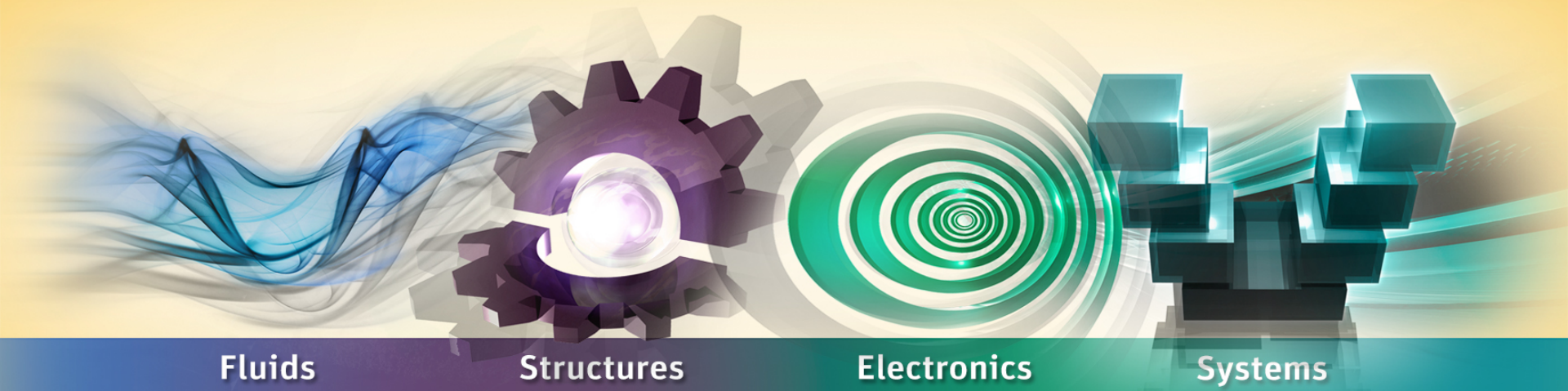


SCADE TRAINING



**SCADE Suite co-simulation in the
Simplorer Environment**



1. Course Introduction
2. SCADE Suite Code Generation Workflow for Simulation in Simplorer
3. Import of the SCADE Suite-generated DLL in the Simplorer Environment
4. Black-Box Co-simulation Example
5. White-Box Co-simulation Example

INTRODUCTION

Training Objectives

Prerequisites & Training Content

Co-simulation principles

Learn how to use the smooth coupling between SCADE® software models and ANSYS® Simplorer® simulation environment, including:

- Co-Simulation Principles
- SCADE Suite Code Generation Workflow for Simulation in Simplorer
- Import of the SCADE Suite-generated Dynamic Link Library (DLL) in the Simplorer Environment
- Execution of an example in Black-Box and White-Box co-simulation modes in the Simplorer Environment

INTRODUCTION

Training Objectives

Prerequisites & Training Content

Co-simulation principles

This coupling uses the SCADE® Suite™ and ANSYS® Simplorer® product line

This training relies on the following knowledge:

- SCADE® Suite™
- ANSYS® Simplorer®

Start with an introduction to the Co-Simulation Principles

Then, present the Coupling Workflow Architecture

Learn how to use the Simplorer Target in SCADE Suite and import of SCADE Suite-generated DLL in the Simplorer Environment

Finally, execute an example in the Black-Box and White-Box Co-simulation modes in Simplorer

INTRODUCTION

Training Objectives

Prerequisites & Training Content

Co-simulation principles

SCADE Suite co-simulation capabilities enable the simulation of SCADE Suite models into external simulation applications

Have access to built-in co-simulation with various development tools and environments

Have the possibility to extend other external applications to take remote control over SCADE Suite Simulator using a set of simulation API C functions

In SCADE Suite, co-simulation can operate in either of the following modes, Black-box and White-box modes:

- Black-box co-simulation:
 - Encapsulate the KCG code which is generated from SCADE Suite models into dedicated interface C code layers required by the external application
 - Execute the SCADE Suite-generated KCG code directly within the external application

White-box co-simulation:

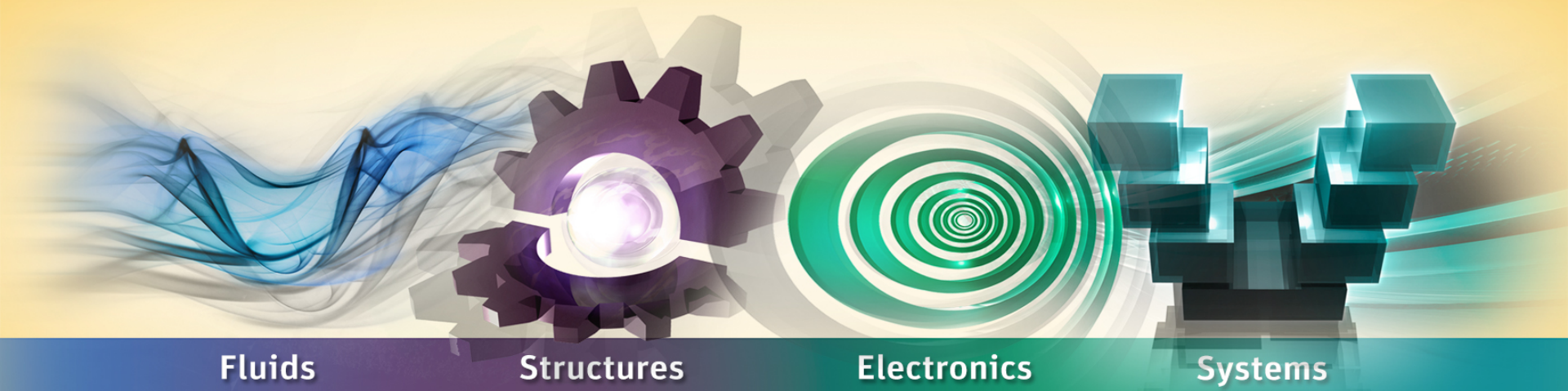
Generate SCADE Suite simulation code in a Windows DLL

Generate a separate adaptation C code layer for the external application

Interact with the SCADE Suite simulator:

- The adaptation layer forwards model inputs to SCADE
- Suite Simulator and sends model outputs to the
- external application through SCADE Suite simulation
- API C functions

