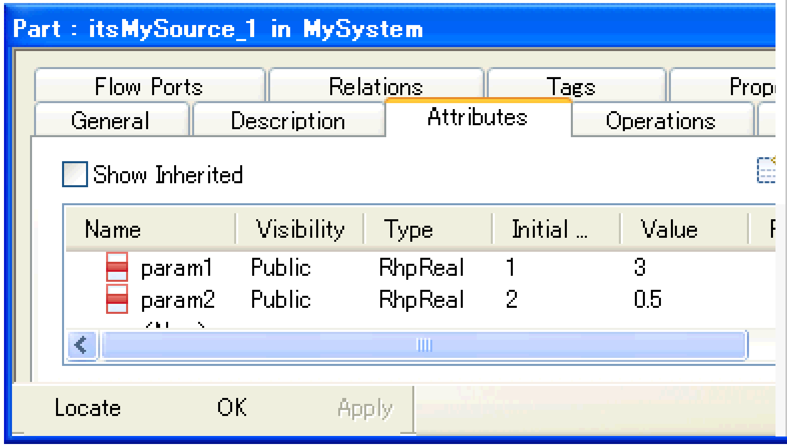


Creating a Block Definition Diagram (BDD)

- Create MySystem, MySource, MySink, MySuper, MySineWave, and MyConst blocks.
  - – Apply <<StructuredSimulinkBlock>> stereotype to MySystem and MySource blocks.
  - – Apply <<SimulinkBlock>> stereotype to MySink, MySineWave, and MyConst blocks.
  - – Use Generalization to define MATLAB variables used in multiple Simulink models.



Creating MySystem’s Internal Block Diagram (IBD)



Part : itsMySource_1 in MySystem

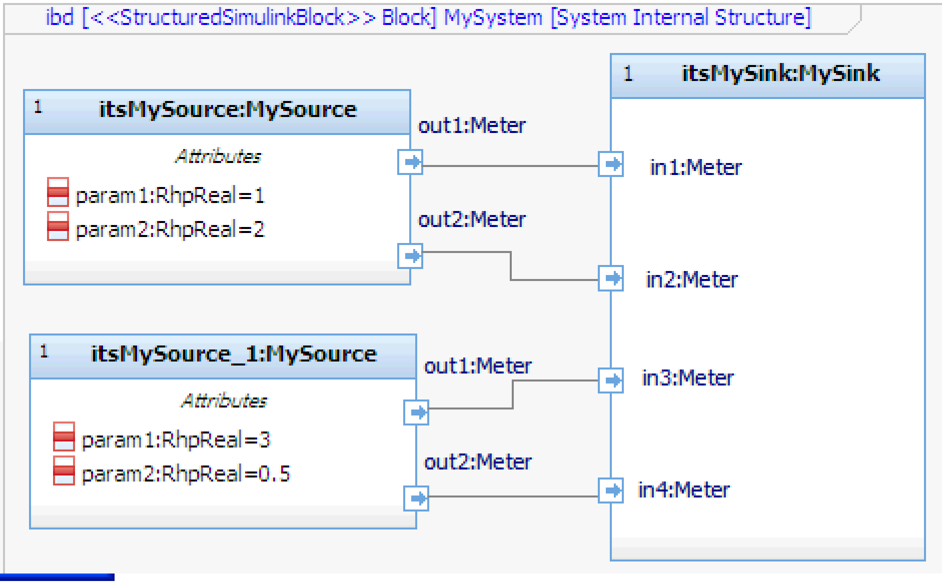
Flow Ports | Relations | Tags | Prop

General | Description | Attributes | Operations

□ Show Inherited

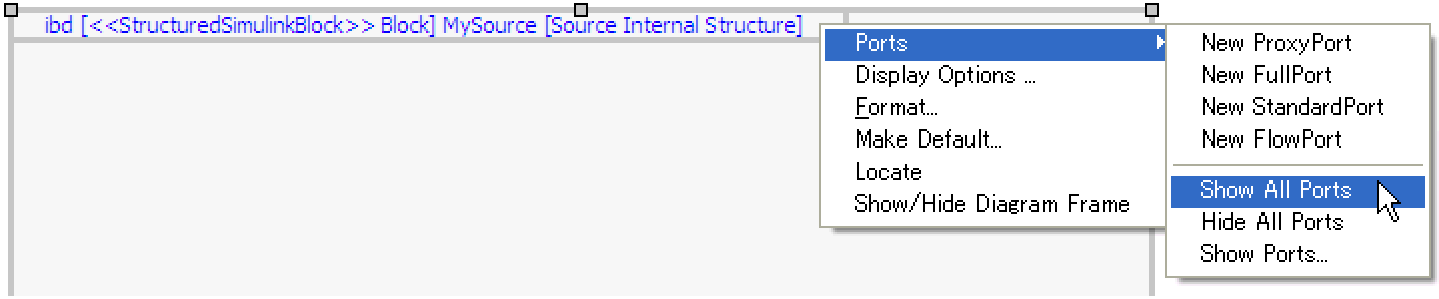
Name	Visibility	Type	Initial ...	Value	F
param1	Public	RhpReal	1	3	
param2	Public	RhpReal	2	0.5	

Locate   OK   Apply



Creating MySource's Internal Block Diagram (IBD) (1/2)

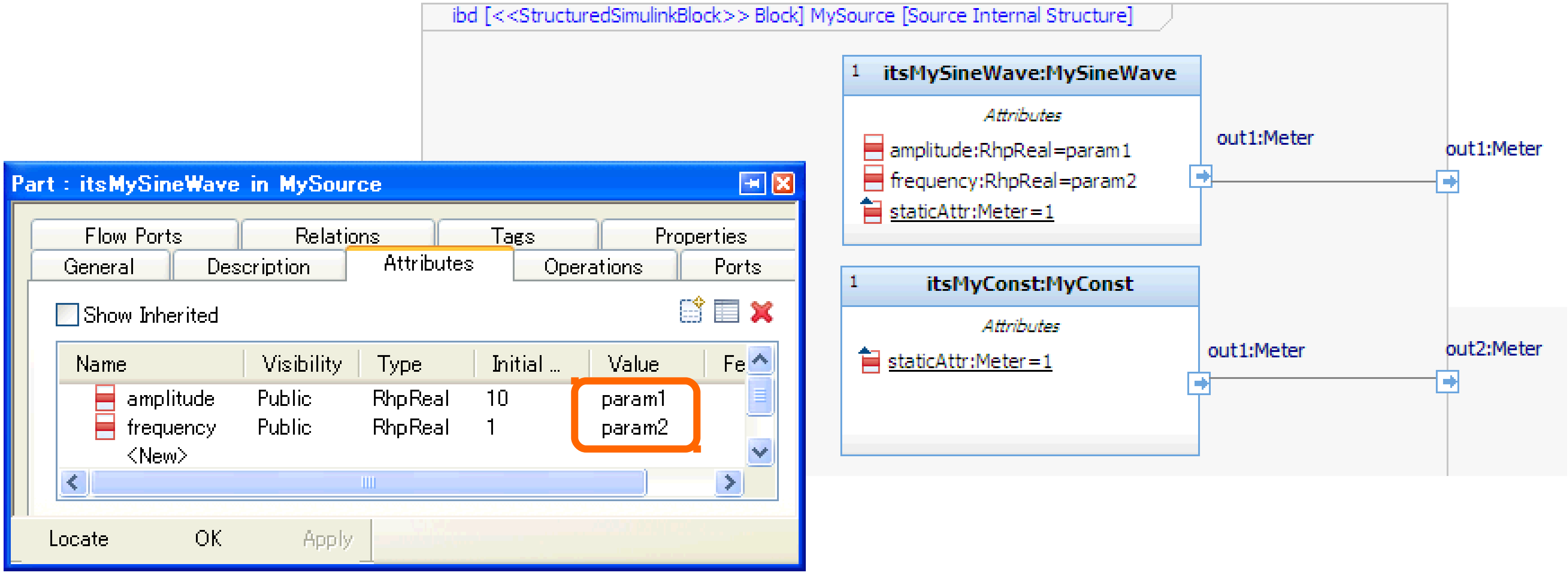
- Create MySource's Internal Block Diagram (IBD).
- Right-click IBD's frame, and select "Ports"/"Show All Ports".



ibd [<<StructuredSimulinkBlock>> Block] MySource [Source Internal Structure]

Creating MySource’s Internal Block Diagram (IBD) (2/2)

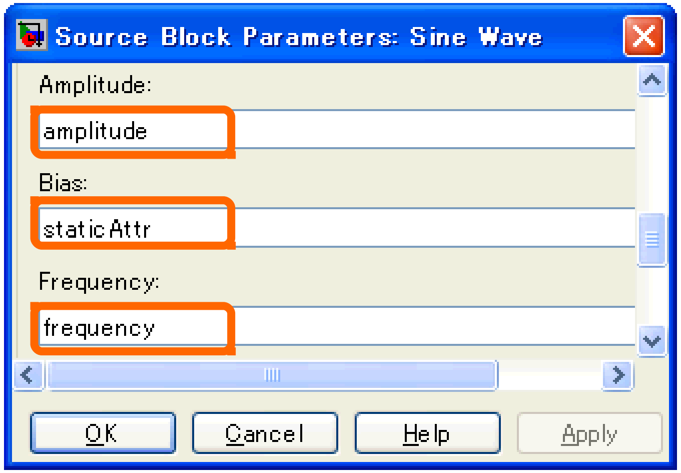
- Note: param1 and param2 are MySource’s attributes.



□ Show Inherited

Name	Visibility	Type	Initial ...	Value	Fe
amplitude	Public	RhpReal	10	param1	
frequency	Public	RhpReal	1	param2	
<New>					

Exporting a Simulink model template from MySineWave block, and editing it



Source Block Parameters: Sine Wave

Amplitude:

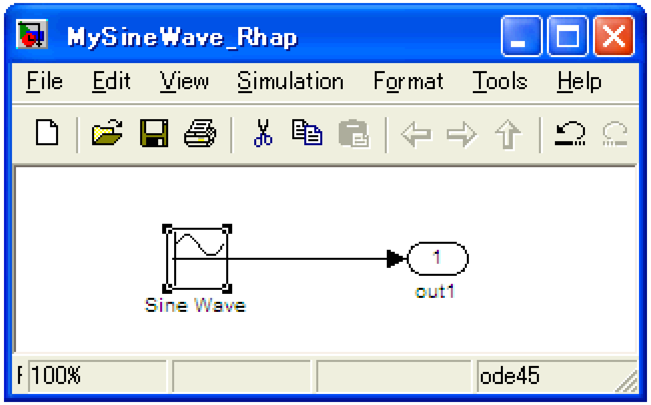
amplitude

Bias:

staticAttr

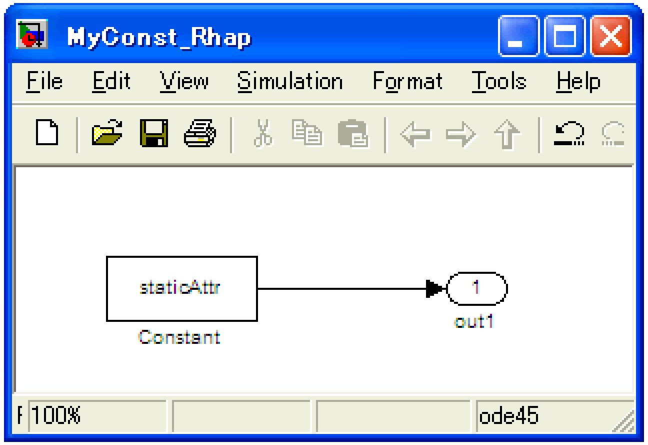
Frequency:

frequency

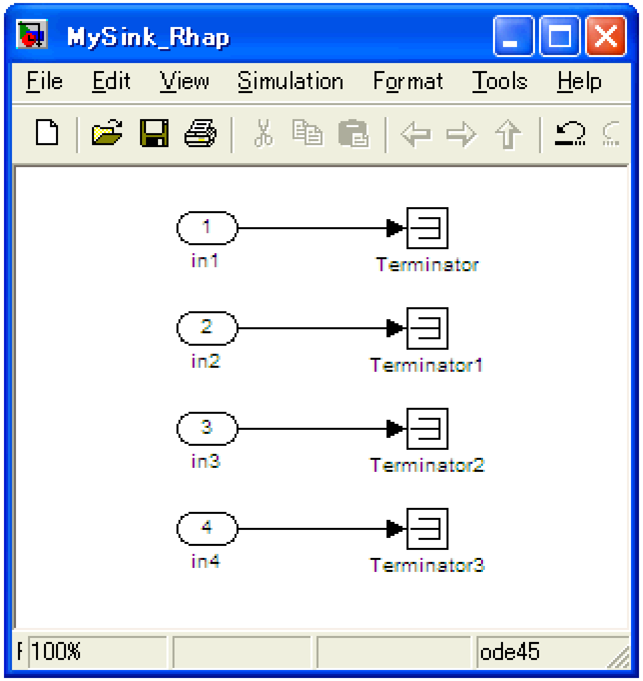


MySineWave_Rhap

Exporting a Simulink model template from MyConst block, and editing it

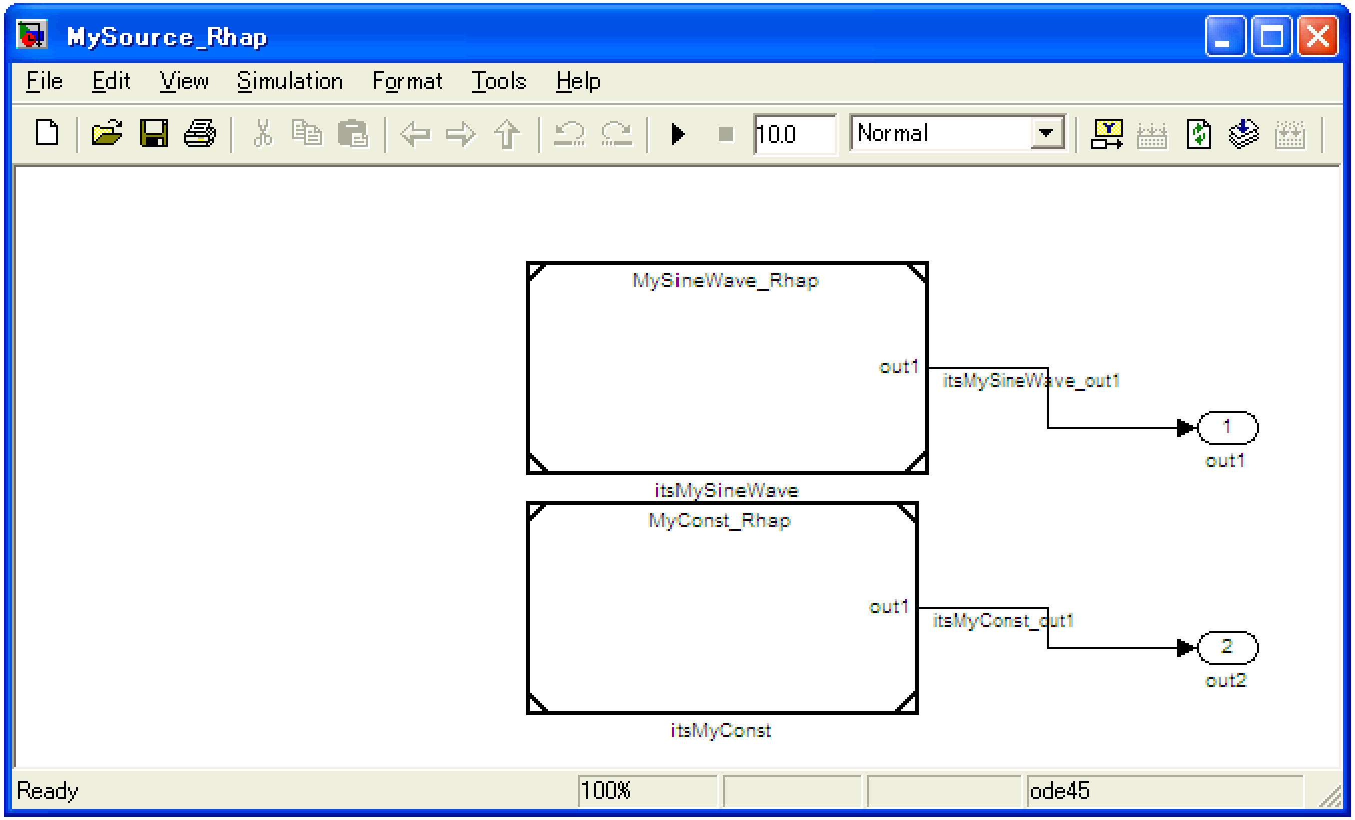


Exporting a Simulink model template from MySink block, and editing it



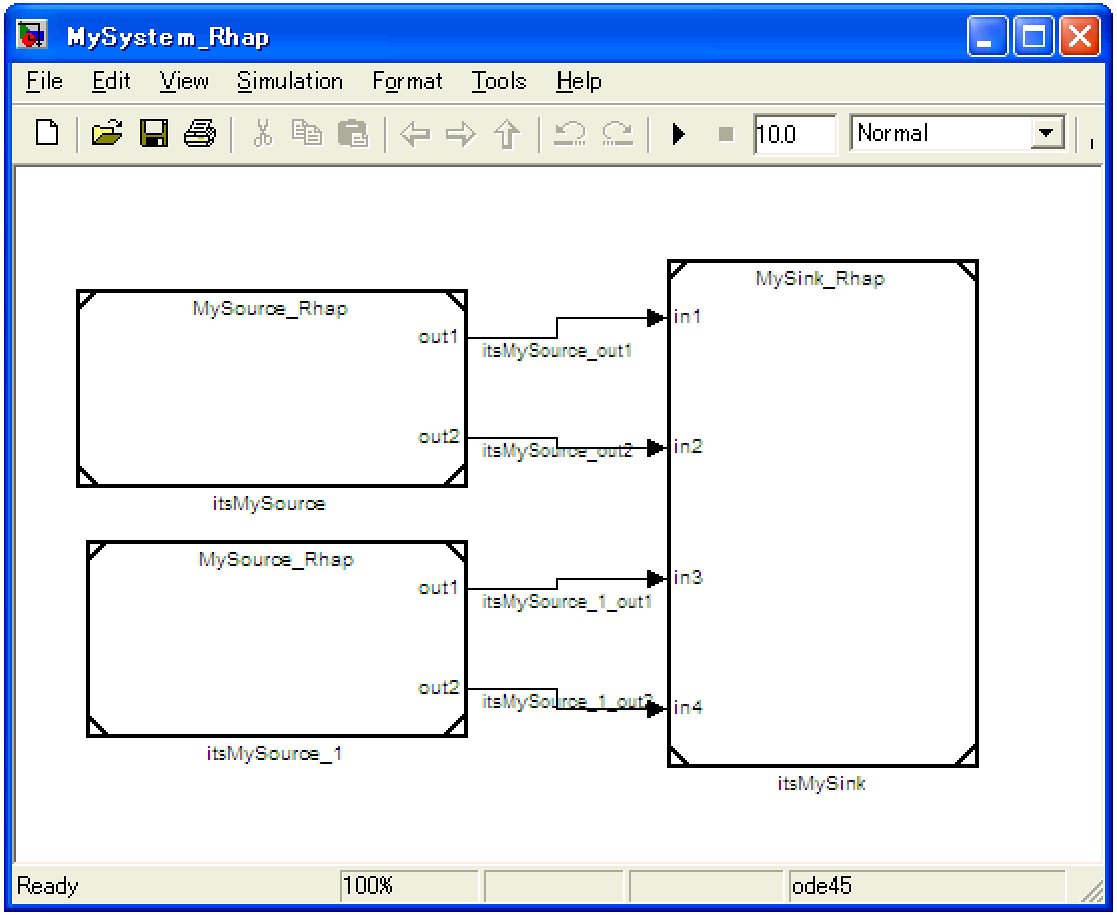
Exporting a Simulink model from MySource block

- Exported Simulink model.