SCADE TRAINING



SCADE Suite Code Generation Workflow for Simulation in Simplorer

SCADE SUITE CODE GENERATION WORKFLOW FOR SIMULATION IN SIMPLORER

Coupling Workflow Architecture

SCADE Suite GUI Overview

Setting Code Generator

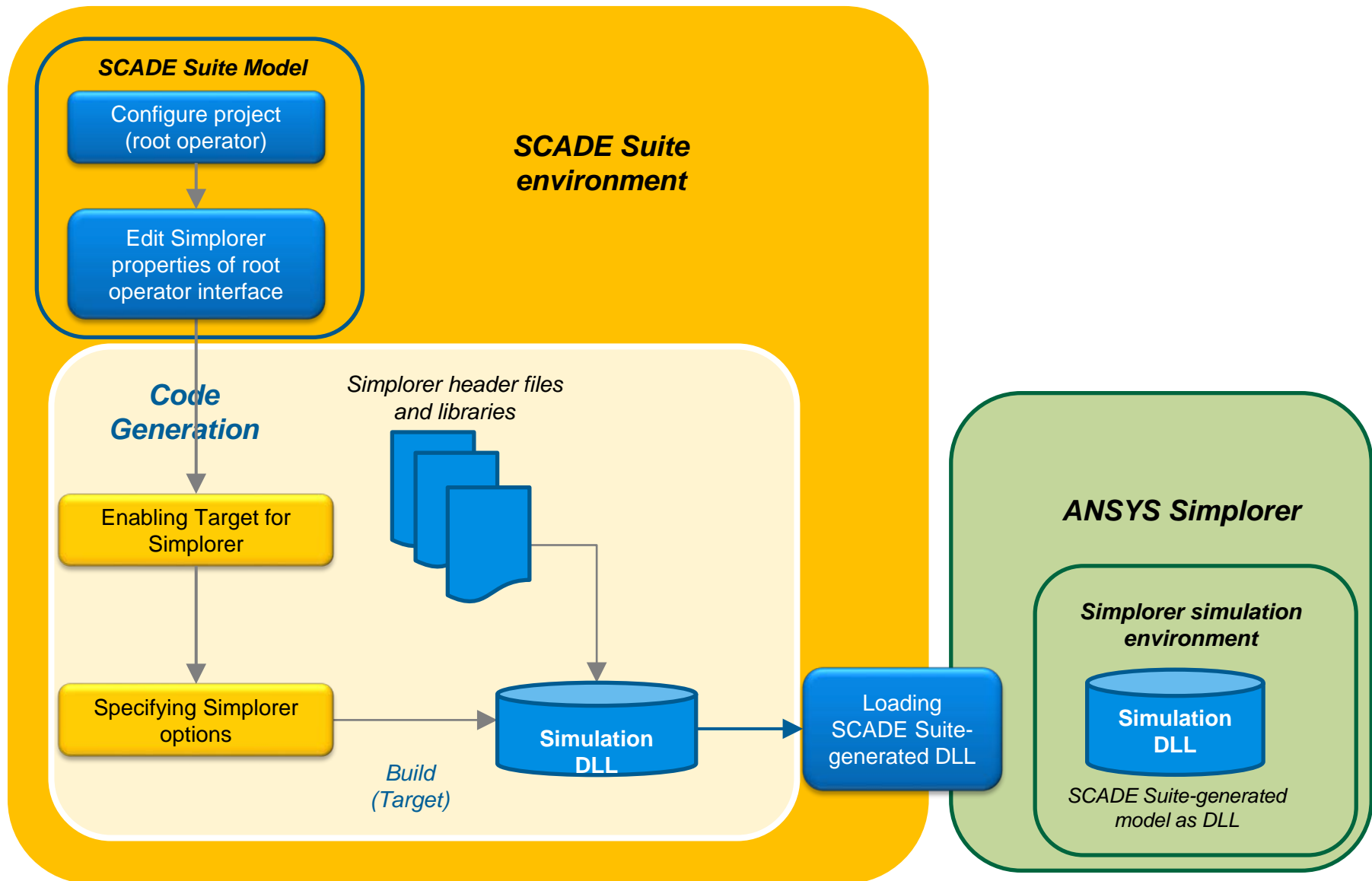
Exporting SCADE Suite Model

SCADE Suite solution for Simplorer generates a Windows DLL (Visual C compiler)

for integrating SCADE Suite models into

ANSYS Simplorer simulation models:

- Enable Black-box and White-box co-simulation capabilities
- DLL embeds the SCADE Suite KCG generated code
- DLL based on the Software Development Kit (SDK) provided by Simplorer



Use SCADE Suite Code Integration Target for Simplorer to generate, compile, and link SCADE Suite KCG code with all Simplorer files necessary to run the co-simulation of SCADE Suite models within Simplorer environment

The DLL publishes as many Simplorer C-models as there are root operators declared for the code generation in SCADE Suite (for Black-Box mode only)

This Code wrapper faces following constraint to embed a SCADE operator generated code into a Simplorer DLL:

- The imported types are not supported for interfaces: sensors, inputs and outputs

SCADE SUITE CODE GENERATION WORKFLOW FOR SIMULATION IN SIMPLORER

Coupling Workflow Architecture

SCADE Suite GUI Overview

Setting Code Generator

Exporting SCADE Suite Model

SCADE Suite GUI Overview: Simplorer properties

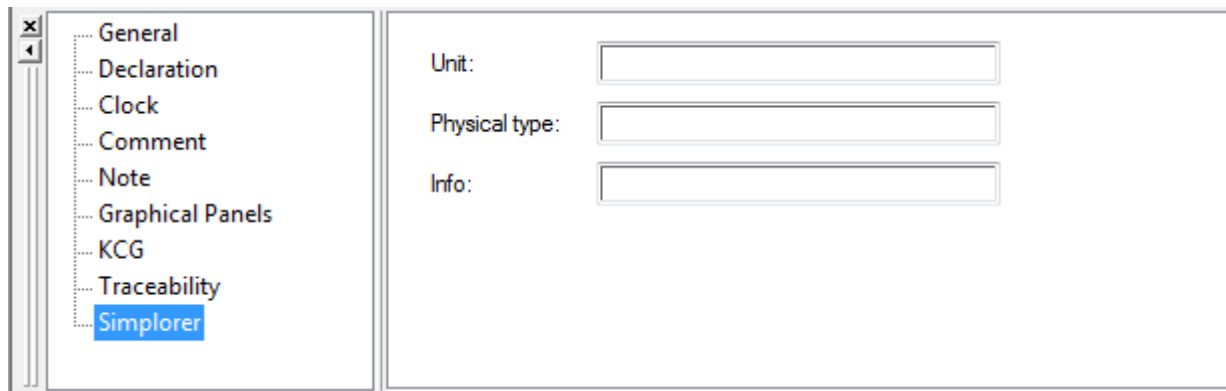
Simplorer allows defining properties for the sensors and the inputs/outputs of an SCADE operator:

- A property page named “Simplorer” is available to set these following properties into SCADE Suite:
 - Unit: String representing the unit of the quantity in or out of the model
 - Physical type: String representing the physical type of the quantity in or out of the model
 - Info: Info line to be displayed in Simplorer Schematic and Model Agent
- These properties are string values which may remain empty

SCADE Suite GUI Overview: Simplorer properties

To set these properties from the GUI

- Right-click an item from an operator interface definition or a sensor in the Framework View
- Select its Properties Box
- Select Simplorer item on the left and set values on the right



- Example:

Unit	m/s
Physical type	Speed
Info	Car speed

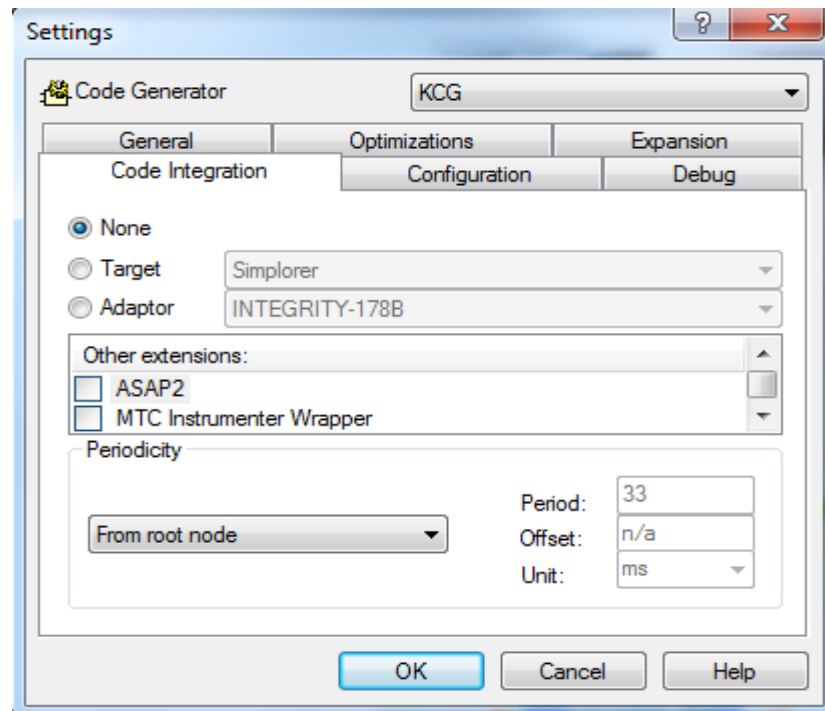
SCADE Suite GUI Overview: Code Generator settings

The module of SCADE Suite Wrapper for Simplorer is a Code Generator wrapper specified as Target:

Select **Settings** or select **Project - Code Generator – Settings** in SCADE Suite menu to open the Settings window