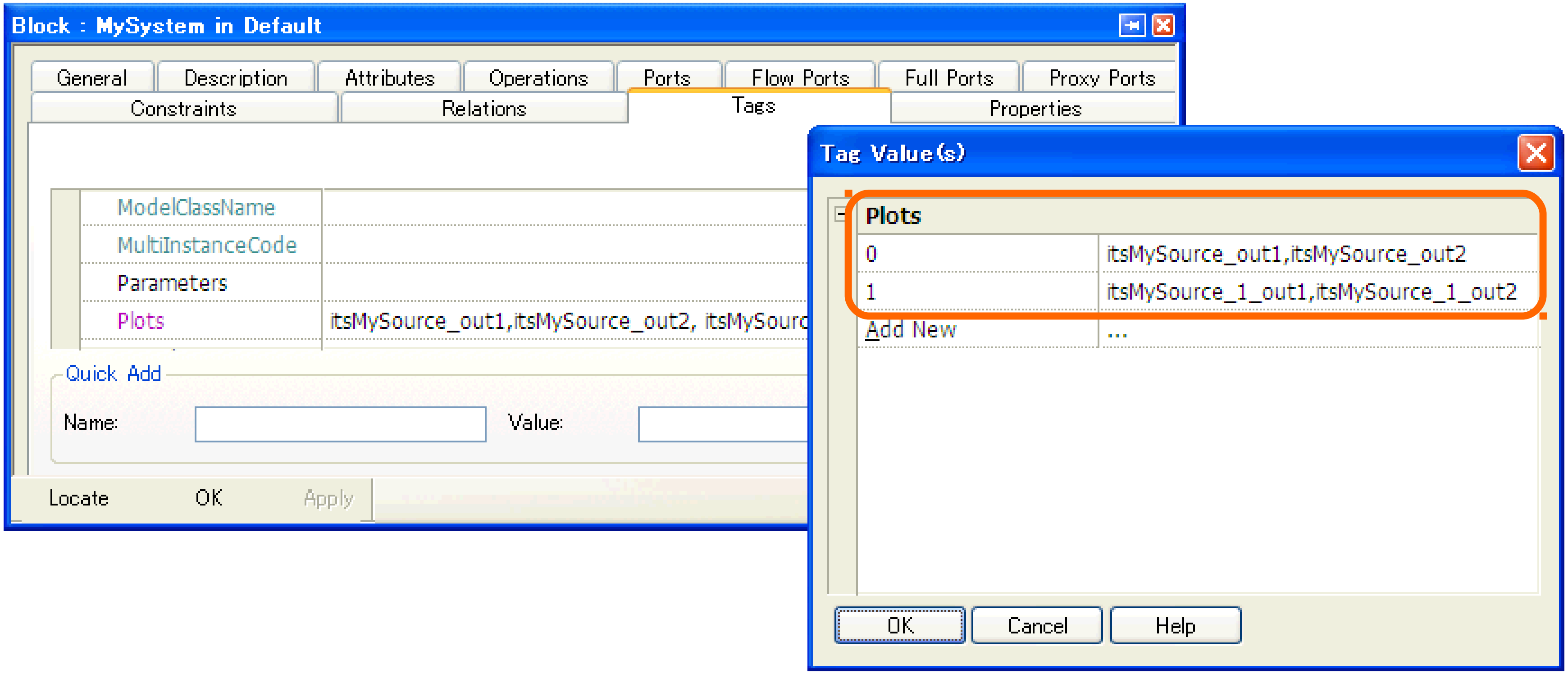
Exporting a Simulink model from MySystem block

- Exported Simulink model.



Setting signal names to be plotted

- Set the Plots tag value of MySystem block.



Block : MySystem in Default

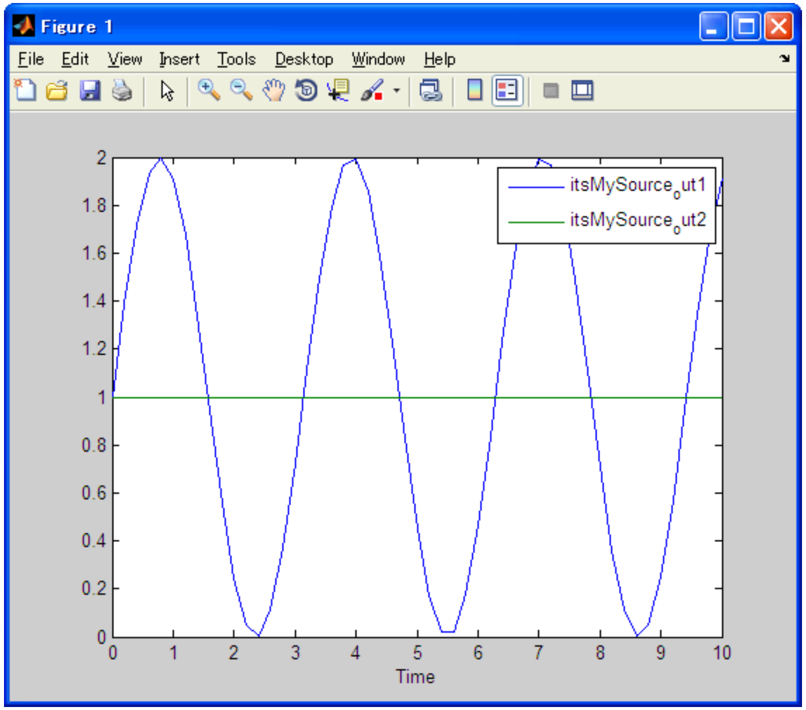
ModelClassName	
MultiInstanceCode	
Parameters	
Plots	itsMySource_out1,itsMySource_out2, itsMySourc

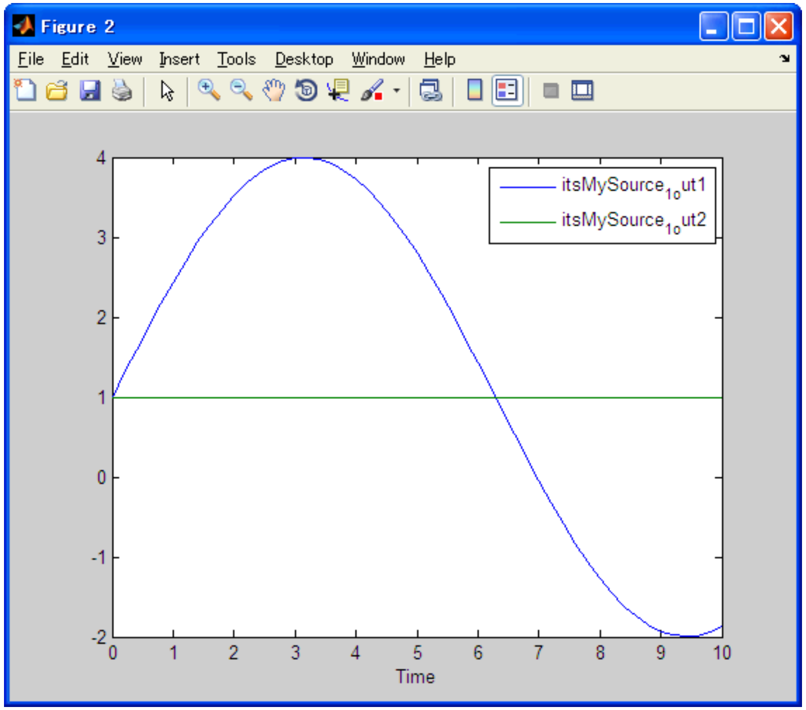
Tag Value(s)

Plots	
0	itsMySource_out1,itsMySource_out2
1	itsMySource_1_out1,itsMySource_1_out2
Add New	...

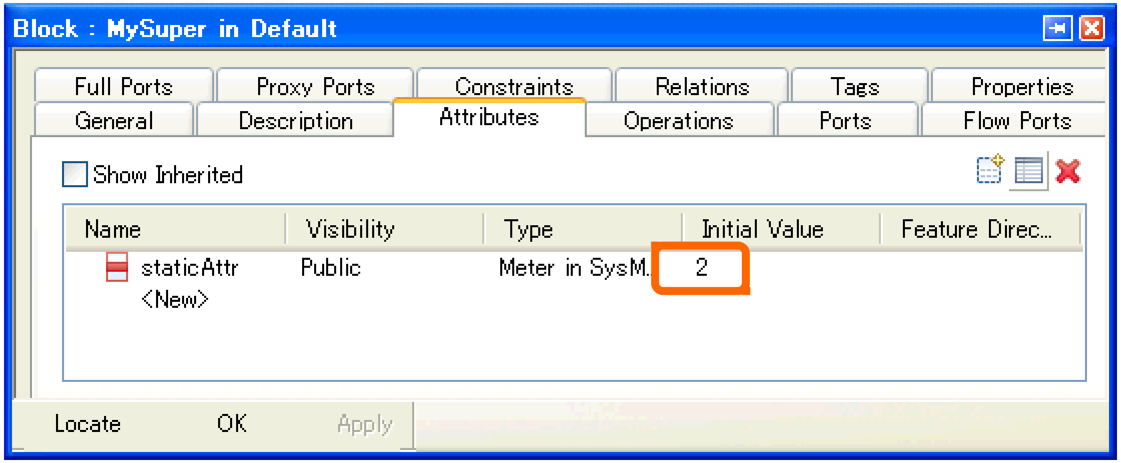
Executing simulation

- Perform “Open Referenced Model and Simulate”.
  - You can see the figures.





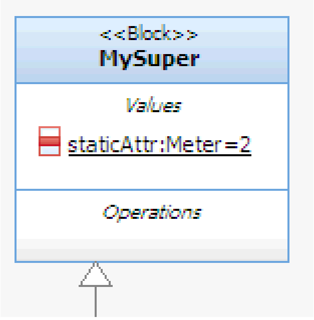
Changing the MySuper’s static attribute value



Block : MySuper in Default

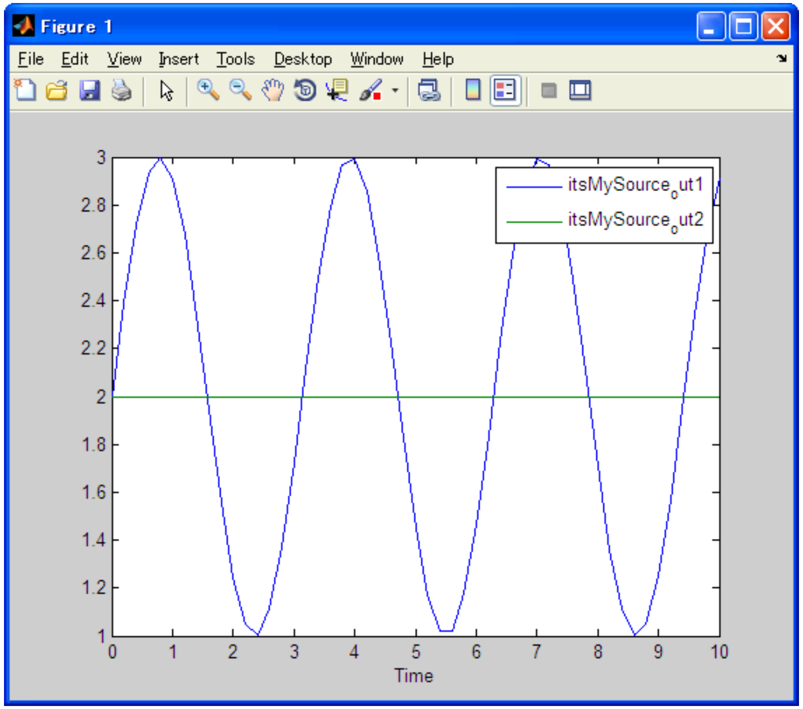
□ Show Inherited

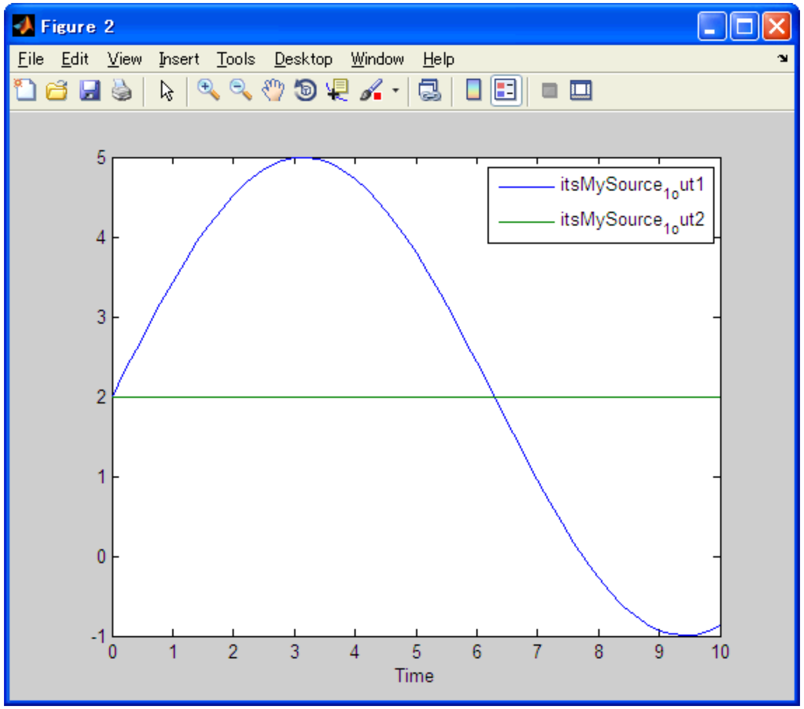
Name	Visibility	Type	Initial Value	Feature Direc...
staticAttr	Public	Meter in SysM.	2	
<New>				



Executing simulation

- Perform “Open Referenced Model and Simulate”.
  - You can see the figures.
  - The value of MATLAB variable “staticAttr” is changed both in MySineWave and in MyConst because MySuper is a super class of MySineWave and MyConst.

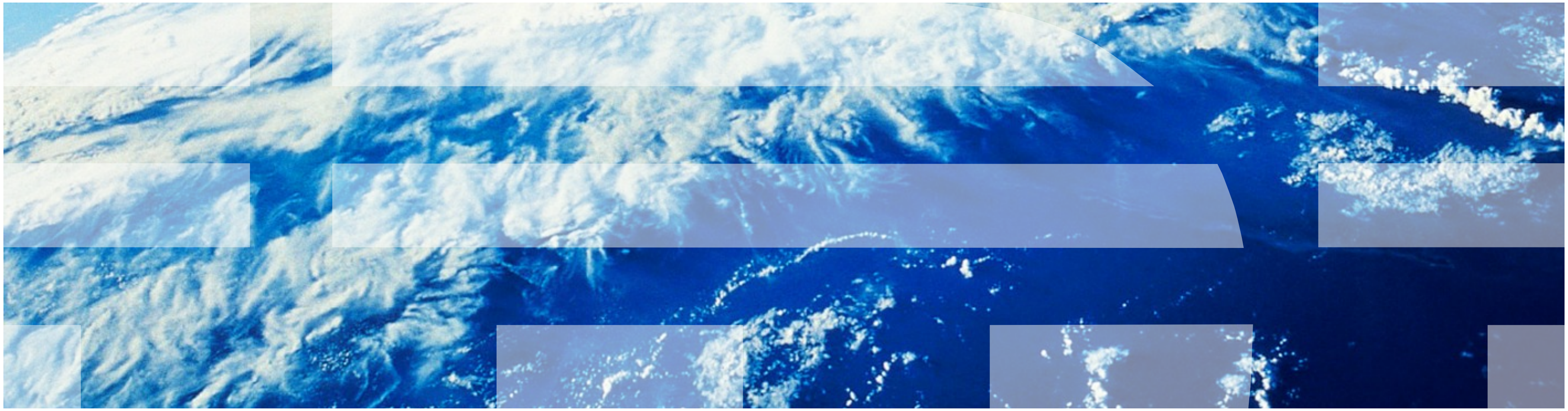




Saving the project

- Save the project.

Lesson 5: Generating an S-Function

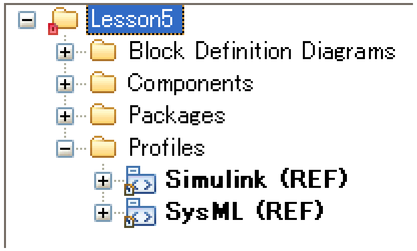


Contents of this lesson

- Define behavior using a statechart.
- Generate an S-Function and export a Simulink model.
- Execute the exported Simulink model.
- Define a Panel Diagram.
- Use Rhapsody’s animation to confirm the generated S-Function’s behavior.

Preparations

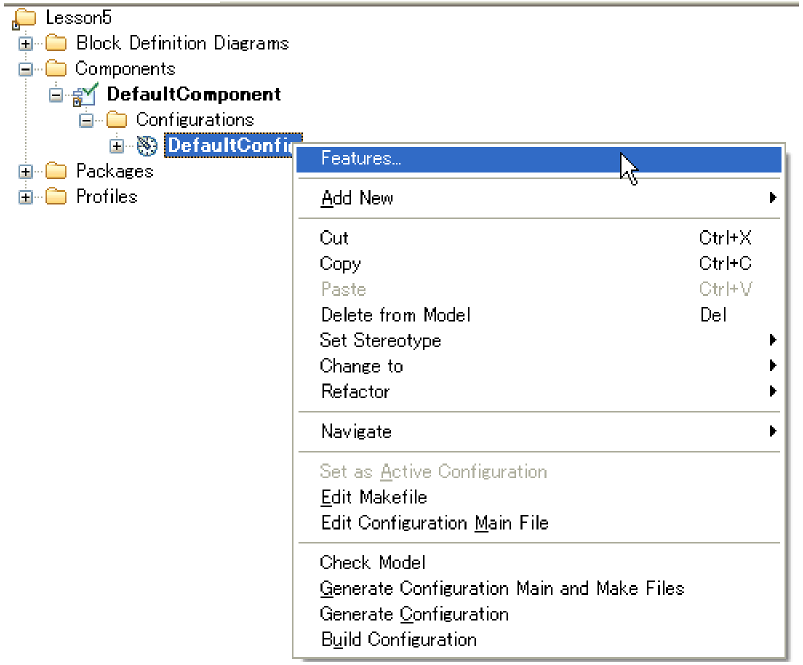
- Start Rhapsody Developer for C++ or Rhapsody Designer for Systems Engineers.
- Create a SysML project Lesson5 in C:\Work\.
- Add the Simulink profile to the model.

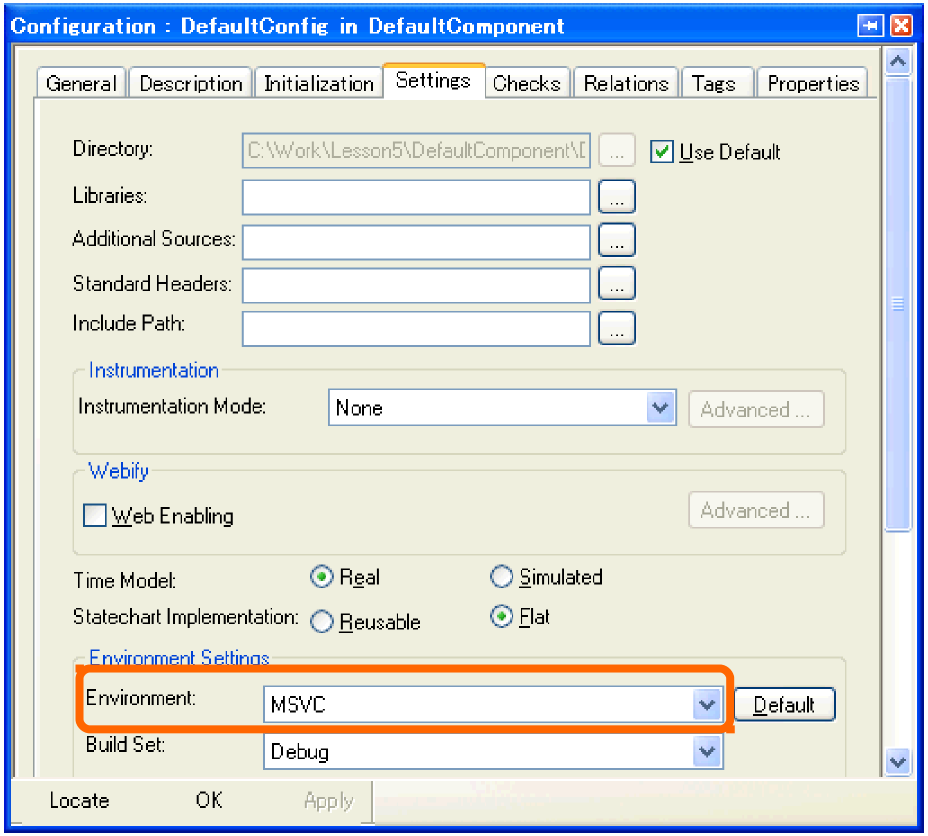


Lesson5
Block Definition Diagrams
Components
Packages
Profiles
Simulink (REF)
SysML (REF)

Setting up a compiler

- Right-click DefaultConfiguration configuration, and select “Features…”.
- Select “Settings” tab, and select “Environment” so that it specifies your compiler.
  - If you want to use Microsoft Visual Studio 2010 or Microsoft Visual Studio 2008, select “MSVC”.




Configuration : DefaultConfig in DefaultComponent

Directory: C:\Work\Lesson5\DefaultComponent\[☑ Use Default

Libraries:

Additional Sources:

Standard Headers:

Include Path:

Instrumentation

Instrumentation Mode: None

Webify

□ Web Enabling

Time Model: ● Real ○ Simulated

Statechart Implementation: ○ Reusable ● Flat

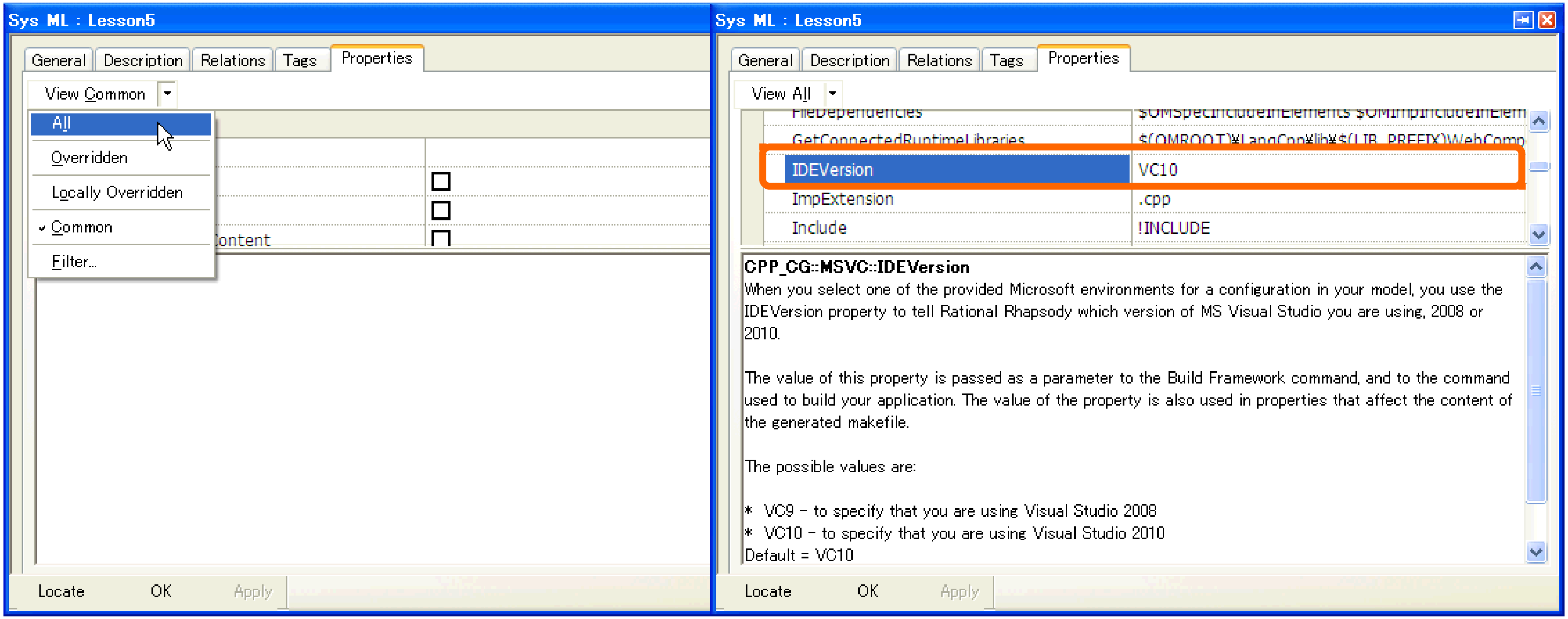
Environment Settings

Environment: MSVC

Build Set: Debug

Setting up an IDE Version

- Right-click Lesson5 project, and select “Features…”.
- Click “Properties” tab, and select “All”.
- Expand “CPP_CG” and “MSVC”, and set VC9 or VC10 for IDEVersion
- Click “OK” button.



FileDependencies	$OMSpecIncludeInElements $OMImplIncludeInElem
GetConnectedRuntimeLibraries	$(OMROOT)¥LangCpp¥lib¥$(LIB_PREFIX)WebComp
IDEVersion	VC10
ImpExtension	.cpp
Include	!INCLUDE

CPP_CG::MSVC::IDEVersion

When you select one of the provided Microsoft environments for a configuration in your model, you use the IDEVersion property to tell Rational Rhapsody which version of MS Visual Studio you are using, 2008 or 2010.

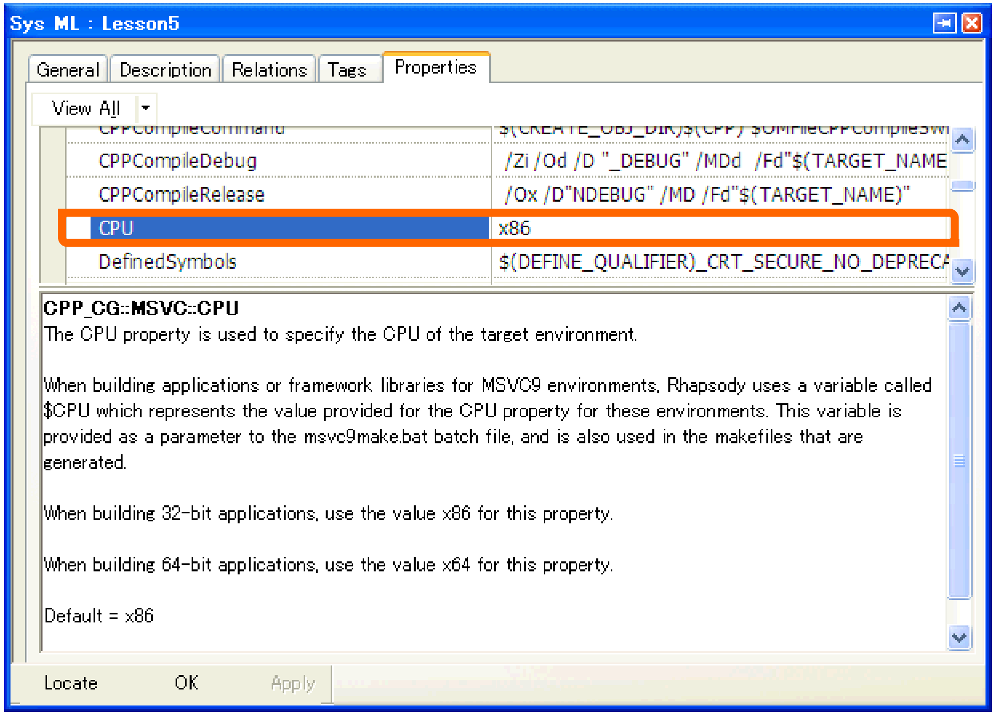
The value of this property is passed as a parameter to the Build Framework command, and to the command used to build your application. The value of the property is also used in properties that affect the content of the generated makefile.

The possible values are:

* VC9 - to specify that you are using Visual Studio 2008
* VC10 - to specify that you are using Visual Studio 2010
Default = VC10

Setting up a CPU

- If you want to use 64-bit compiler, perform the following steps:
  - – Right-click Lesson5 project, and select “Features…”.
  - – Click “Properties” tab, and select “All”.
  - – Expand “CPP_CG” and “MSVC”, and set “x64” for CPU.
  - – Click “OK” button.



CPPCompileCommand	$(CREATE_OBJ_DIR)$(CPP) $OMFileCPPCompileSwi
CPPCompileDebug	/Zi /Od /D "_DEBUG" /MDd /Fd"$(TARGET_NAME
CPPCompileRelease	/Ox /D"NDEBUG" /MD /Fd"$(TARGET_NAME)"
CPU	x86
DefinedSymbols	$(DEFINE_QUALIFIER)_CRT_SECURE_NO_DEPRECA

CPP_CG::MSVC::CPU
The CPU property is used to specify the CPU of the target environment.

When building applications or framework libraries for MSVC9 environments, Rhapsody uses a variable called $CPU which represents the value provided for the CPU property for these environments. This variable is provided as a parameter to the msvc9make.bat batch file, and is also used in the makefiles that are generated.

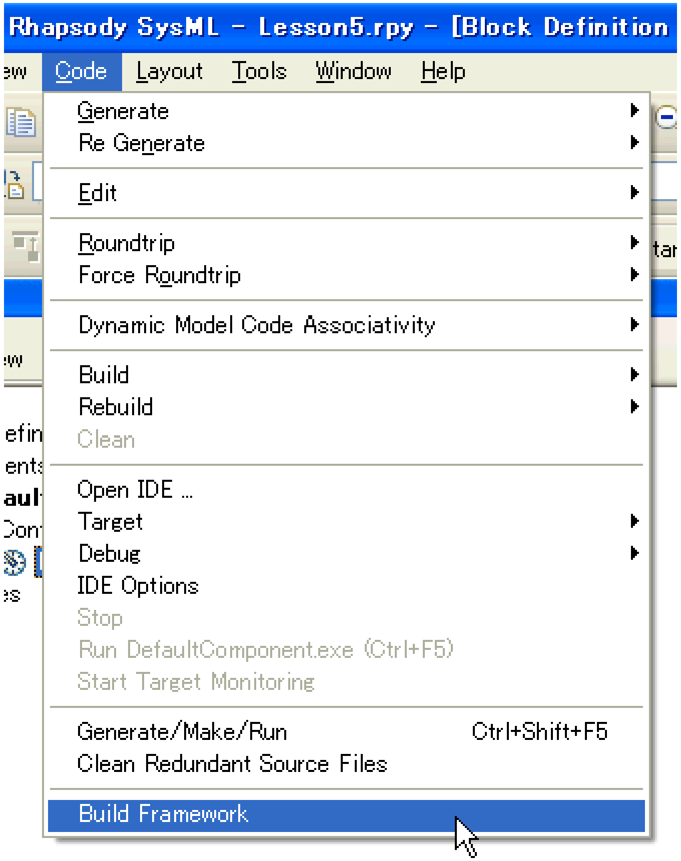
When building 32-bit applications, use the value x86 for this property.

When building 64-bit applications, use the value x64 for this property.

Default = x86

Building framework

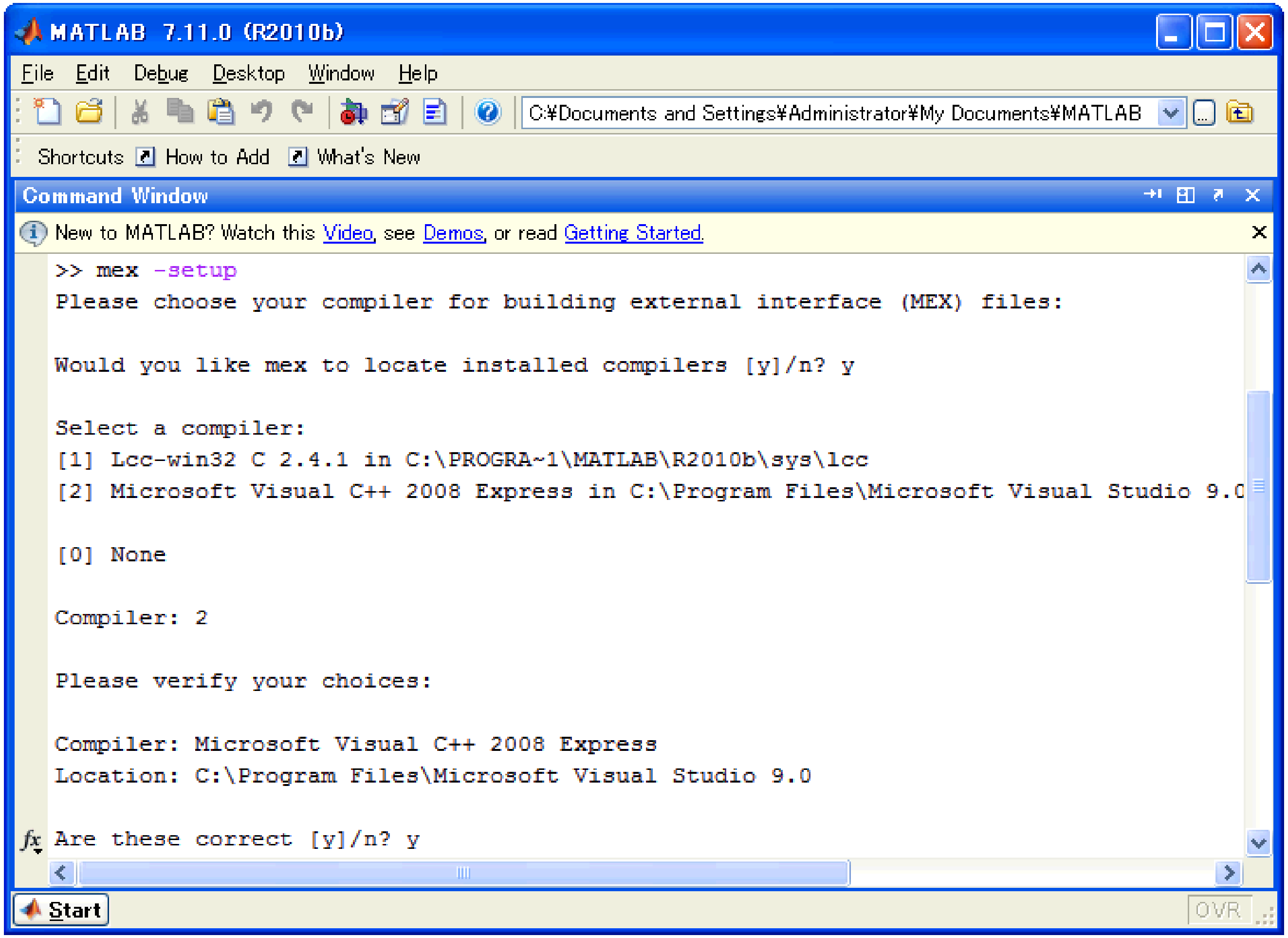
- If you want to use 6-bit compiler, select “Code”/”Build Framework”.