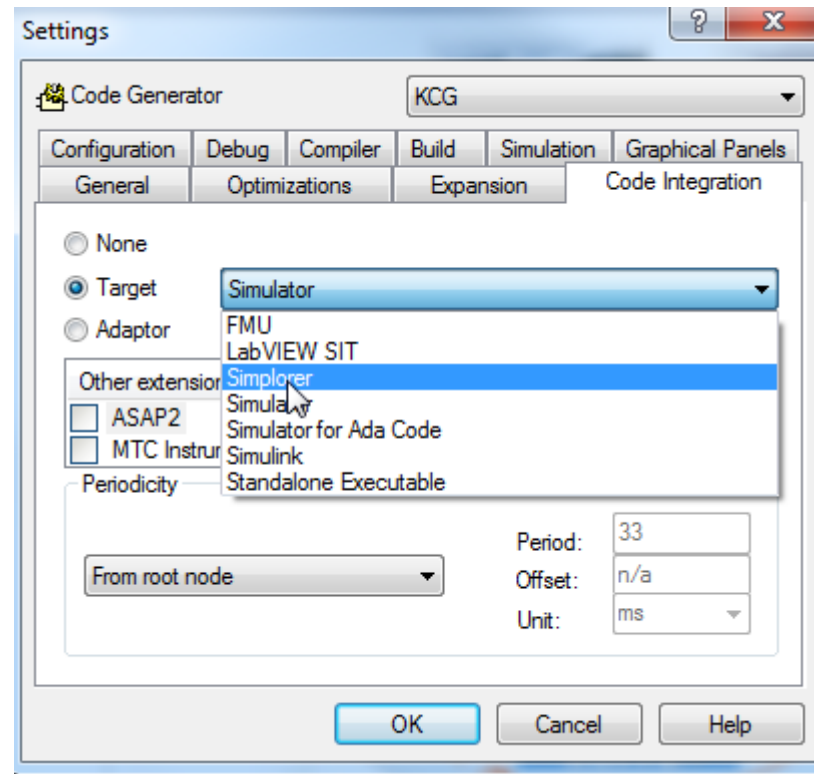
Code Generator Toolbar



Code Generator Settings Window

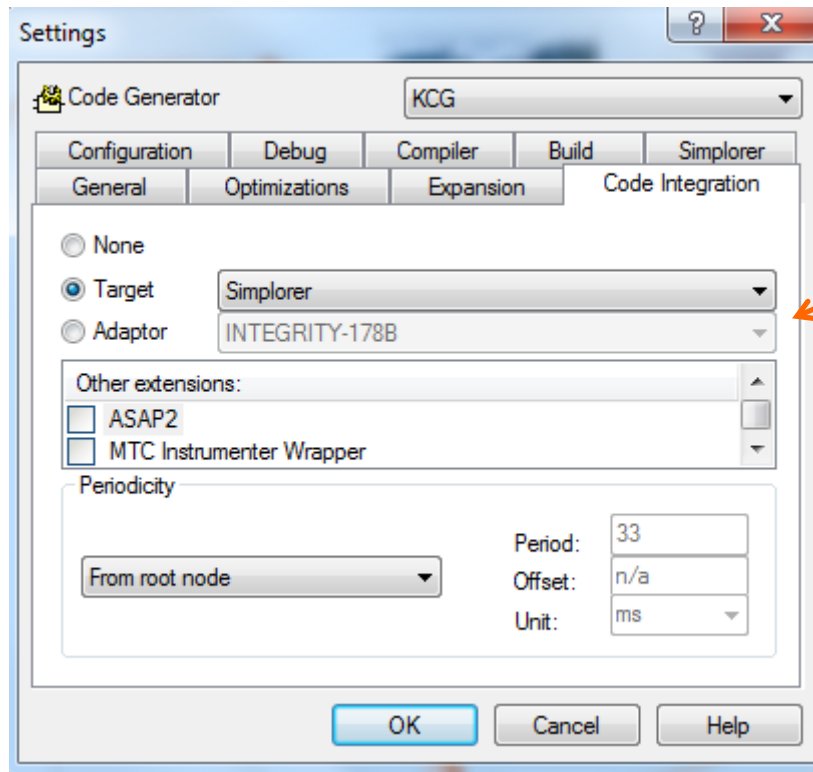
SCADE Suite GUI Overview: Code Generator settings

A Code Generator target named “Simplorer” is available in the list of targets, in the “ Code Integration ” tab



SCADE Suite GUI Overview: Code Generator settings

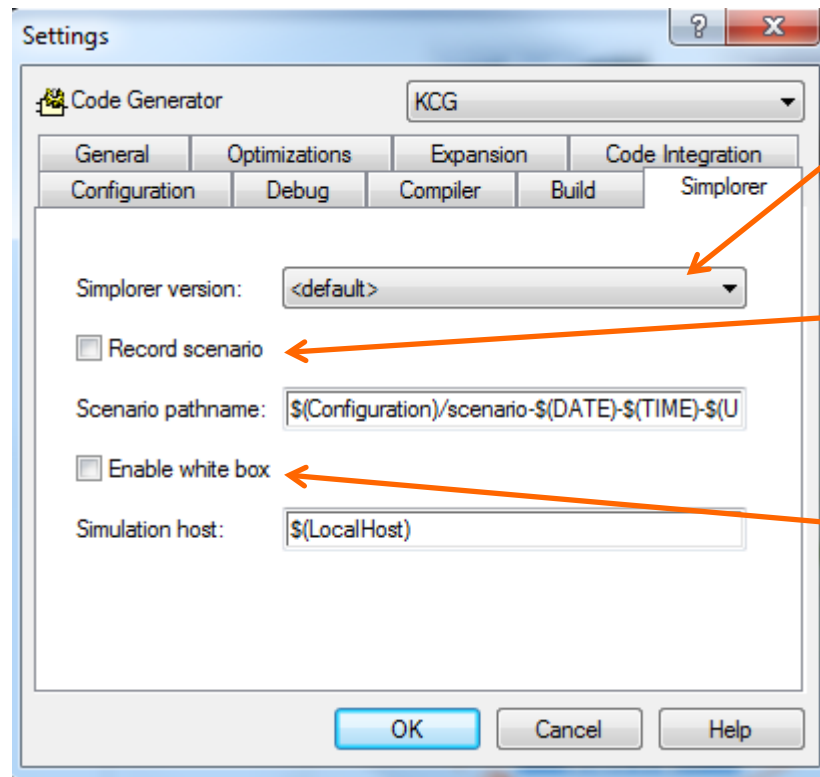
When the “Simplorer” Target is checked



a **Settings** page named **Simplorer** is available

SCADE Suite GUI Overview: Code Generator settings

The Simplorer Settings page provides three options:



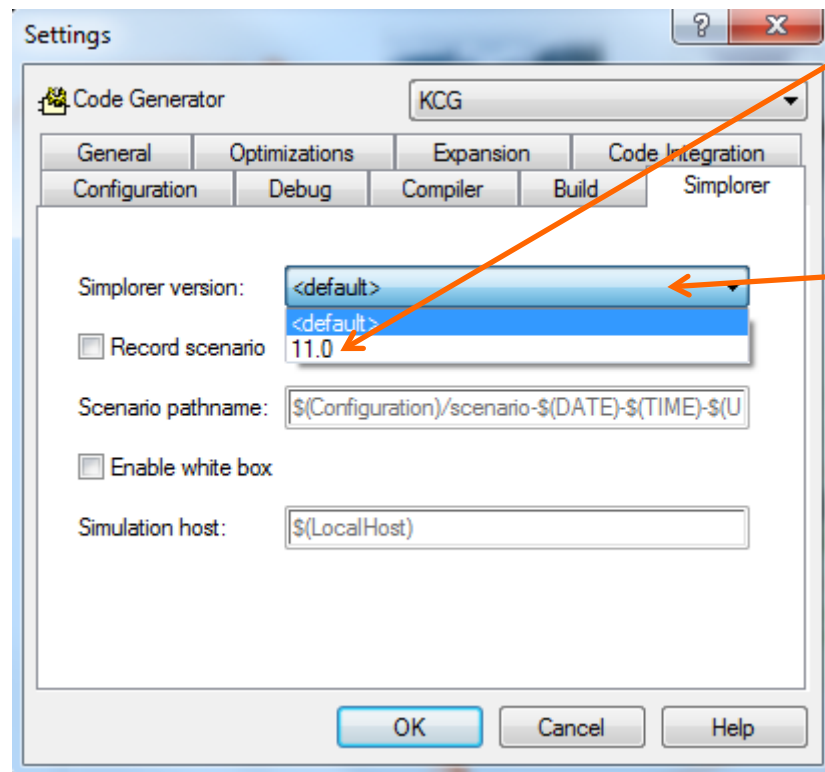
Simplorer version

Record scenario

Enable white box

SCADE Suite GUI Overview: Code Generator settings

“Simplorer version” option displays all installed versions of Simplorer and a generic choice named “<default>”



All Simplorer versions

When **<default>** is selected, the most recent installed version of Simplorer is used

SCADE Suite GUI Overview: Code Generator settings

Check “Record scenario” to record the input scenarios (.in format) during simulation

When checked, this option activates a text area to enter the pattern of the **scenario pathnames** to be generated

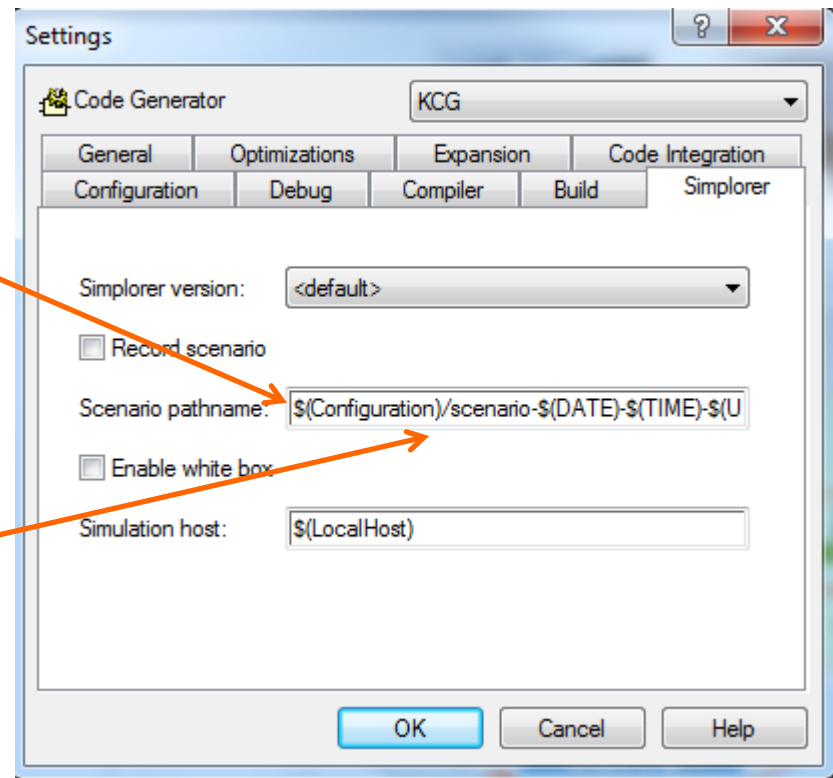
The pattern can use the following macros:

\$(Configuration): current configuration name

\$(DATE): creation date of the scenario file

\$(TIME): creation time of the scenario file

\$(USE): Operator instance name within Simplorer

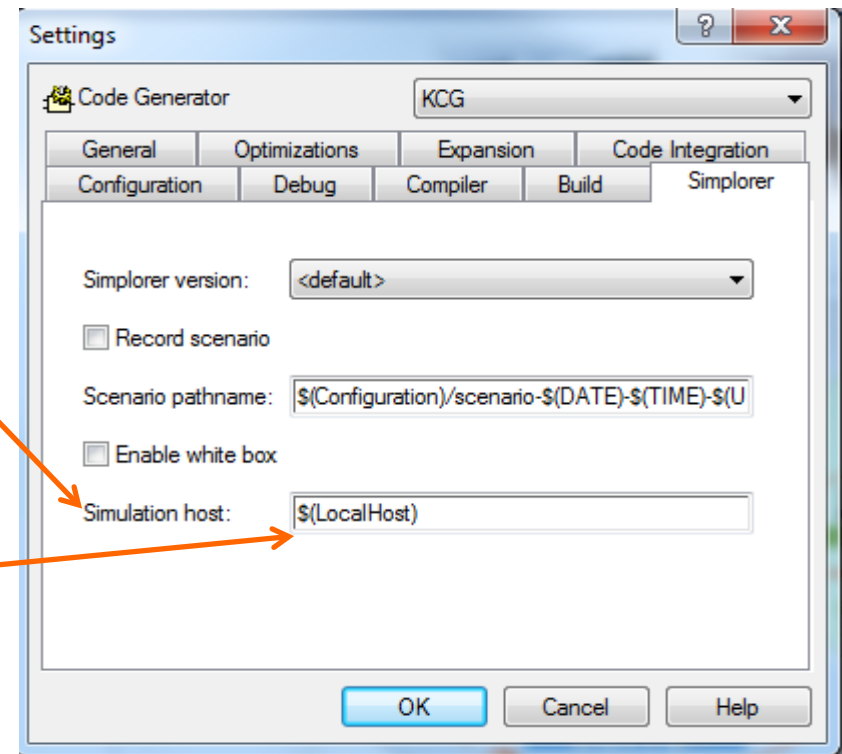


SCADE Suite GUI Overview: Code Generator settings

Check “ Enable white box ” if you want to run the SCADE Suite model in White-box co-simulation or uncheck this option if you want to run Black-box co-simulation

When checked, this option activates a text area to enter the **IP address of the co-simulation host**

The value can use the following macro:
\$(LocalHost): Translated to
127.0.0.1.



SCADE SUITE CODE GENERATION WORKFLOW FOR SIMULATION IN SIMPLORER

Coupling Workflow Architecture

SCADE Suite GUI Overview

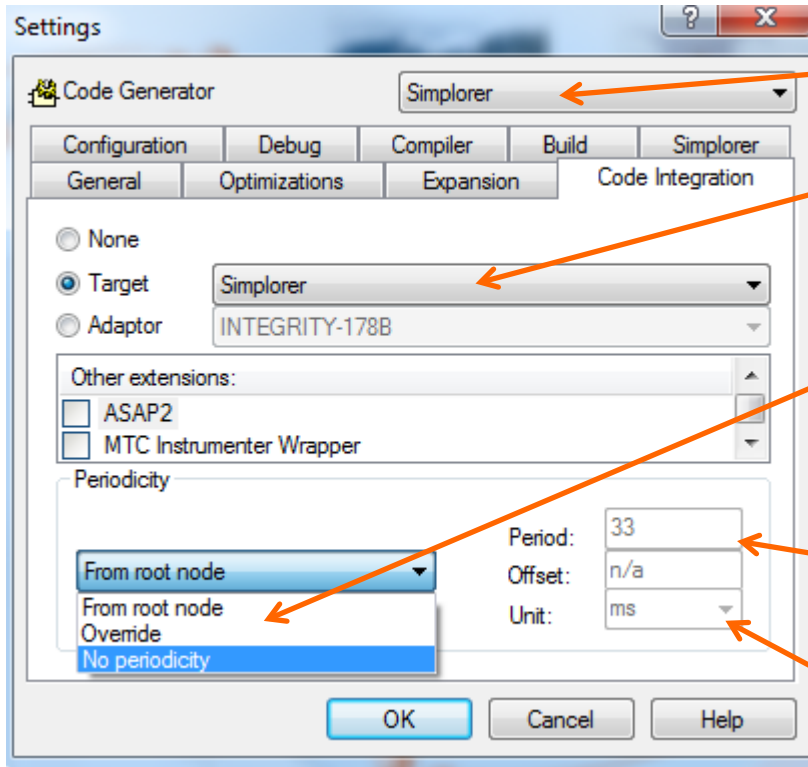
Setting Code Generator

Exporting SCADE Suite Model

To generate SCADE Suite model for Simplorer simulation, you must enable code generation and specify previous options from SCADE Suite Code Generator Settings

Create a custom configuration (named “Simplorer” for example)

Setting Code Generator



Simplorer configuration

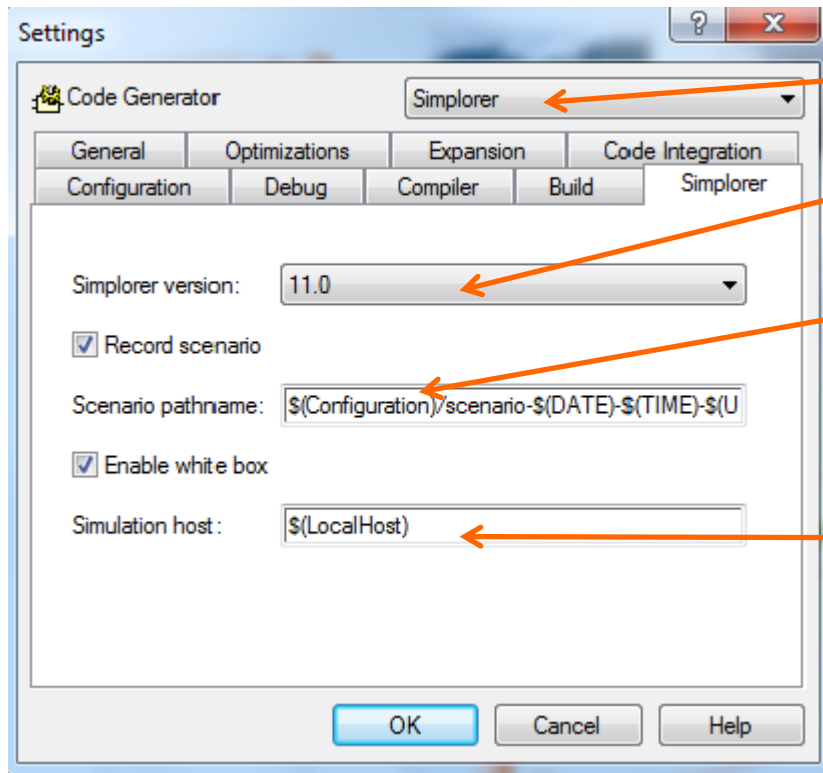
Simplorer Target

Periodicity: select from the drop-down list following options:
From root node: code execution is based on the root node periodicity **Override:** enables the periodic code execution by overriding periodicity specified at root node level. Set the periodicity information: **Period, Offset,** and **Unit** fields

Tip #1 “From root node” is the default option

Tip #2 “From root node” these values specified at root node level are uneditable

Setting Code Generator



Simplorer configuration

Simplorer version

Record Scenario and scenario location

Simulation mode and host IP address

Tip #1 Display the General tab to set target directory, \$(Configuration) by default

Tip #2 Display the Compiler tab to set Visual C compiler version (with specific options if required by your project)

Tip #3 Display the Build tab to set specific build options if required by your project