Rhapsody SysML - Lesson5.rpy - [Block Definition
Code
Layout
Tools
Window
Help
Generate
Re Generate
Edit
Roundtrip
Force Roundtrip
Dynamic Model Code Associativity
Build
Rebuild
Clean
Open IDE ...
Target
Debug
IDE Options
Stop
Run DefaultComponent.exe (Ctrl+F5)
Start Target Monitoring
Generate/Make/Run Ctrl+Shift+F5
Clean Redundant Source Files
Build Framework

Setting up a compiler in MATLAB

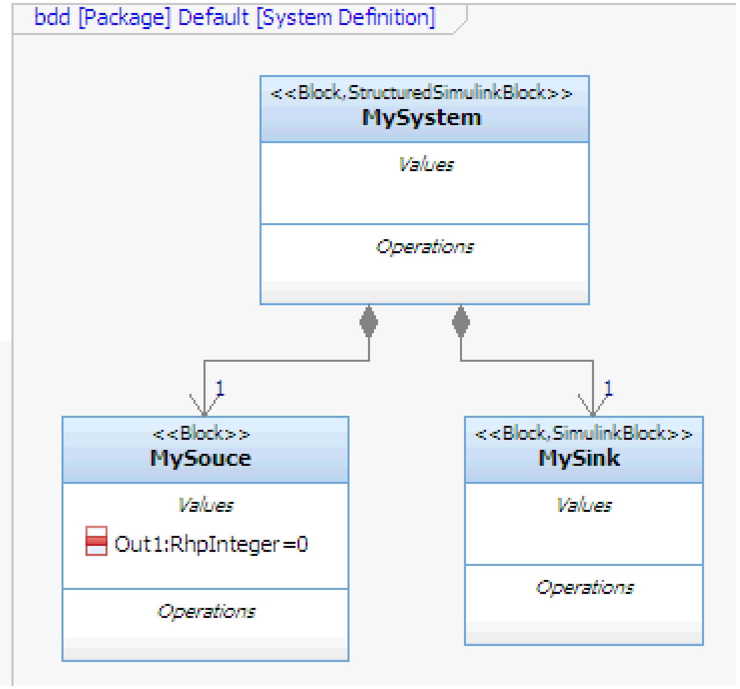
- We have to set up a compiler in MATLAB to compile an S-Function.
- Open a MATLAB command window, execute “mex -setup” on the window, and close the window.


MATLAB 7.11.0 (R2010b)
File Edit Debug Desktop Window Help
C:¥Documents and Settings¥Administrator¥My Documents¥MATLAB
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Start
OVR

```
>> mex -setup
Please choose your compiler for building external interface (MEX) files:

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:
[1] Lcc-win32 C 2.4.1 in C:\PROGRA~1\MATLAB\R2010b\sys\lcc
[2] Microsoft Visual C++ 2008 Express in C:\Program Files\Microsoft Visual Studio 9.0

[0] None

Compiler: 2

Please verify your choices:

Compiler: Microsoft Visual C++ 2008 Express
Location: C:\Program Files\Microsoft Visual Studio 9.0

Are these correct [y]/n? y
```

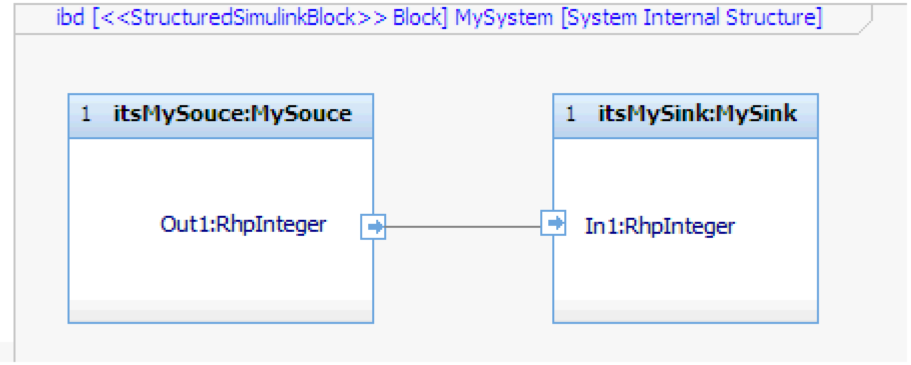
Create a Block Definition Diagram (BDD)

- Create a BDD under the Default package and place MySystem, MySource, and MySink blocks.
  - – Apply <<StructuredSimulinkBlock>> to MySystem, and <<SimulinkBlock>> to MySink.
  - – Add an attribute “Out1” to MySource.

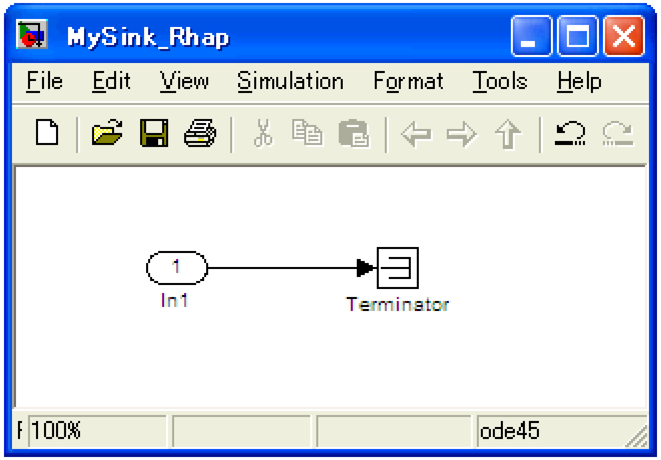


Creating an IBD of MySystem block

- Create an IBM of MySystem block as below.

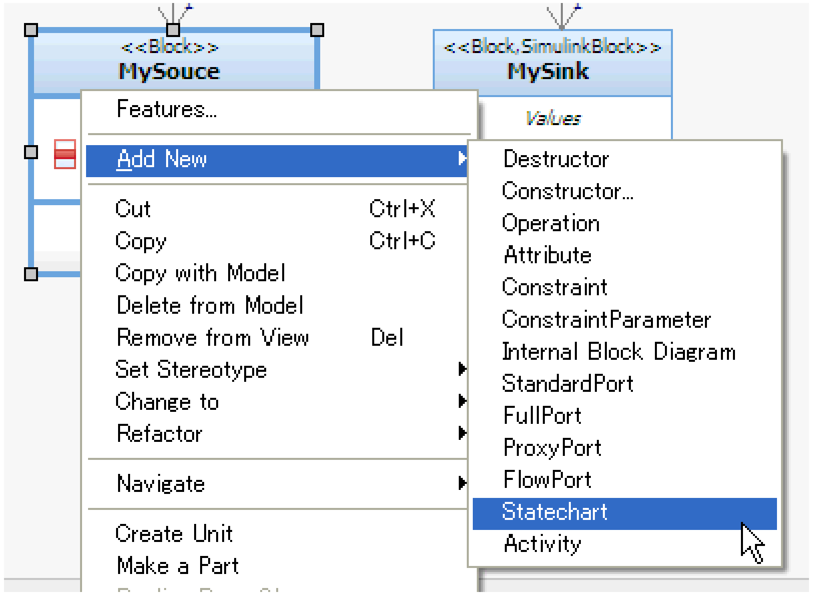

ibd [<<StructuredSimulinkBlock>> Block] MySystem [System Internal Structure]
1 itsMySouce:MySouce
Out1:RhpInteger
1 itsMySink:MySink
In1:RhpInteger

Exporting a Simulink model template from MySink block, and editing it



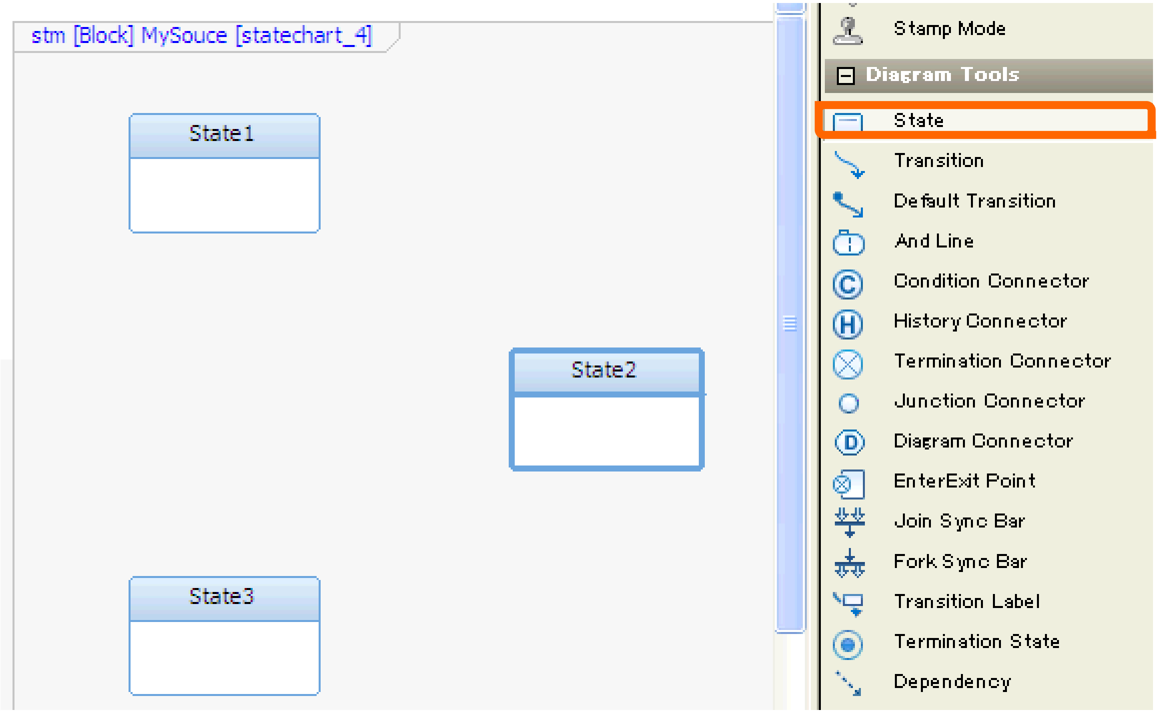
Creating a statechart (1/4)

- Add a statechart to MySource block.



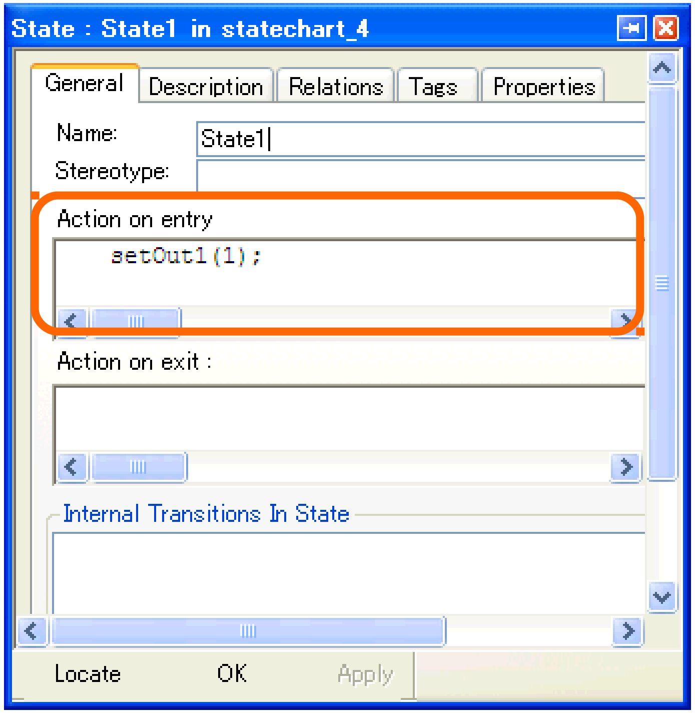
Creating a statechart (2/4)

- Add three states, and name them State1, State2, and State3.



Creating a statechart (3/4)

- Set “Action on entry” to each state.



State : State1 in statechart_4

Name: State1

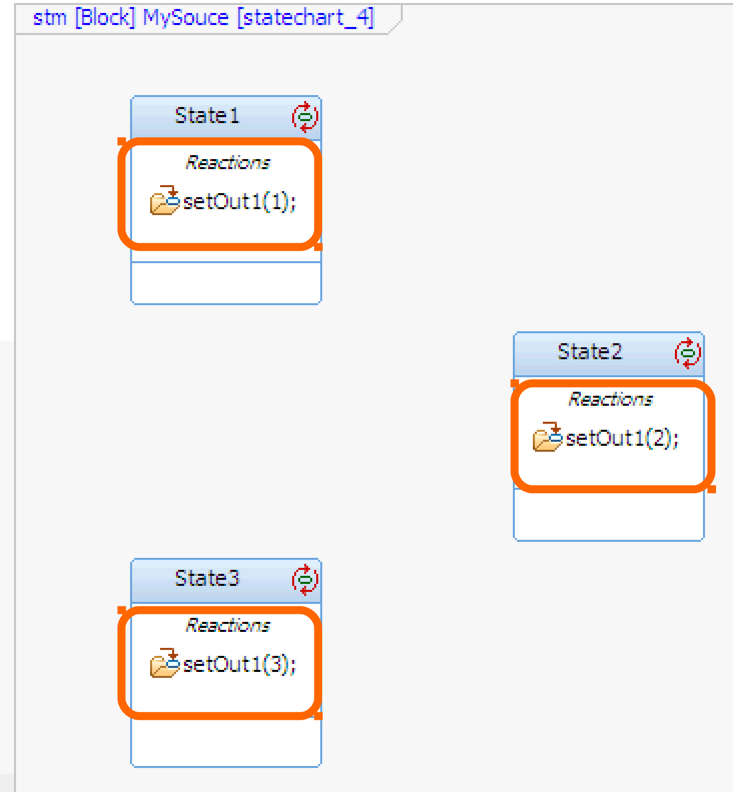
Stereotype:

Action on entry

```
    setOut1(1);
```

Action on exit :

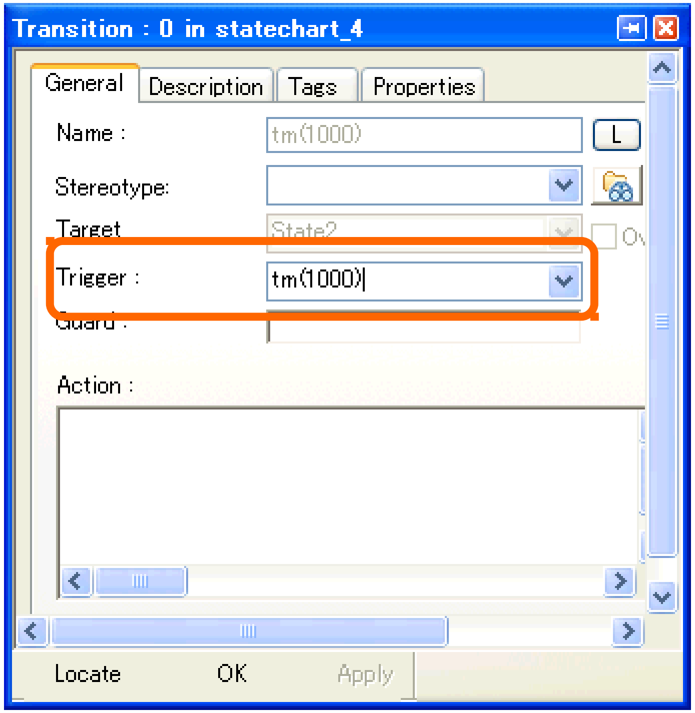
Internal Transitions In State



stm [Block] MySouce [statechart_4]

Creating a statechart (4/4)

- Add transitions, and set tm(1000) as Timeout Triggers.
  - Note: The unit of tm(…) is milliseconds.
- Add a default transition.



Transition : 0 in statechart_4

General | Description | Tags | Properties

Name : tm(1000)

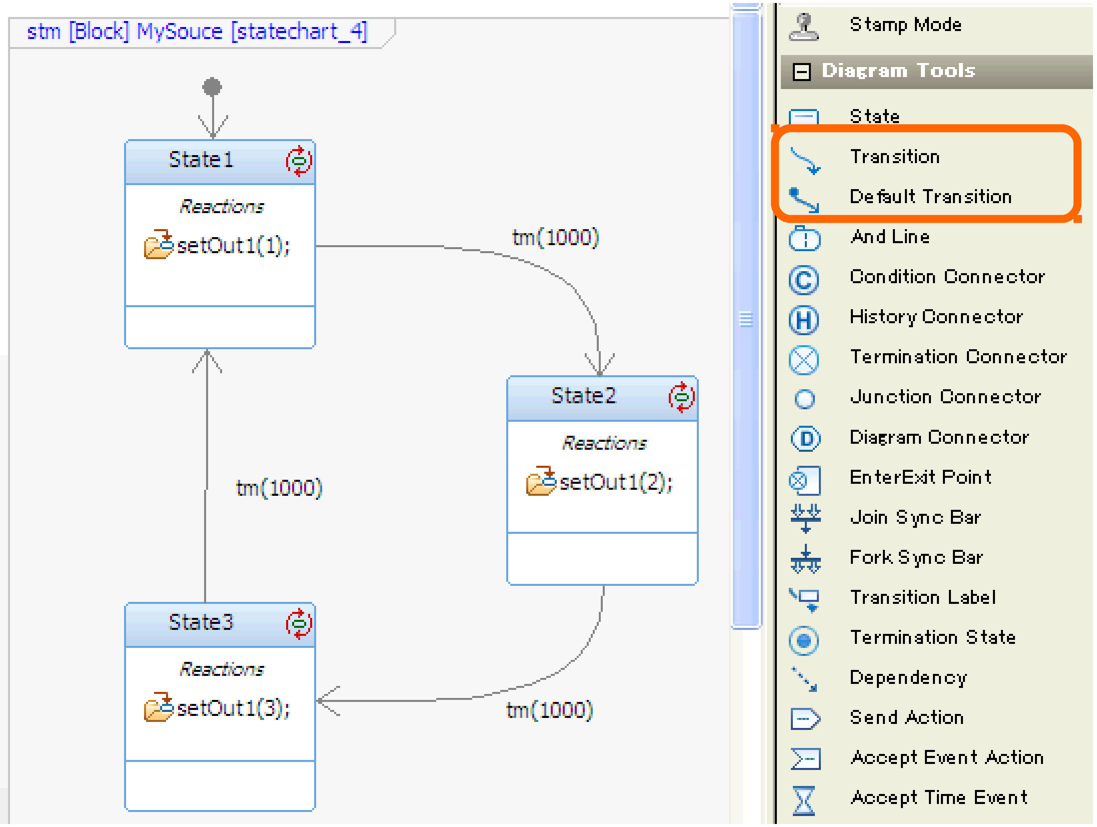
Stereotype:

Target : State2

Trigger : tm(1000)

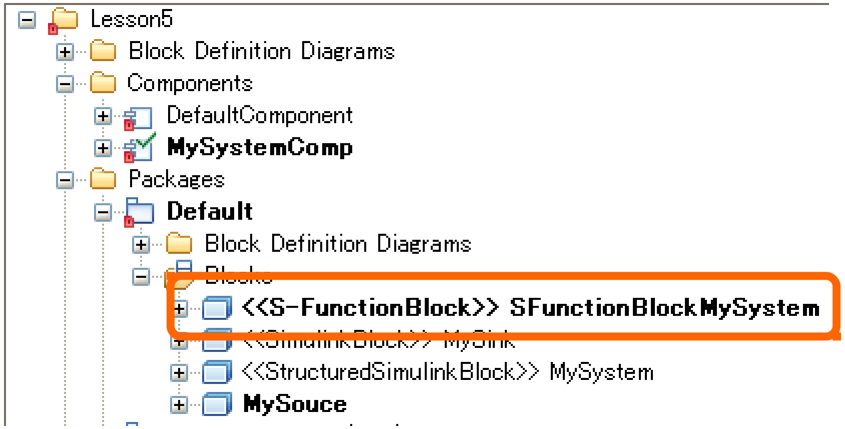
Guard :

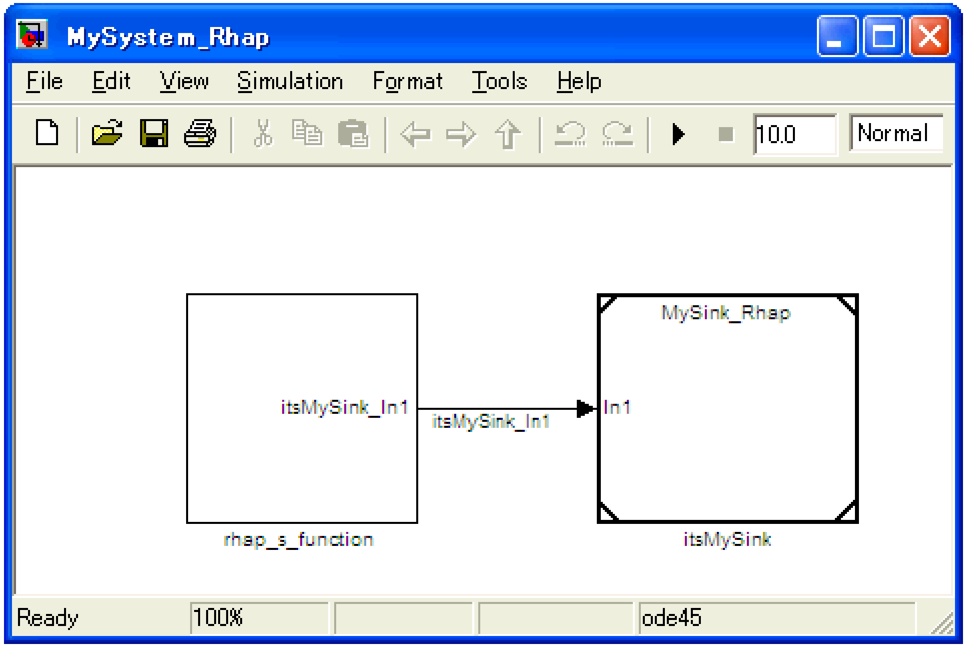
Action :



Exporting a Simulink model, and simulating it

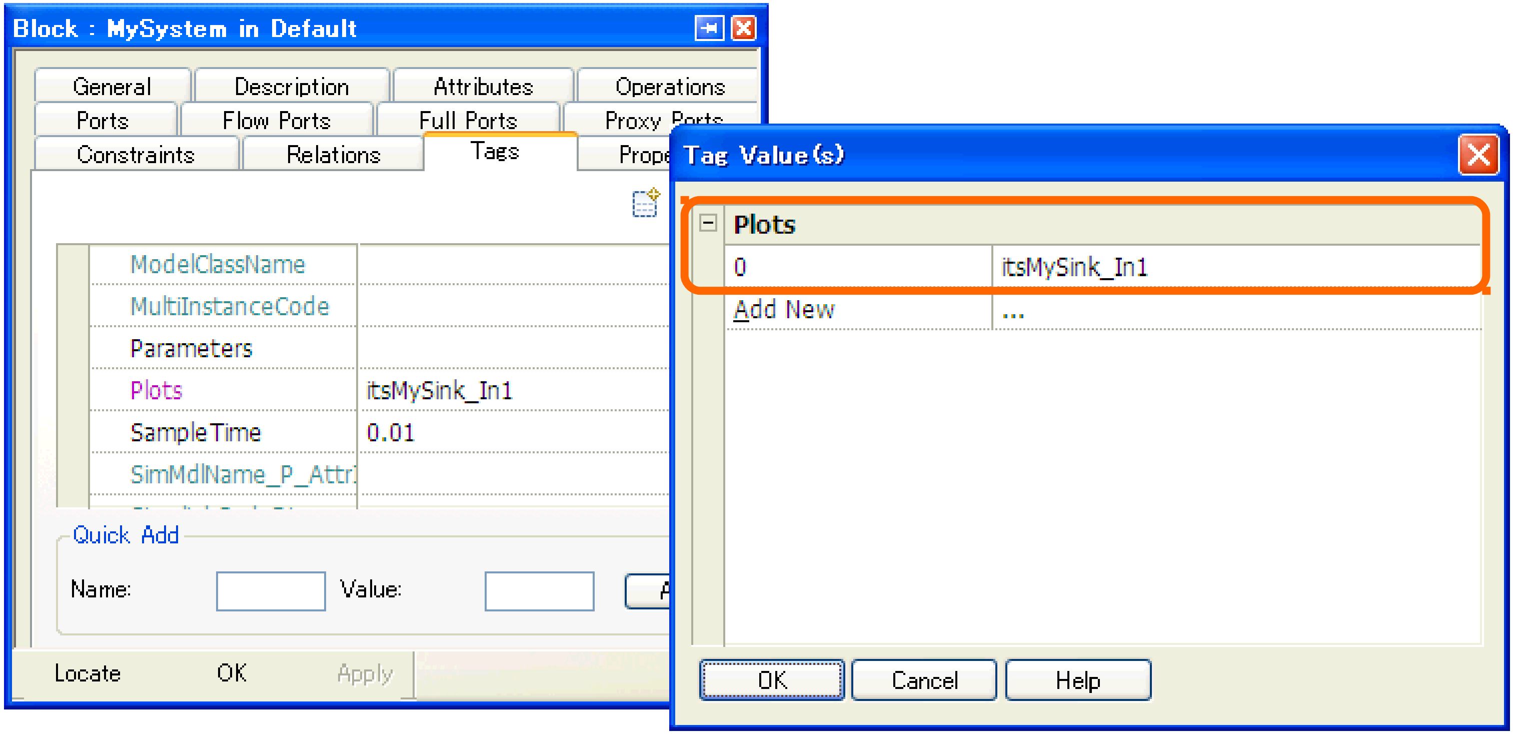
- If you are using Rhapsody Designer for Systems Engineers, skip this page.
- Right-click on MySystem block, and select “Simulink”/”Export”.
  - Note: Building an S-Function takes long time. Please do not change the active component and the active configuration until the export finishes.
  - You can find that <<S-FunctionBlock>> SFunctionBlockSystem block is generated on the browser.


Lesson5
Block Definition Diagrams
Components
DefaultComponent
MySystemComp
Packages
Default
Block Definition Diagrams
Blocks
<<S-FunctionBlock>> SFunctionBlockMySystem
<<SimulinkBlock>> MySink
<<StructuredSimulinkBlock>> MySystem
MySouce



Setting a signal name to be plotted

- Set the Plots tag value of MySystem block.



Block : MySystem in Default

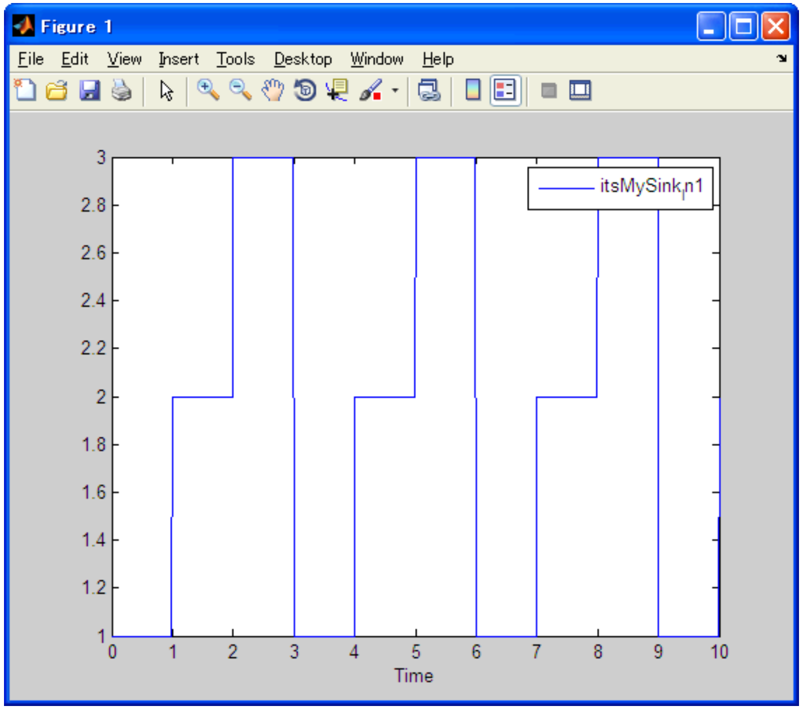
ModelClassName	
MultiInstanceCode	
Parameters	
Plots	itsMySink_In1
SampleTime	0.01
SimMdlName_P_Attr	

Tag Value(s)

Plots	
0	itsMySink_In1
Add New	...

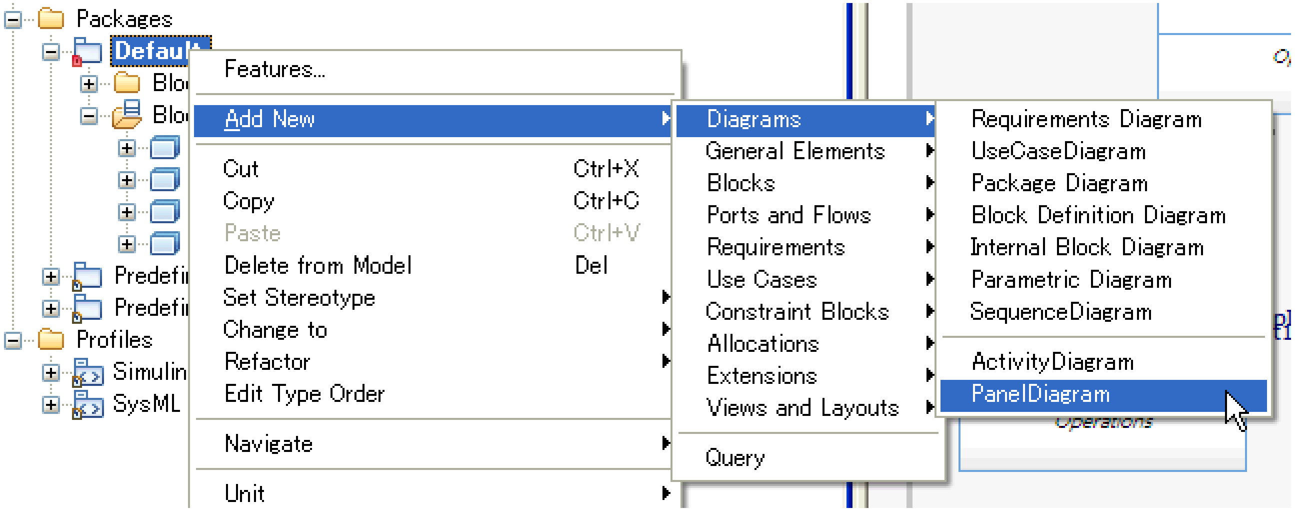
Executing simulation

- If you are using Rhapsody Designer for Systems Engineers, skip this page.
- Right-click on the MySystem block, and select “Simulink”/“Open Referenced Model and Simulate”.



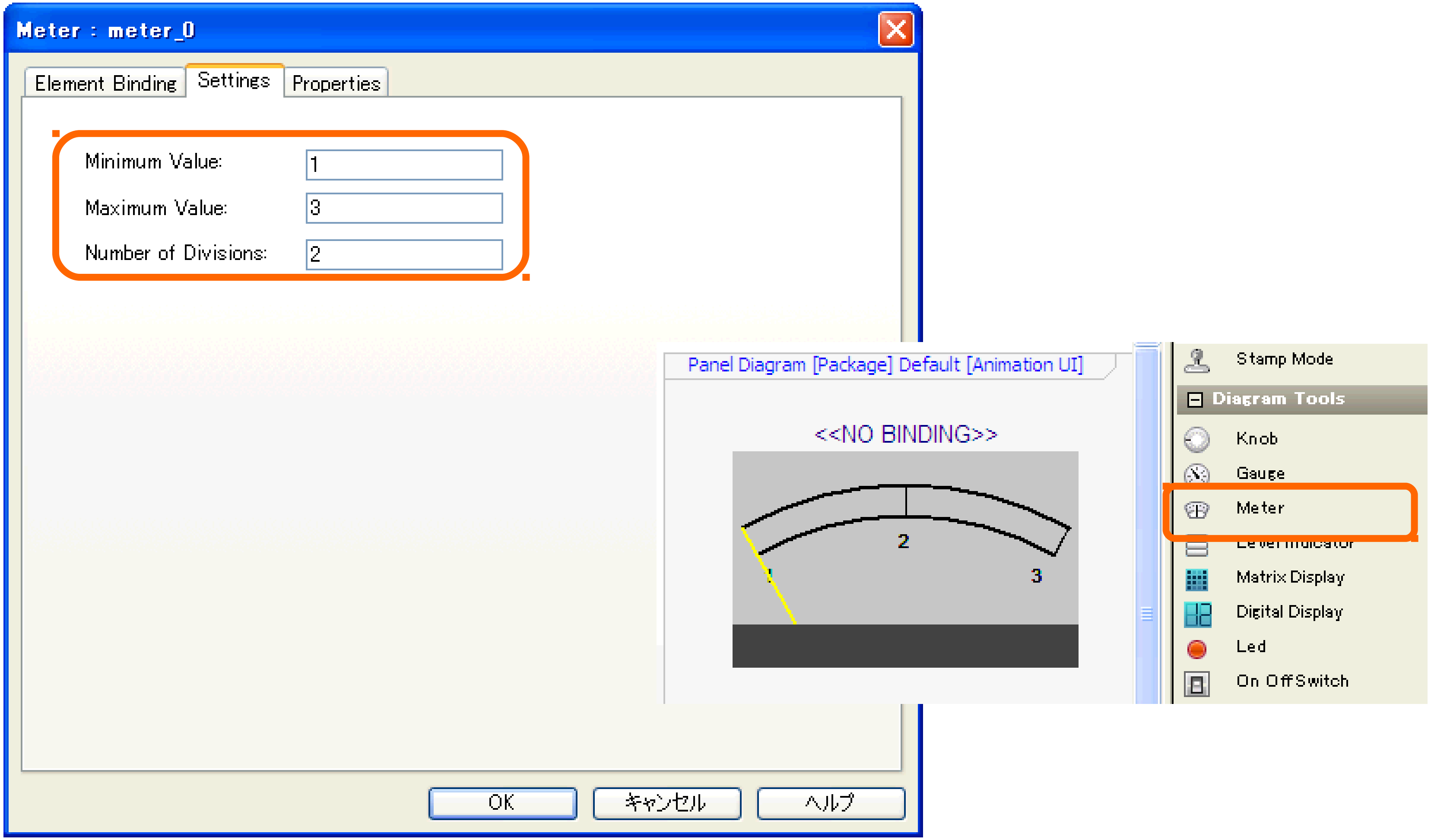
Creating a Panel Diagram (1/3)

- Create a Panel Diagram under Default package.



Creating a Panel Diagram (2/3)

- Place a Meter, and chang the Settings.



Meter : meter_0

Element Binding | Settings | Properties

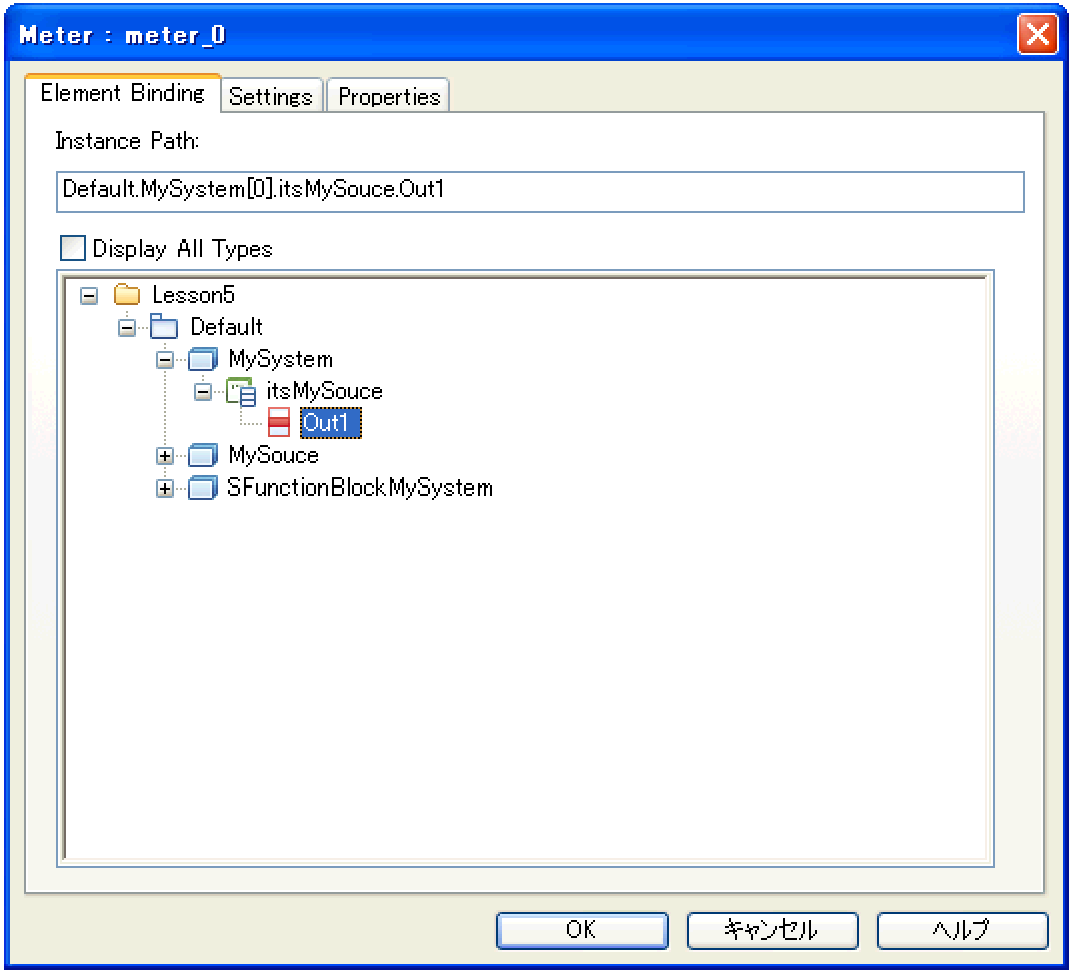
Minimum Value: 1

Maximum Value: 3

Number of Divisions: 2

Creating a Panel Diagram

- Bind an element.