SCADE SUITE CODE GENERATION WORKFLOW FOR SIMULATION IN SIMPLORER

Coupling Workflow Architecture

SCADE Suite GUI Overview

Setting Code Generator

Exporting SCADE Suite Model

Exporting SCADE Suite Model: C Code Generation

Once the Simplorer configuration is set, use it to generate C code to check if no error occurs

- From the Code Generator toolbar
 - Select the Simplorer configuration
 - Click on the Generate icon



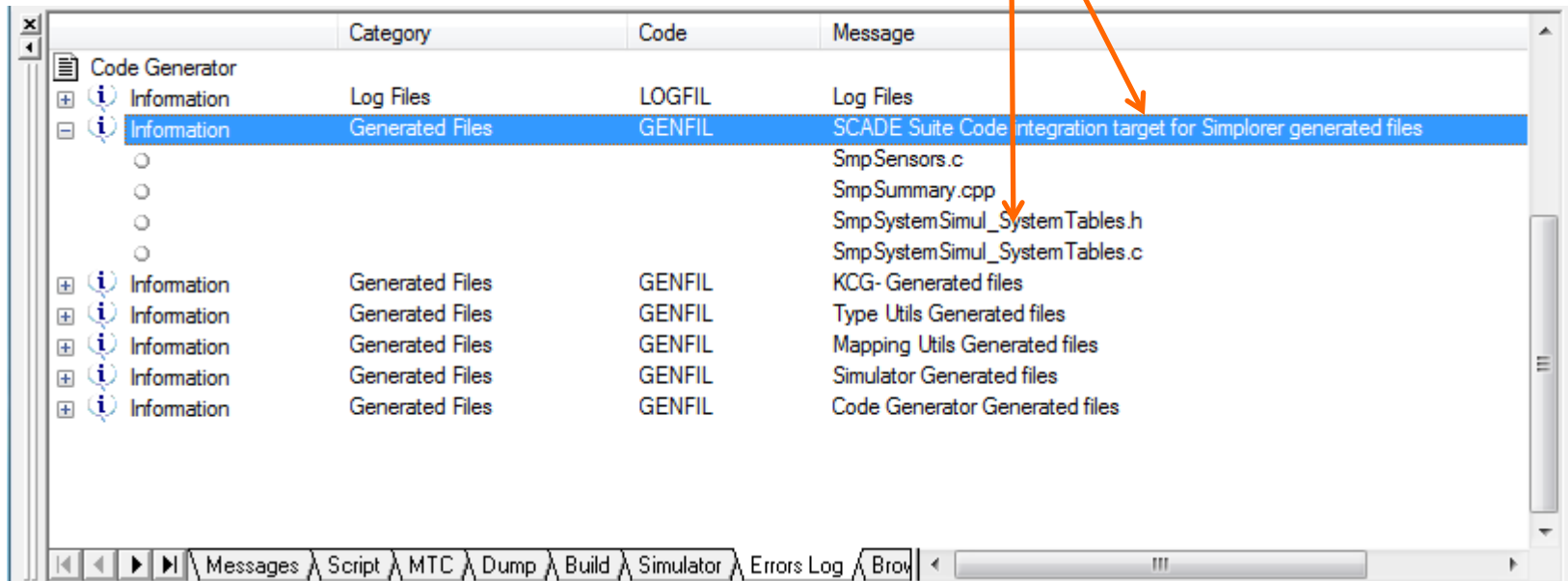
The wrapper generates the C wrapping code in the target directory, which is named \$(Configuration) by default

- As many C files as there are root operators: Smp<root>Tables.c
- One C file declaring the sensors: SmpSensors.c
- One C++ file with model level data: SmpSummary.cpp

Exporting SCADE Suite Model: C Code Generation

The wrapper list the wrapping generated files in a sub-tree of the “Errors Log” output tab

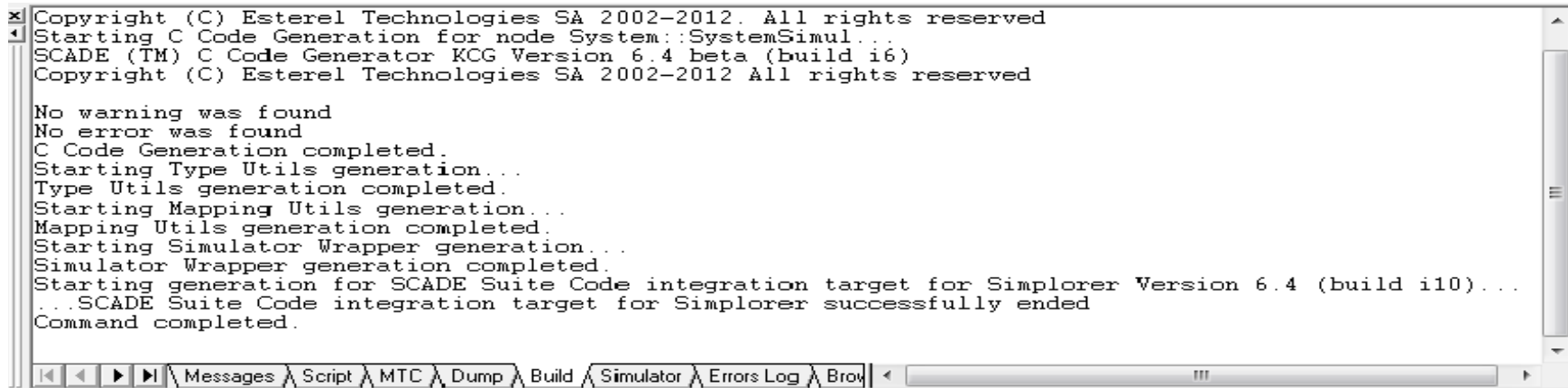
- Displays wrapper generated code in a sub-tree named
- “SCADE Suite integration for Simplorer generated files”



Exporting SCADE Suite Model: C Code Generation

The wrapper list the status of the code generation in “Build” output tab

- Reports all errors otherwise “No warning / No error was found” message
- Displays a banner with the wrapper version
- Displays when the code generation is completed, the following message, “... SCADE Suite Code integration target for Simplorer successfully ended
- Command completed”



```
Copyright (C) Esterel Technologies SA 2002-2012. All rights reserved
Starting C Code Generation for node System::SystemSimul...
SCADE (TM) C Code Generator KCG Version 6.4 beta (build i6)
Copyright (C) Esterel Technologies SA 2002-2012 All rights reserved

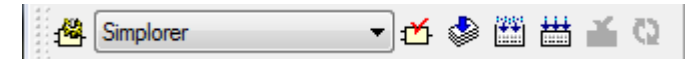
No warning was found
No error was found
C Code Generation completed.
Starting Type Utils generation...
Type Utils generation completed.
Starting Mapping Utils generation...
Mapping Utils generation completed.
Starting Simulator Wrapper generation...
Simulator Wrapper generation completed.
Starting generation for SCADE Suite Code integration target for Simplorer Version 6.4 (build i10)...
...SCADE Suite Code integration target for Simplorer successfully ended
Command completed.
```

The screenshot shows a terminal window with a tab bar at the bottom containing 'Messages', 'Script', 'MTC', 'Dump', 'Build', 'Simulator', 'Errors Log', and 'Browser'. The 'Build' tab is selected, displaying the output text. The output indicates that the C code generation process completed successfully without any warnings or errors, and the SCADE Suite Code integration target for Simplorer Version 6.4 (build i10) was successfully ended.

Exporting SCADE Suite Model: Build the DLL

Once the Simplorer configuration is set and the C code is successfully generated, use it to build the Simplorer DLL

- From the Code Generator toolbar
 - Select the Simplorer configuration
 - Click on the Build or “ReBuild All” icon