Meter : meter_0

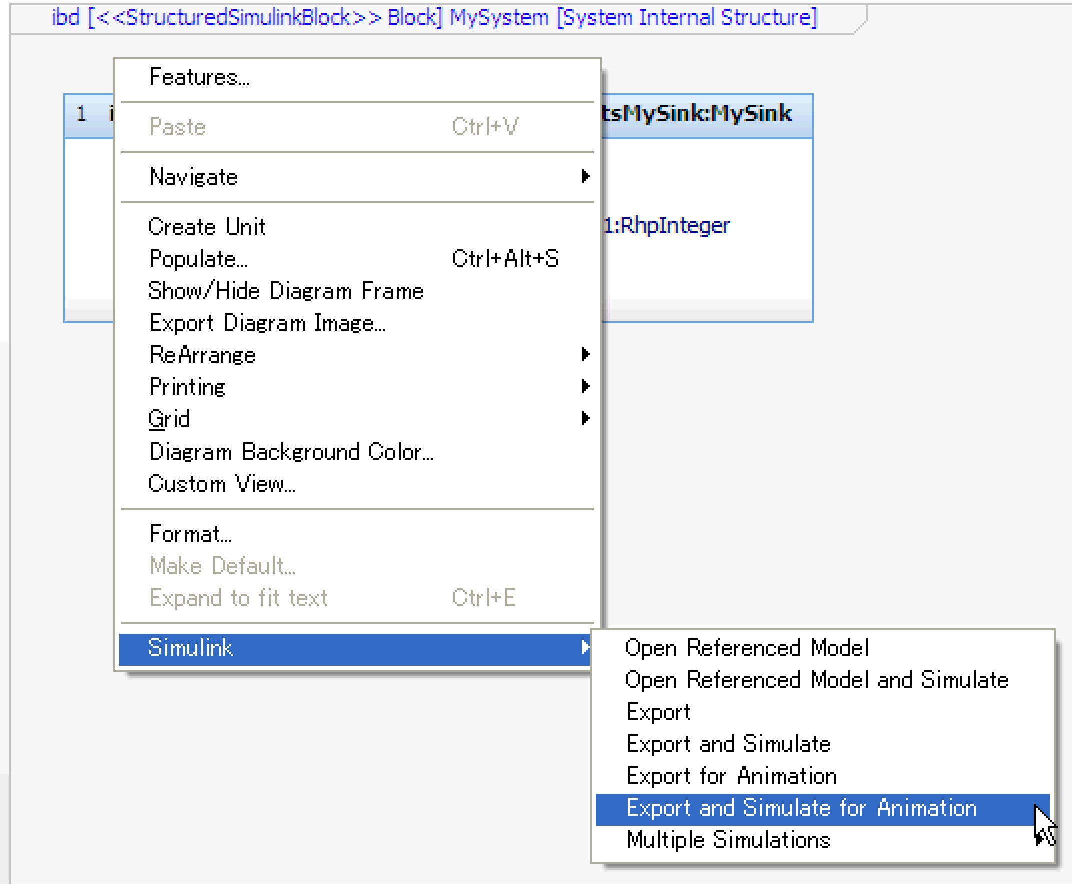
Instance Path:

Default.MySystem[0].itsMySouce.Out1

□ Display All Types

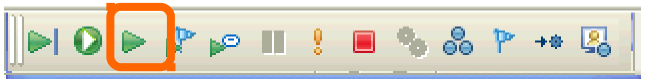
Animation (1/4)

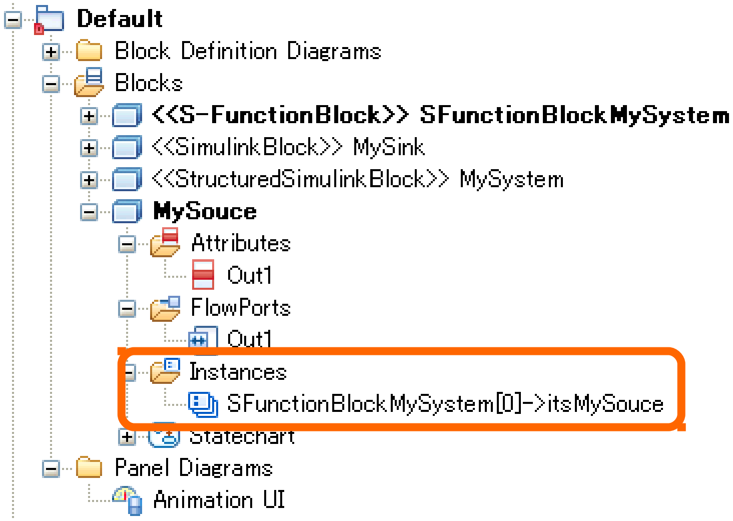
- Right-click on the IBD of the MySystem block, and select “Simulink”/”Export and Simulate for Animation”.



Animation (2/4)

- Icons for animation are appeared in the Rhapsody window.
- Click “Go Idol” icon.
  - You can see an instance of MySource.

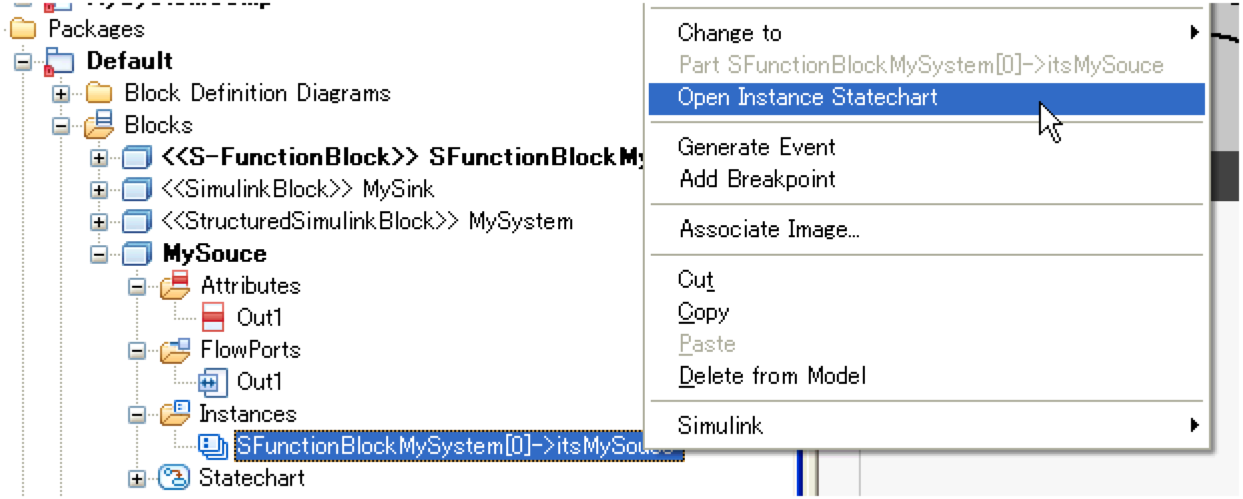




- Default
  - Block Definition Diagrams
  - Blocks
    - <<S-FunctionBlock>> SFunctionBlockMySystem
    - <<SimulinkBlock>> MySink
    - <<StructuredSimulinkBlock>> MySystem
    - MySouce
      - Attributes
        - Out1
      - FlowPorts
        - Out1
      - Instances
        - SFunctionBlockMySystem[0]->itsMySouce
      - Statechart
  - Panel Diagrams
    - Animation UI

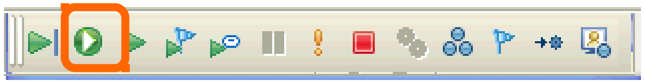
Animation (3/4)

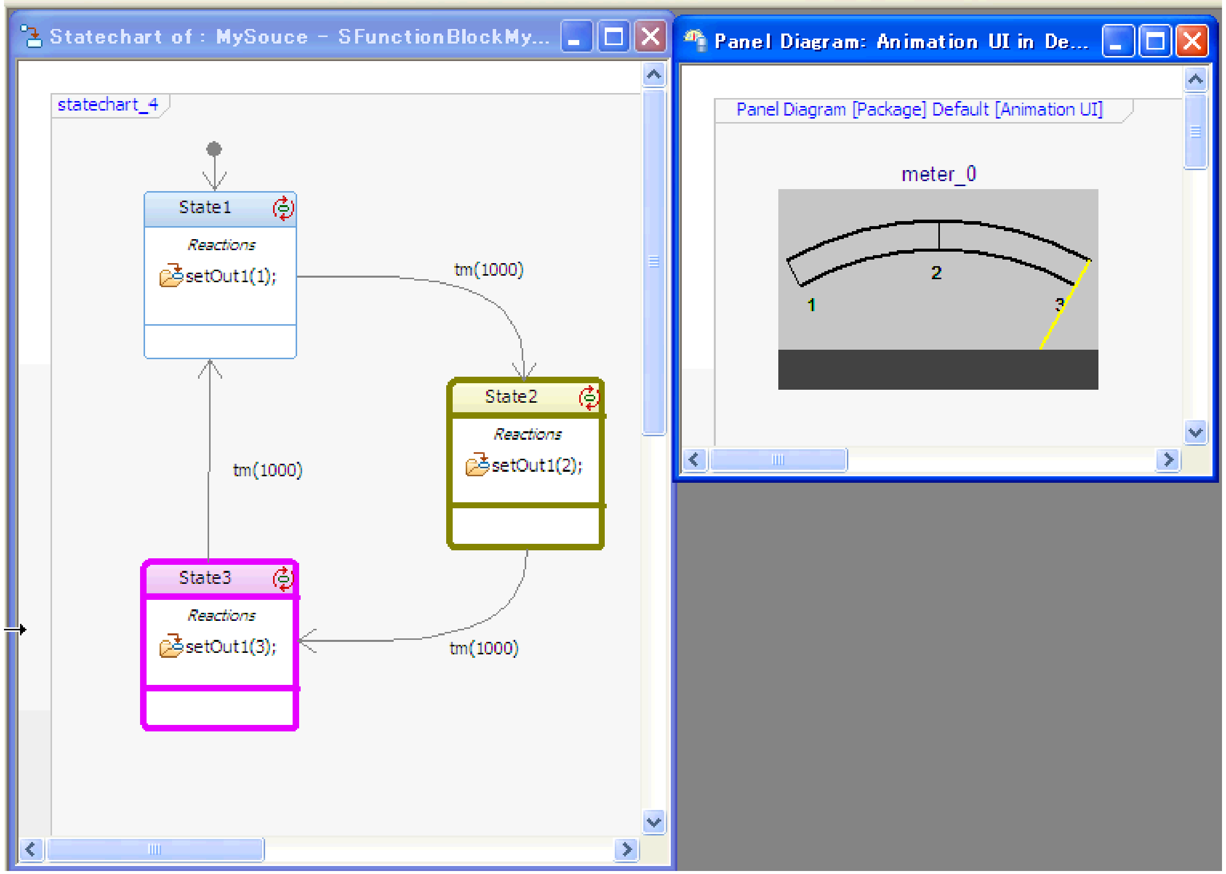
- Right-click on the instance of MySource, and select “Open Instance Statechart”.
  - You can see the instance statechart.



Animation (4/4)

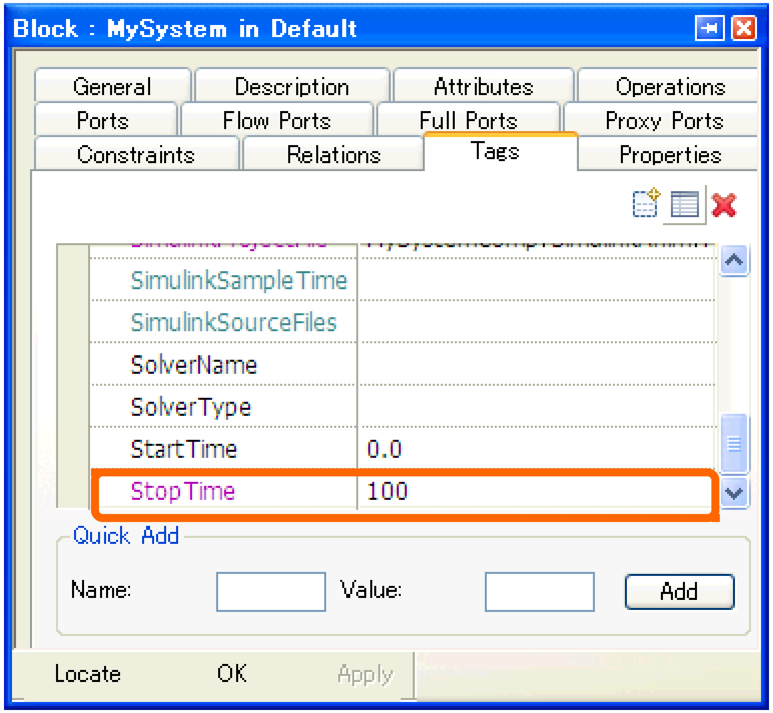
- Click “Go” icon.
  - You can see the instance statechart and the Panel Diagram are animated.




Statechart of : MySouce - SFunctionBlockMy...
statechart_4
State1
Reactions
setOut1(1);
tm(1000)
State2
Reactions
setOut1(2);
tm(1000)
State3
Reactions
setOut1(3);
tm(1000)
Panel Diagram: Animation UI in De...
Panel Diagram [Package] Default [Animation UI]
meter_0
1
2
3

Changing simulation stop time, and simulating the model

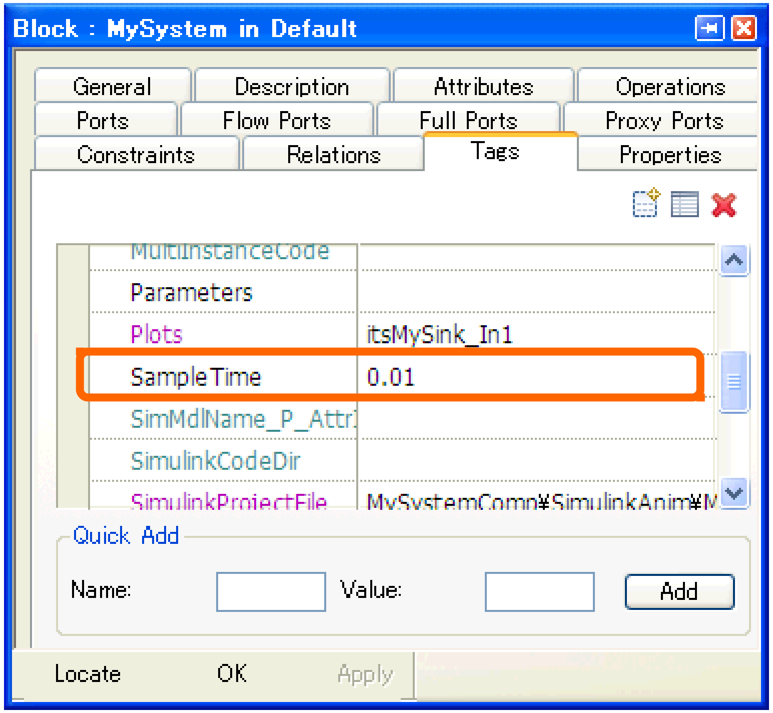
- Change the StopTime tag value of MySystem block to execute long simulation.
  – Note: The units of StartTime, StopTime, and SampleTime are seconds.
- Right-click on the MySystem, and select "Simulink"/"Open Referenced Model and Simulate".



SimulinkSampleTime	
SimulinkSourceFiles	
SolverName	
SolverType	
StartTime	0.0
StopTime	100

Sample time of an S-Function

- Sample time of an S-Function can be specified using SampleTime tag of <<StructuredSimulinkBlock>> stereotype.
- You can check and change the sample time by double-clicking the S-Function block on the exported Simulink model.
- Note: When you change the sample time, you have to perform “Export”. “Open Referenced Model and Simulate” does not change the sample time.
- Note: The unit of SampleTime is seconds.

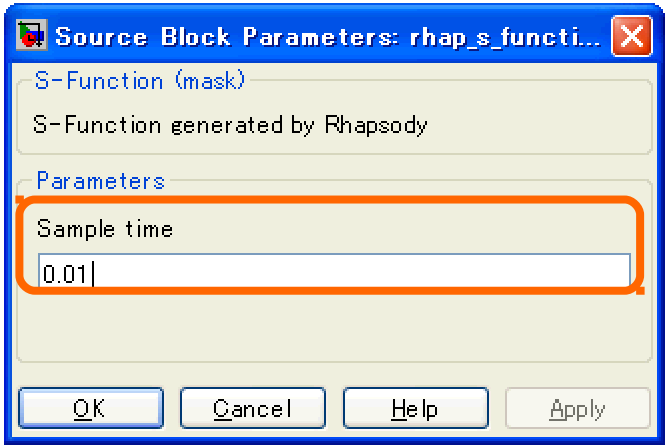


Block : MySystem in Default

MultiInstanceCode	
Parameters	
Plots	itsMySink_In1
SampleTime	0.01
SimMdlName_P_Attr	
SimulinkCodeDir	
SimulinkProjectFile	MySystemComp¥SimulinkAnim¥M

Quick Add

Name: Value:



Source Block Parameters: rhap_s_functi...

S-Function (mask)

S-Function generated by Rhapsody

Parameters

Sample time

0.01

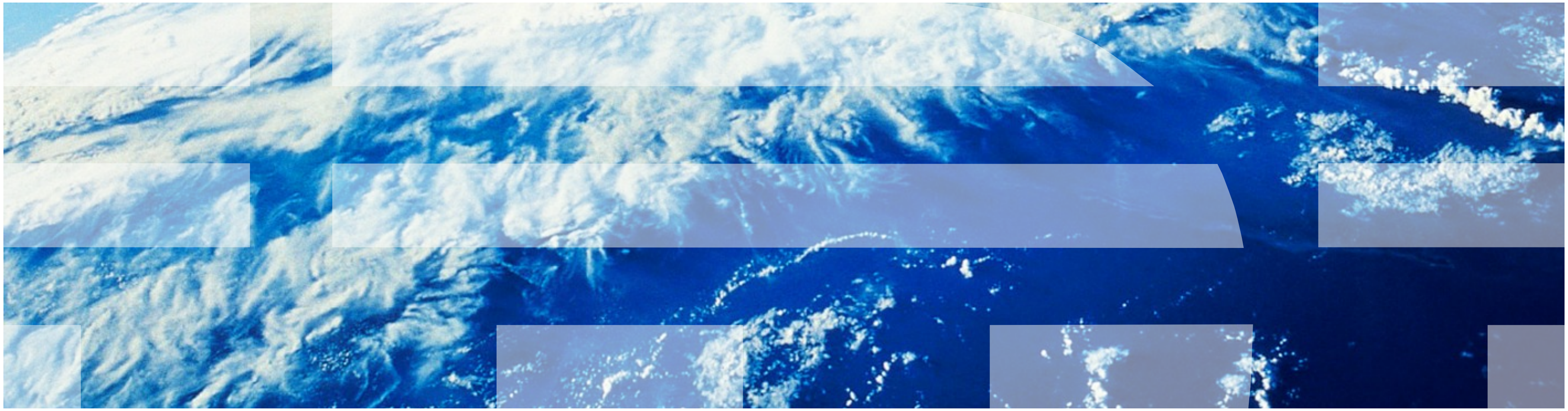
Limitations

- Rhapsody Architect for Systems Engineers cannot simulate an S-Function.
- Rhapsody Designer for Systems Engineers cannot simulate an S-Function without animation.
- Rhapsody can animate only one S-Function.
  - Each <<StructuredSimulinkBlock>> block generates its own S-Function if the block defines a behavior using SysML/UML (not Simulink).