The wrapper generates the Build files in the target directory

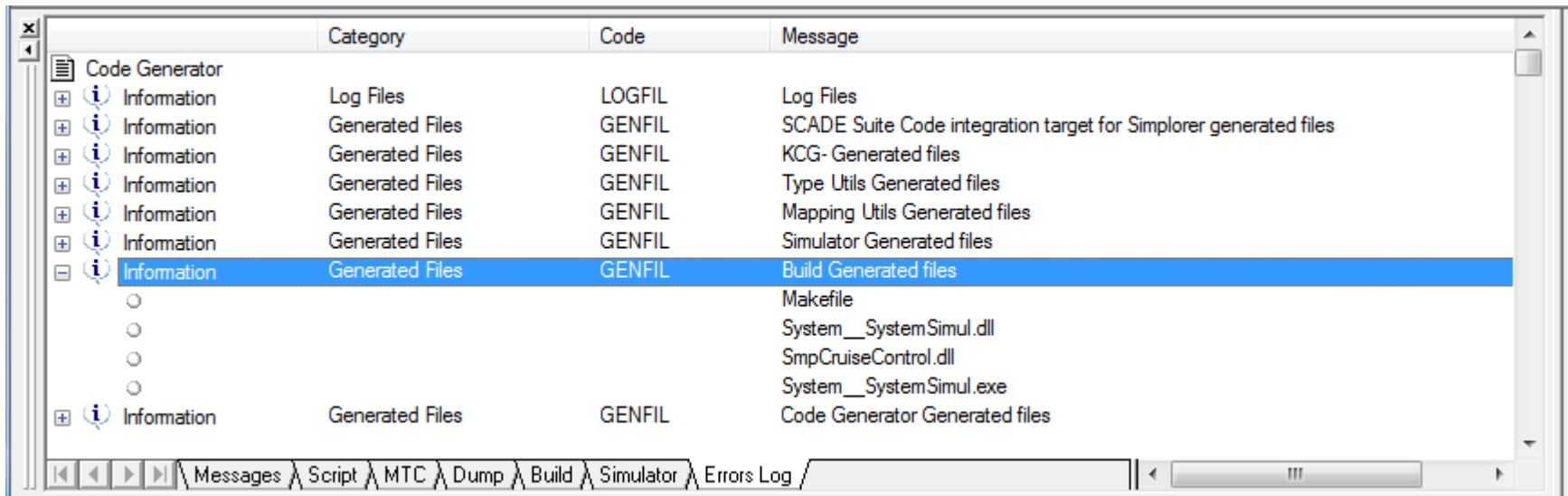
- Makefile
- Simplorer DLL : Smp<project name>.dll
- SCADE Suite Simulator DLL: <root>Simul.dll
- SCADE Suite Simulator executable: <root>Simul.exe

***Tip** The Build flow begins by previous C Code generation and finishes by Linking files*

Exporting SCADE Suite Model: Build the DLL

The wrapper list the Build generated files in a sub-tree of the “Errors Log” output tab

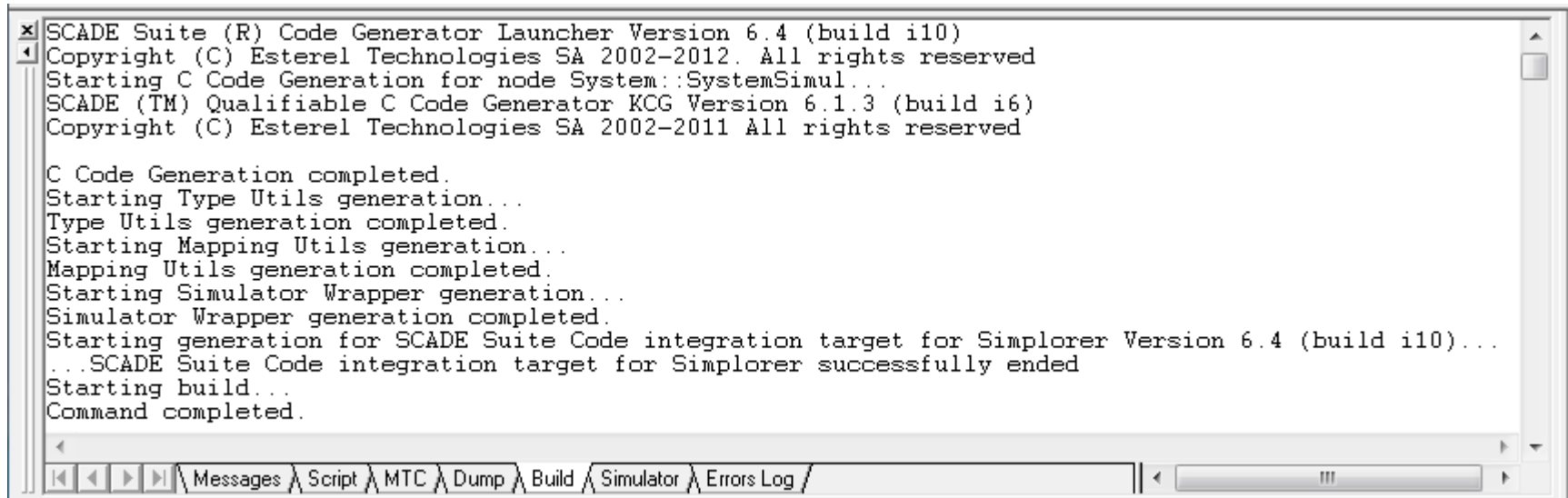
- Displays Build generated code in a sub-tree named “Build Generated files”



Exporting SCADE Suite Model: Build the DLL

The wrapper list the status of the Build generation in “Build” output tab

- Reports all errors
- Displays a banner with its identification and version
- Displays when the build generation is completed, the following message, Starting build
- Command completed.

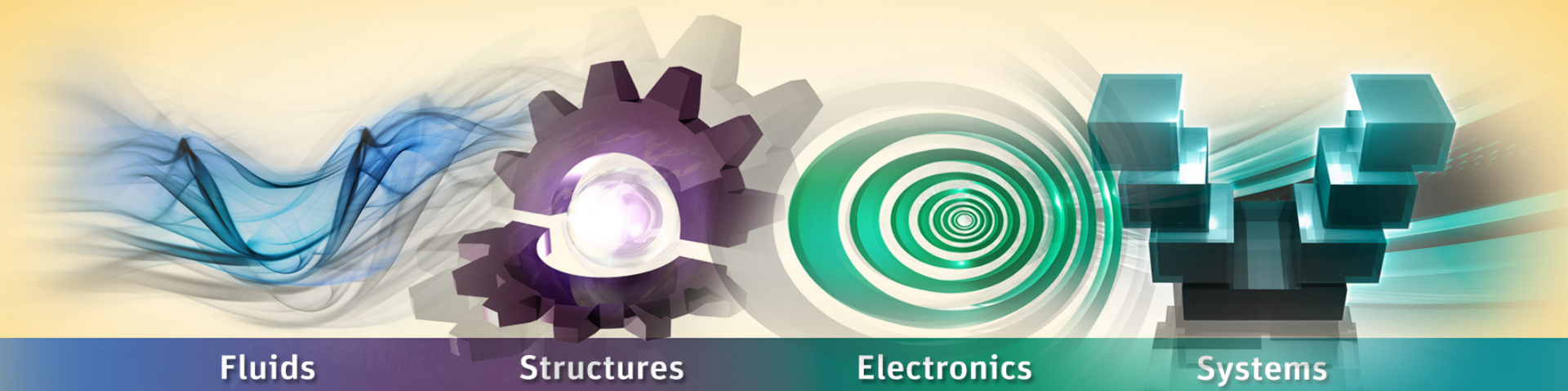


```
SCADE Suite (R) Code Generator Launcher Version 6.4 (build i10)
Copyright (C) Esterel Technologies SA 2002-2012. All rights reserved
Starting C Code Generation for node System::SystemSimul...
SCADE (TM) Qualifiable C Code Generator KCG Version 6.1.3 (build i6)
Copyright (C) Esterel Technologies SA 2002-2011 All rights reserved

C Code Generation completed.
Starting Type Utils generation...
Type Utils generation completed.
Starting Mapping Utils generation...
Mapping Utils generation completed.
Starting Simulator Wrapper generation...
Simulator Wrapper generation completed.
Starting generation for SCADE Suite Code integration target for Simplorer Version 6.4 (build i10)...
...SCADE Suite Code integration target for Simplorer successfully ended
Starting build...
Command completed.
```

Messages Script MTC Dump Build Simulator Errors Log

SCADE TRAINING



**Import of the SCADE Suite-
generated DLL in the Simplorer
Environment**

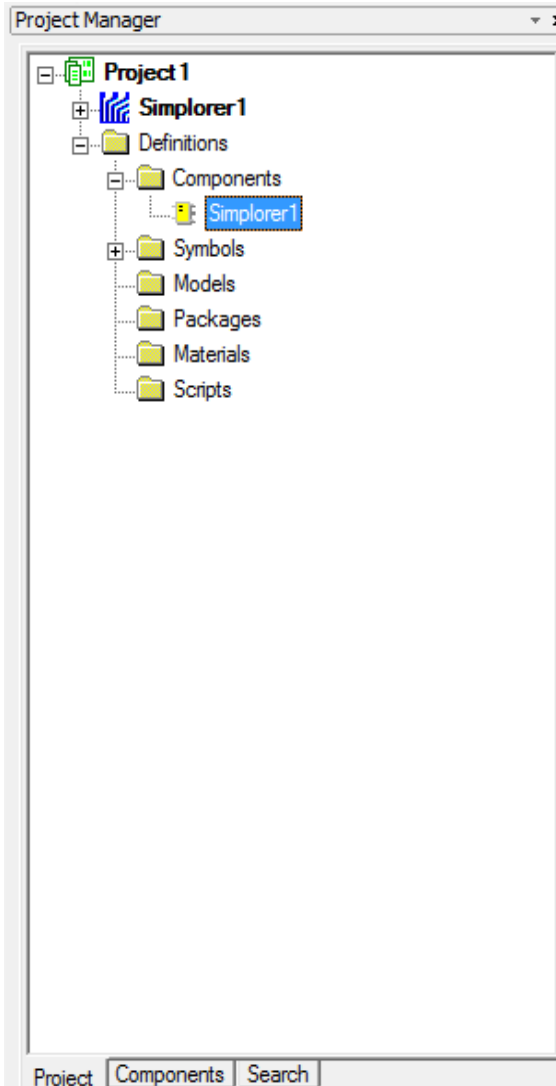
Import of DLL in the Simplorer Environment