Saving the project

- Save the project.

Lesson 6: Cruise Control system

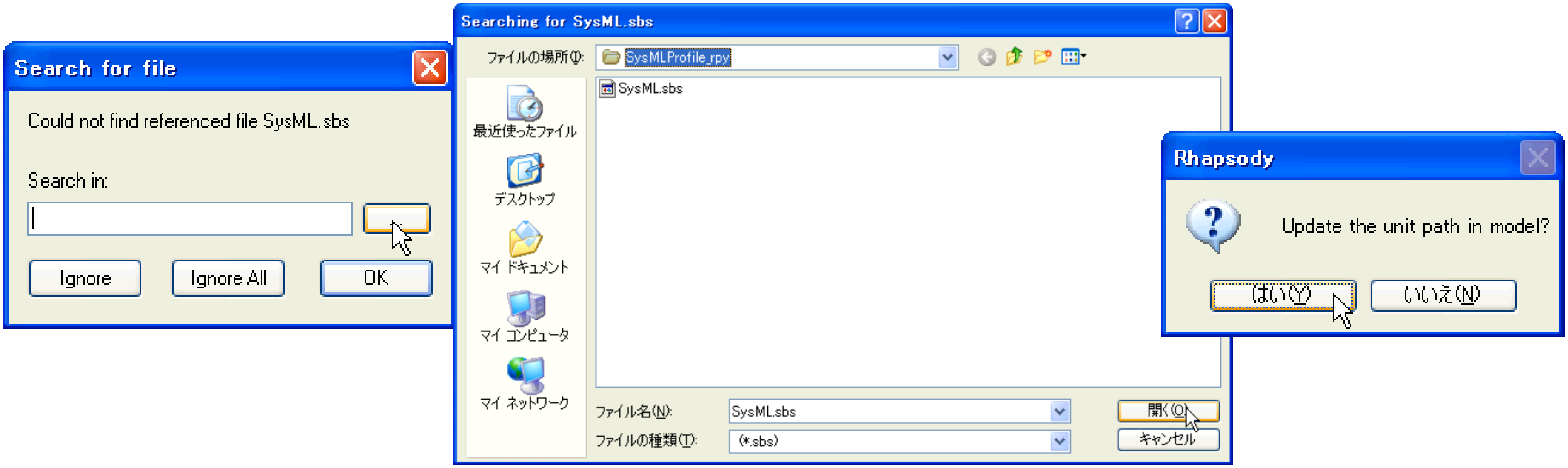


Contents of this lesson

- Try the sample Cruise Control System model.

Preparations

- Unzip “CruiseControlSystemPlantSim Sample Model V1.zip” to C:\Work\.
- Start Rhapsody Developer for C++.
  – Note: We cannot try Lesson 6 using Rhapsody Designer for Systems Engineers.
- Open “C:\Work\CruiseControlSystemPlantSim Sample Model V1\CruiseControlSystem.rpy”.
  – If you see the “Search for file” dialog, specify the SysML.sbs’s path.



Search for file

Could not find referenced file SysML.sbs

Search in:

Searching for SysML.sbs

ファイルの場所(I): SysMLProfile_rpy

SysML.sbs

ファイル名(N): SysML.sbs

ファイルの種類(T): (*.sbs)

Rhapsody

Update the unit path in model?

Setting up a compiler, an IDEVersion, and CPU

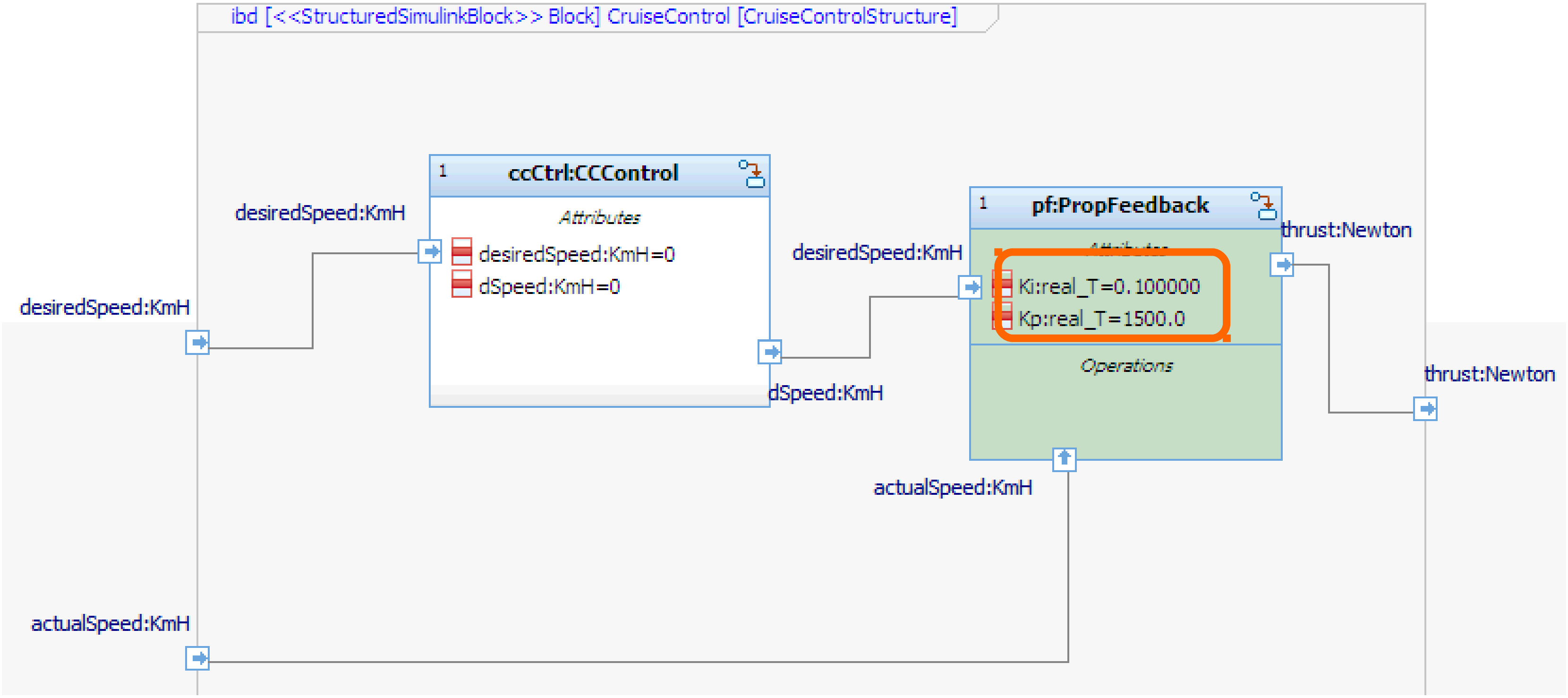
- Set up a compiler, an IDEVersion, and CPU as described in Lesson 5.

About this model

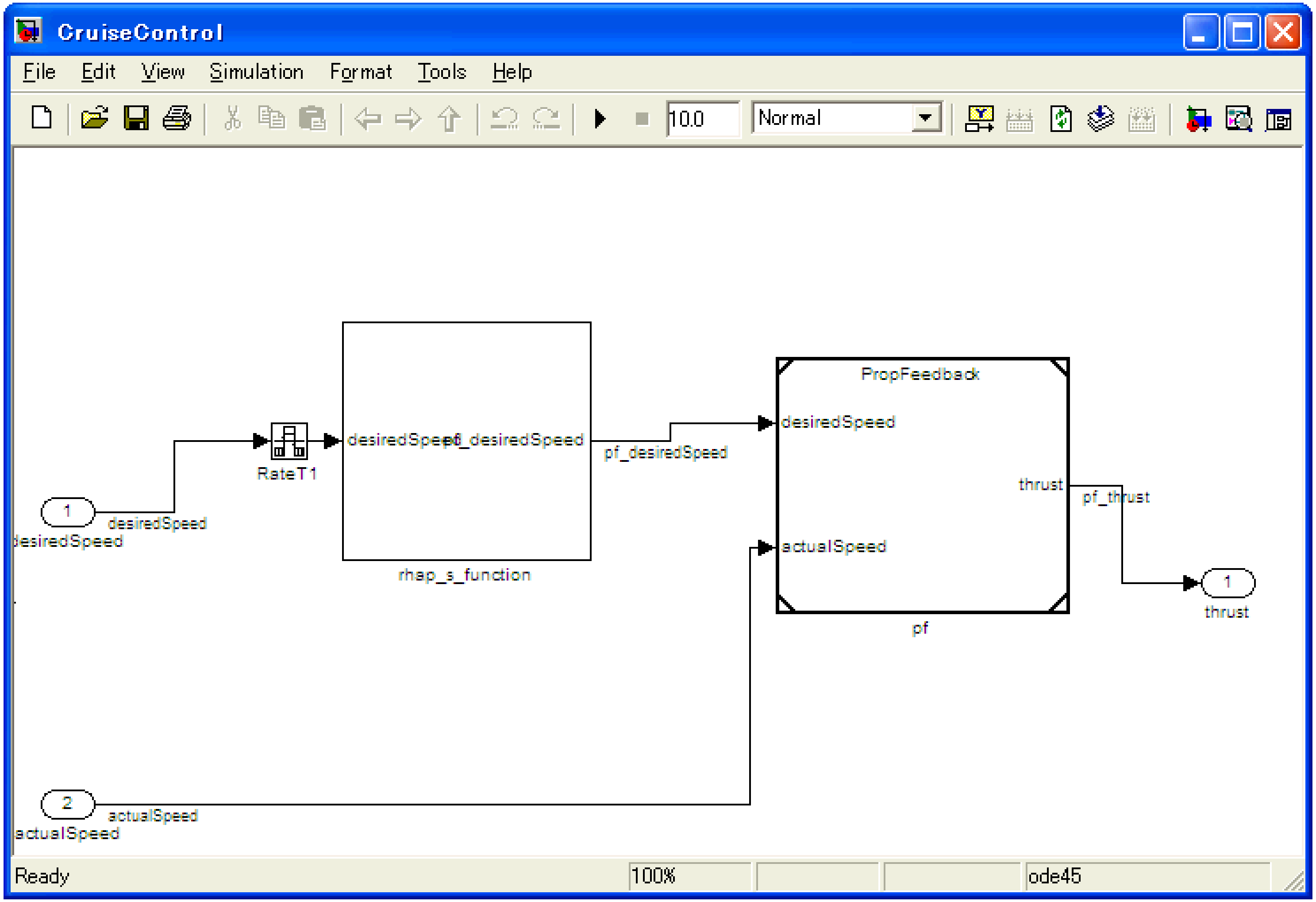
- This model contains three <<StructuredSimulinkBlock>> blocks.
  - CruiseCtrlSimDomain <<StructuredSimulinkBlock>> block has Vehicle block <<StructuredSimulinkBlock>> block
  - Vehicle <<StructuredSimulinkBlock>> block has CruiseControl <<StructuredSimulinkBlock>> block
- To export a Simulink model from a <<StructuredSimulinkBlock>> block, all Simulink models referred by the <<StructuredSimulinkBlock>> block must be exist.

Opening the IBD of CruiseControl block

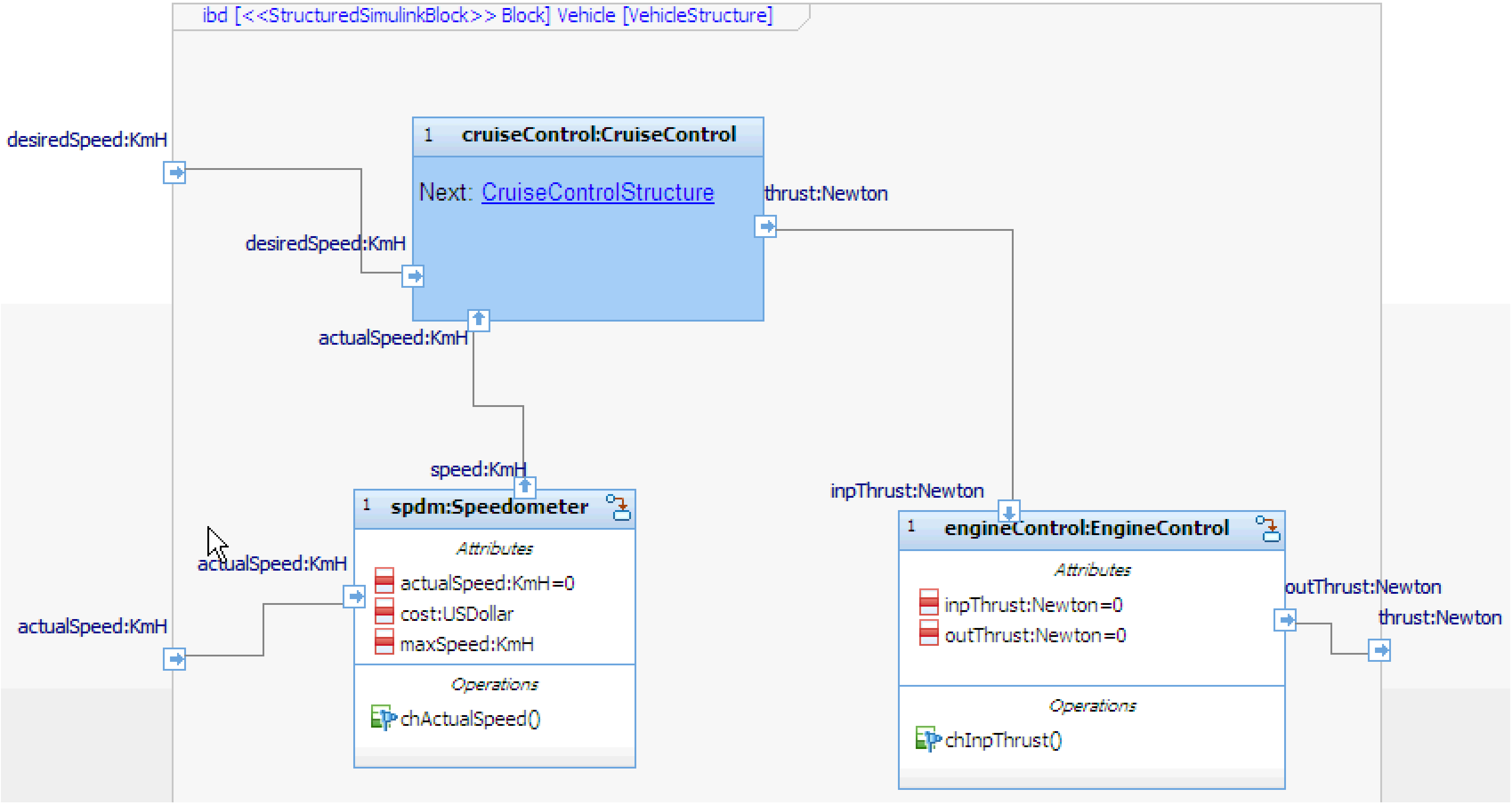
- Open the IBD of CruiseControl block under the CruiseControlSystemPkg.
  - Note: PropFeedback block has attributes Ki and Kp.



Exporting a Simulink model from CruiseCntrol block

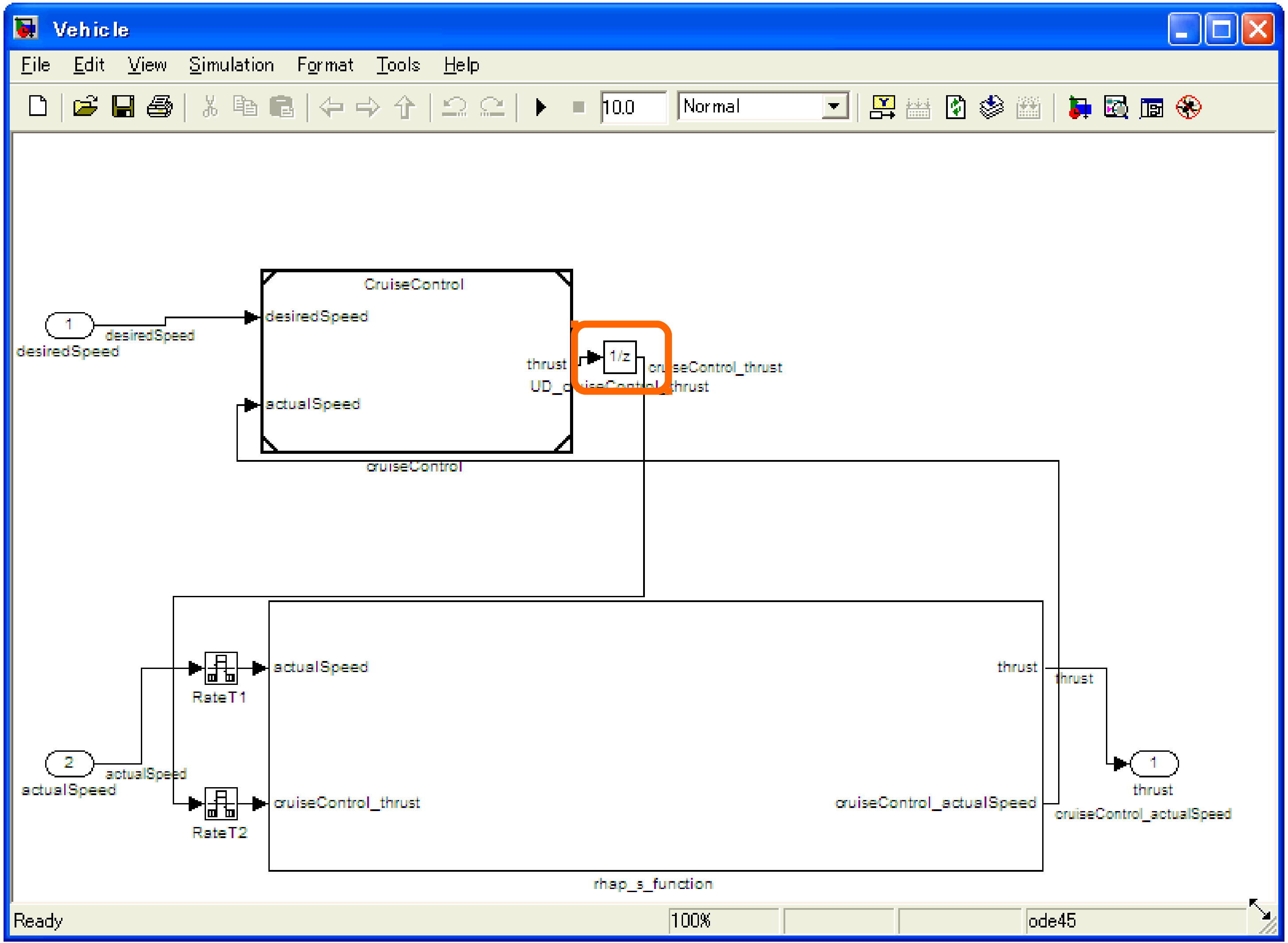


Opening the IBD of Vehicle block



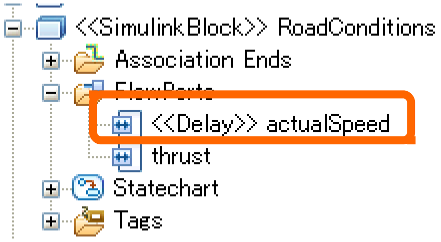
Exporting a Simulink model from Vehicle block

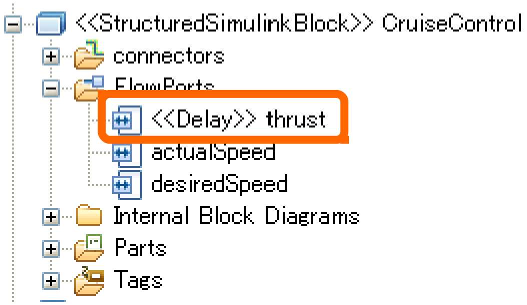
- Note: UnitDelay block is placed in this Simulink model.