Once the SCADE Suite-generated DLL is generated, you can import it in the Simplorer Environment as a Simulation model

When the Simulation model is imported, you can use SCADE Suite operators as any other Simplorer components

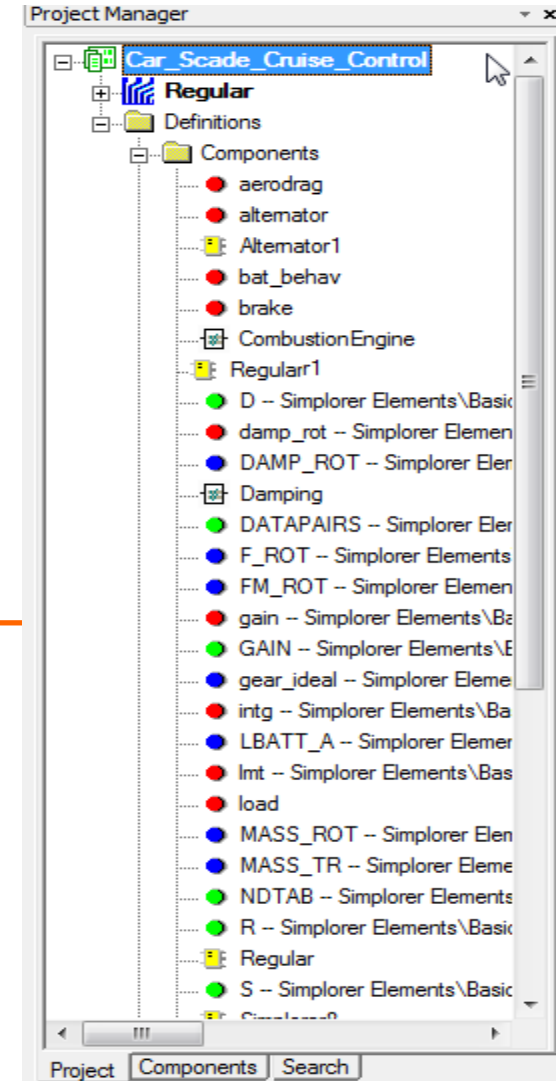
You can simulate SCADE Suite operators in a Simplorer project as any other Simplorer components

Import of DLL in the Simplorer Environment



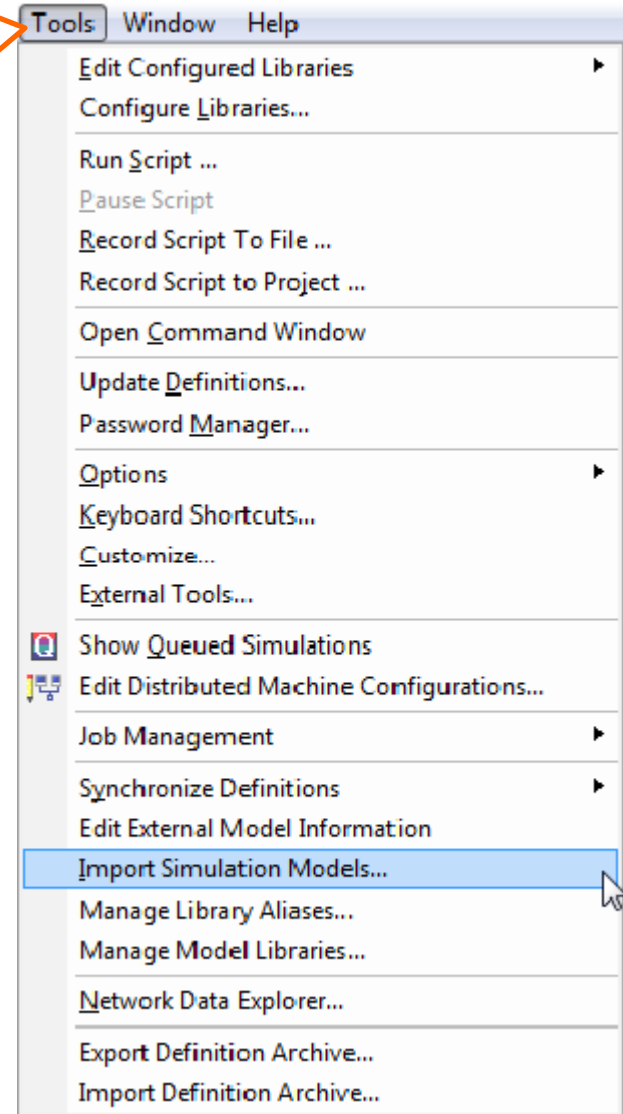
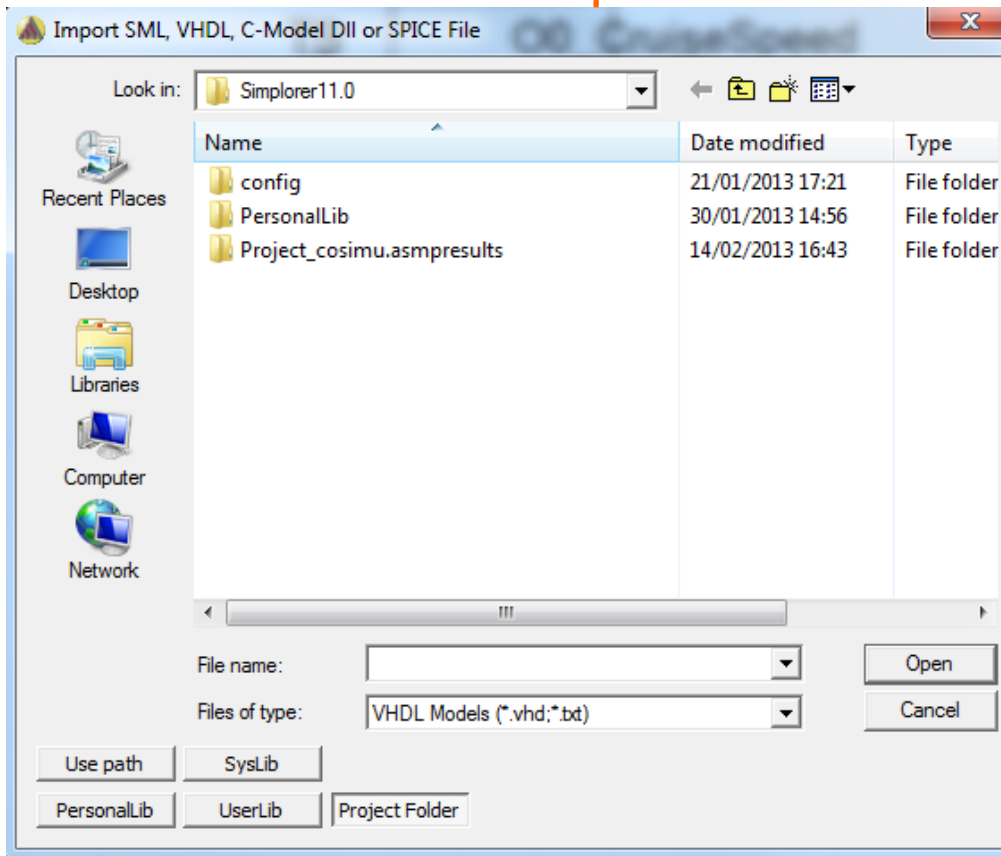
- From Simplorer
- Create a new project
 - Select the target component