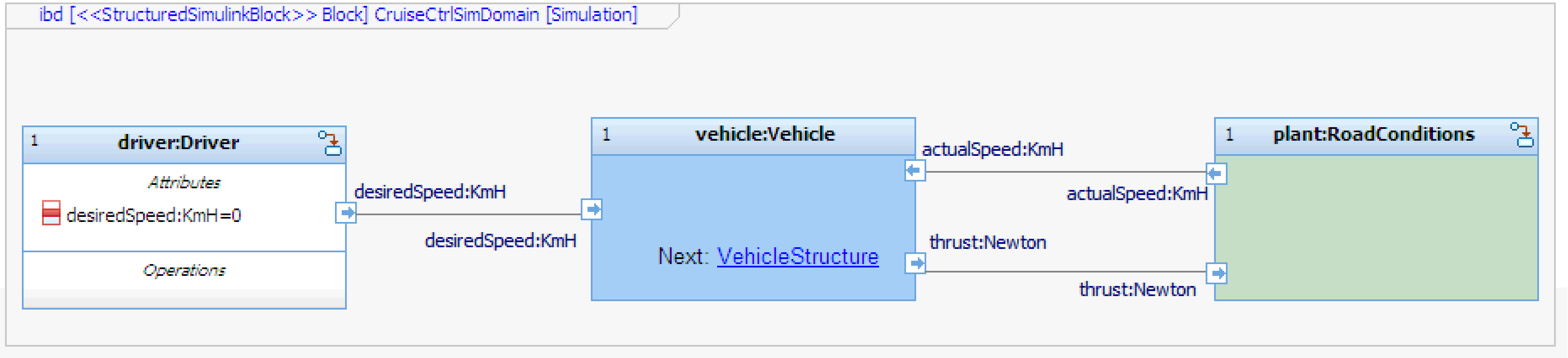
<<Delay>> stereotype

- This stereotype is applicable to a FlowPort. If you want to add a Unit Delay block to a generated Simulink model and link it to an output FlowPort, apply this stereotype to an output FlowPort owned by a <<SimulinkBlock>> block or a <<StructuredSimulinkBlock>> block.
- actualSpeed flow port of RoadConditions block and thrust flow port of CruiseControl block are applied <<Delay>> stereotype.

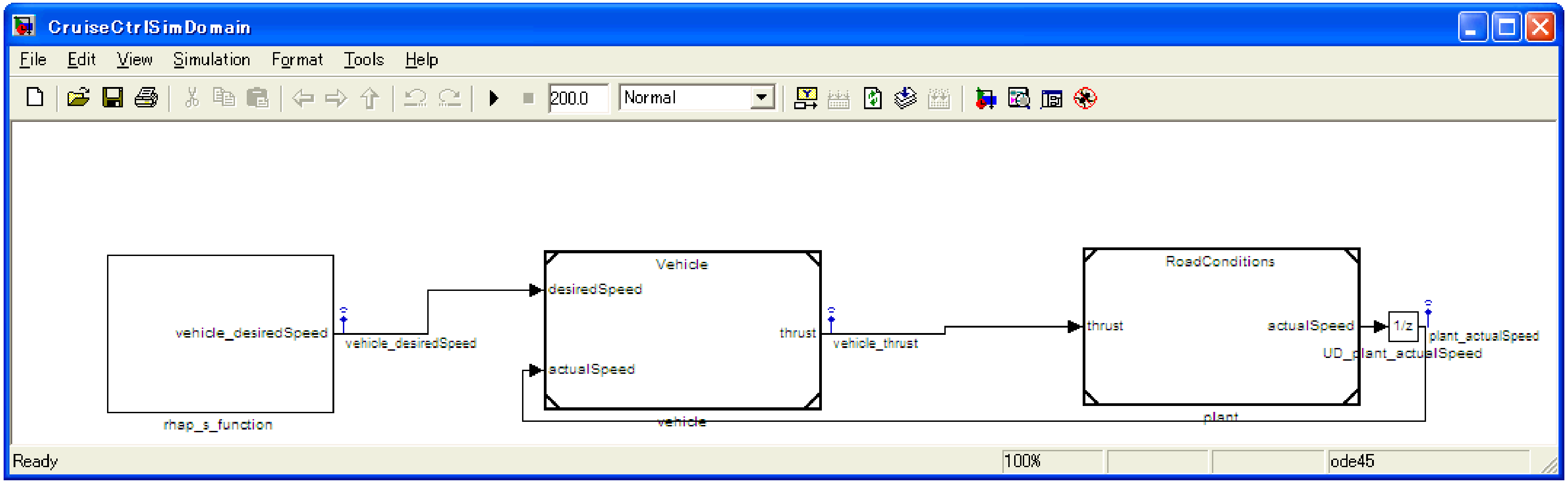




Opening the IBD of CruiseCtrlSimDomain block



Exporting a Simulink model from CruiseCtrlSimDomain block



200.0

Normal

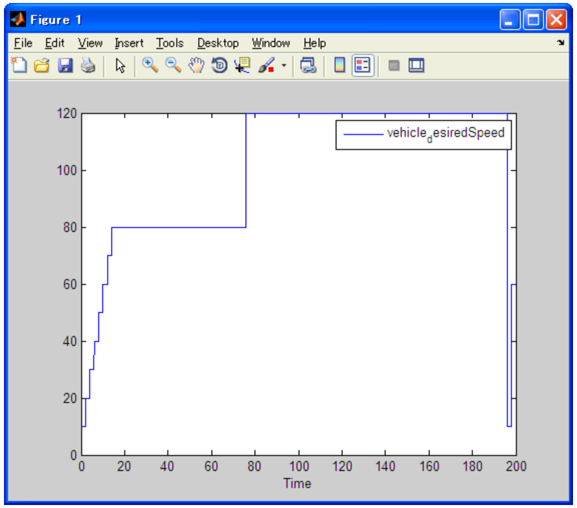
Ready

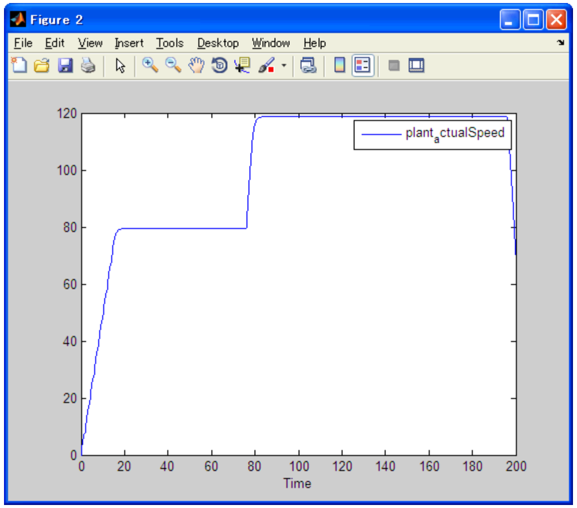
100%

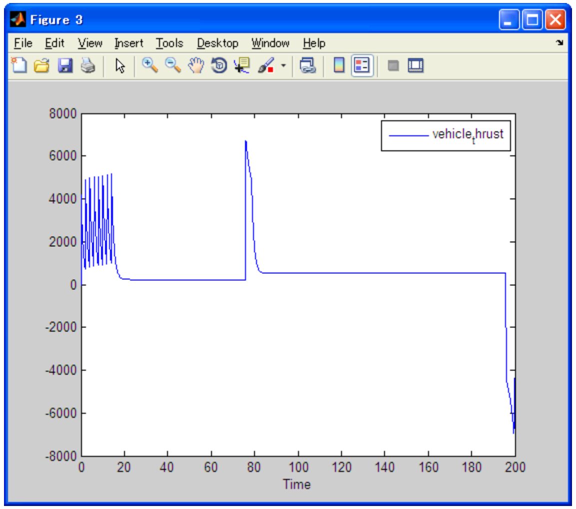
ode45

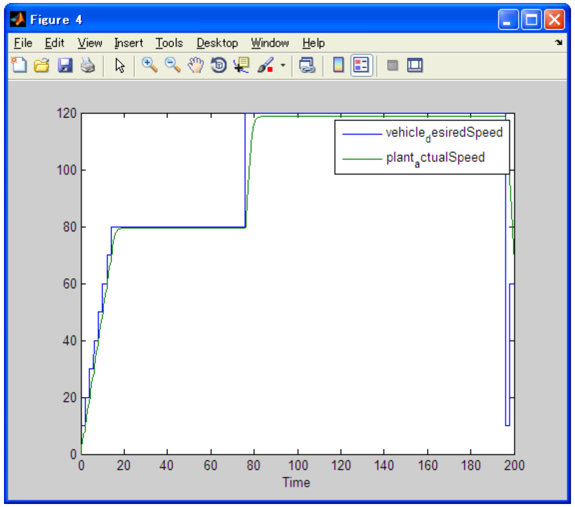
Simulate the CruiseCtrlSimDomain block

- Perform “Open Referenced Model and Simulate”.

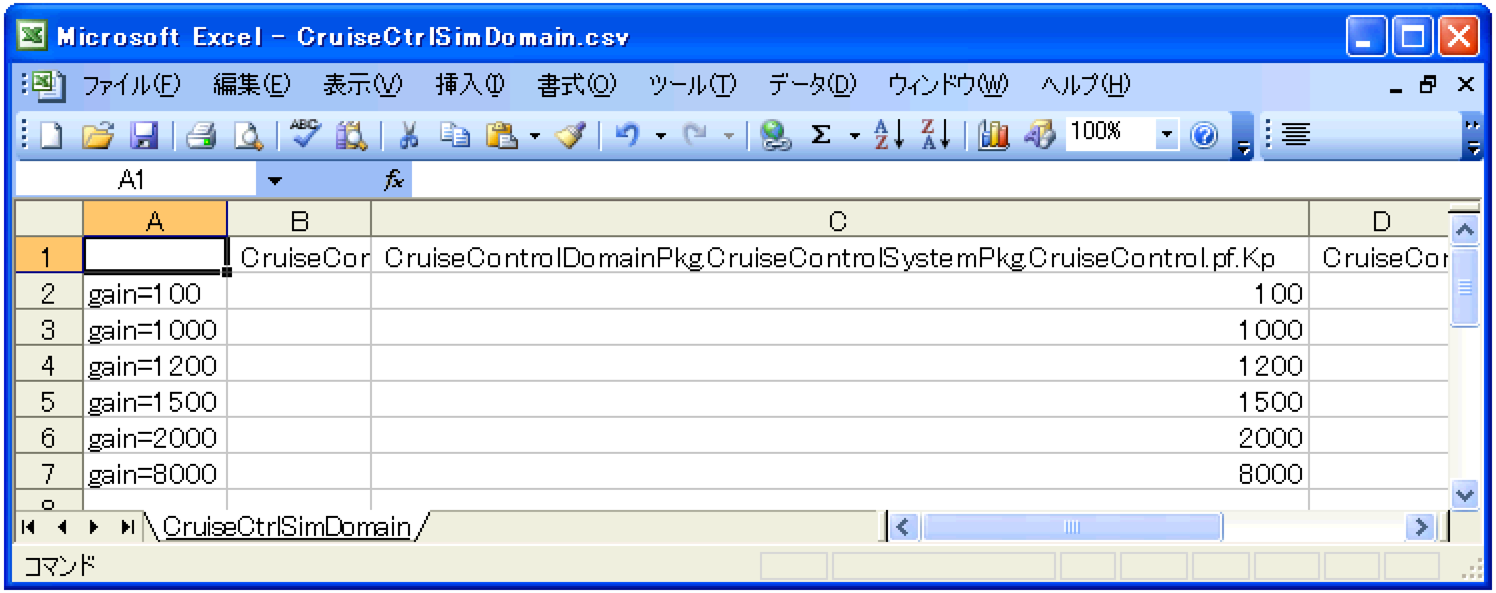








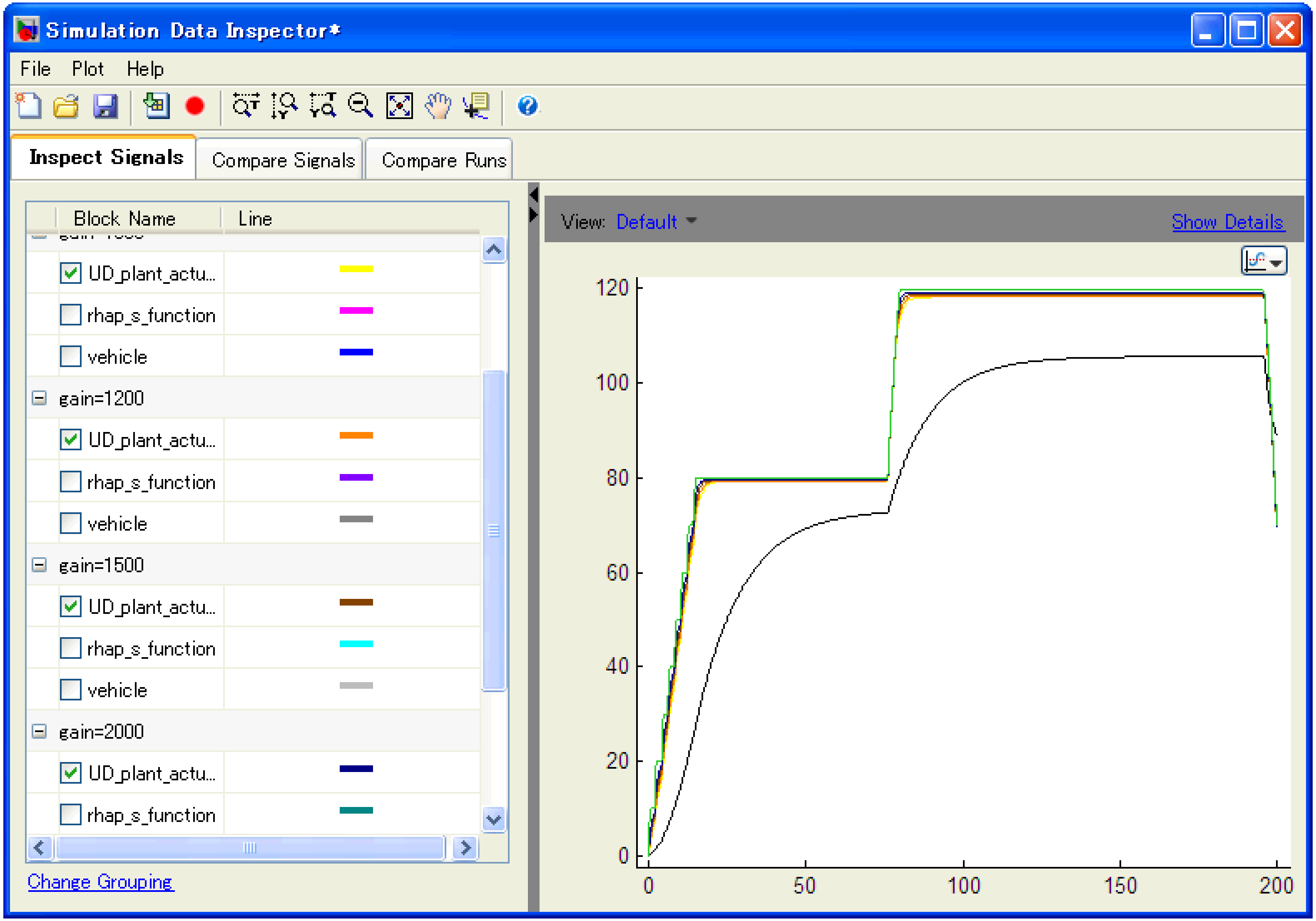
Opening the controlled file of CruiseCtrlSimDomain block



Microsoft Excel - CruiseCtrlSimDomain.csv
ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウィンドウ(W) ヘルプ(H)
100%
A1
fx
CruiseCtrlSimDomain
コマンド

	A	B	C	D
1		CruiseCor	CruiseControlDomainPkg.CruiseControlSystemPkg.CruiseControl.pf.Kp	CruiseCor
2	gain=100		100	
3	gain=1000		1000	
4	gain=1200		1200	
5	gain=1500		1500	
6	gain=2000		2000	
7	gain=8000		8000	

Simulating using CSV



Block Name	Line
☑ UD_plant_actu...	🟨
□ rhap_s_function	🟪
□ vehicle	🟦
⊟ gain=1200	
☑ UD_plant_actu...	🟧
□ rhap_s_function	🟪
□ vehicle	⬛
⊟ gain=1500	
☑ UD_plant_actu...	🟫
□ rhap_s_function	🟦
□ vehicle	⬜
⊟ gain=2000	
☑ UD_plant_actu...	🟦
□ rhap_s_function	🟩

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.