- Or
- Open an existing project
 - Select the target component

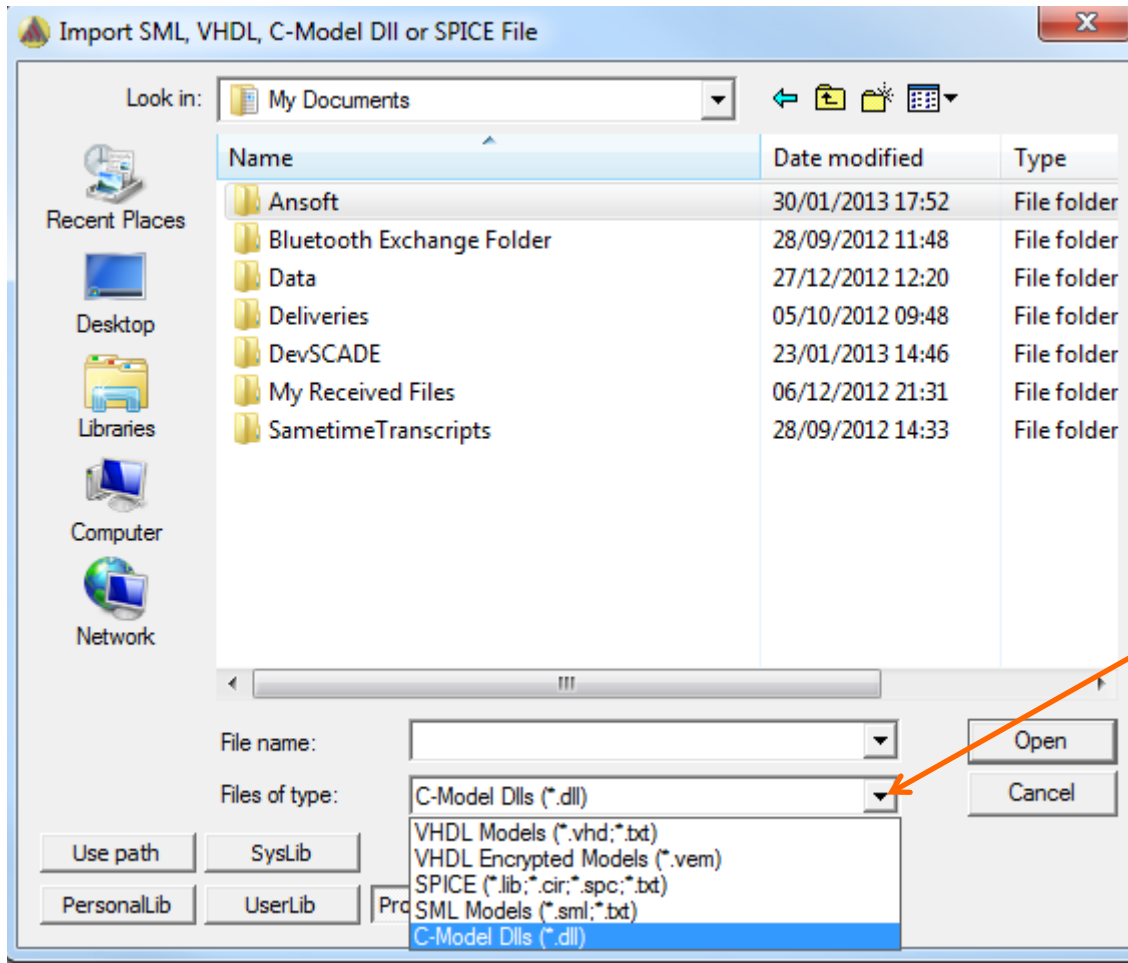


Import of DLL in the Simplorer

Select **Tools – Import Simulation models...** in the Simplorer menu to open the “Import SML, VHDL, C-Model Dll or SPICE file” file selection dialog

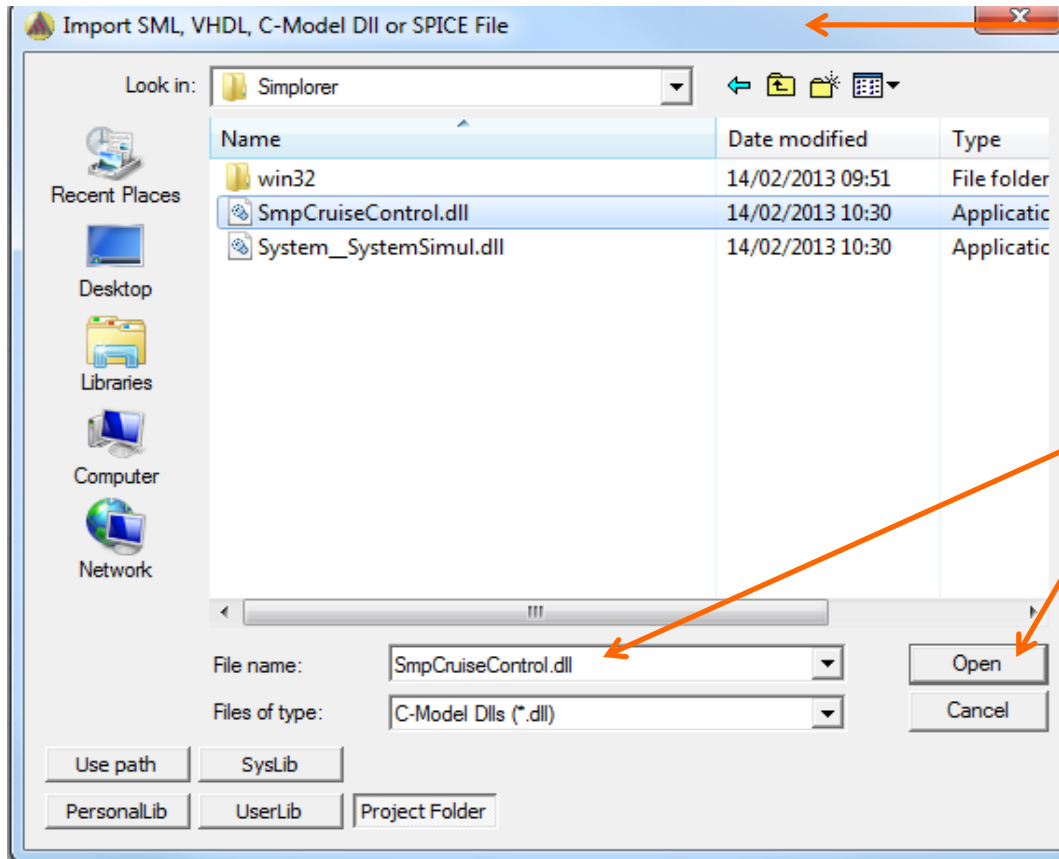


Import of DLL in the Simplorer Environment



Select "C-Model Dlls (*.dll)" from the drop-down list

Import of DLL in the Simplorer Environment



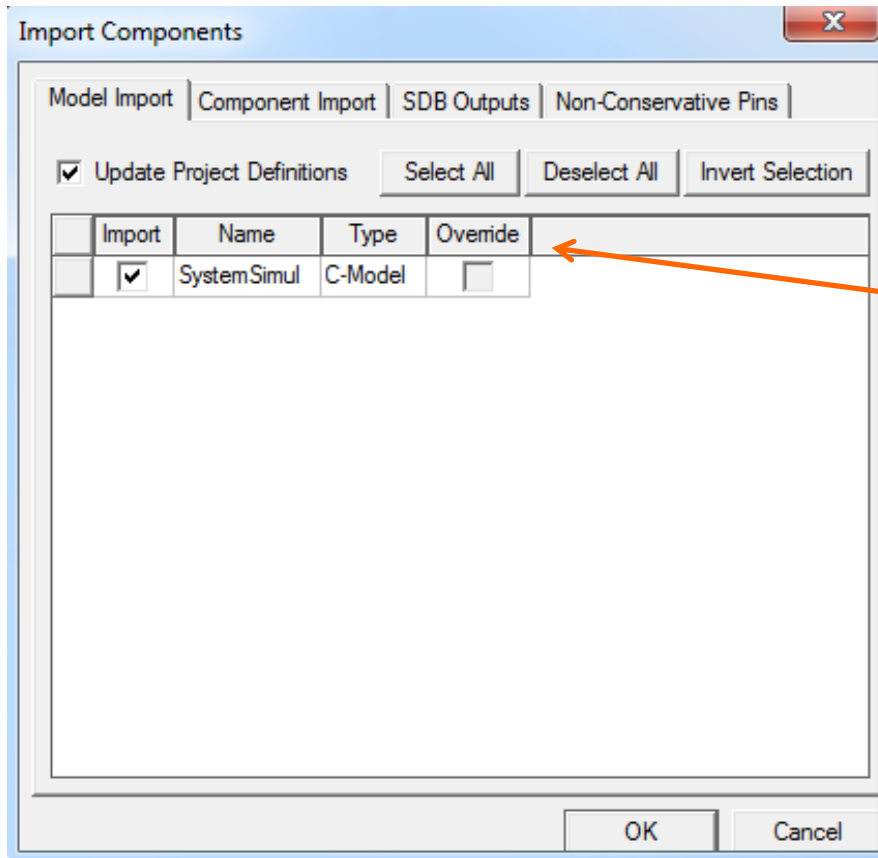
Go to the DLL target directory using the Up-directory icon

Select the Simplorer DLL - Smp<project name>.dll - with the Open button

Tip When importing C-Models, if the selected dll file already exists in libraries, a dialog box appears asking if you want to replace the existing dll with the one you are importing

Import of DLL in the Simplorer Environment

Once the DLL is loaded, the “Import Components” dialog box is displayed with four tabs

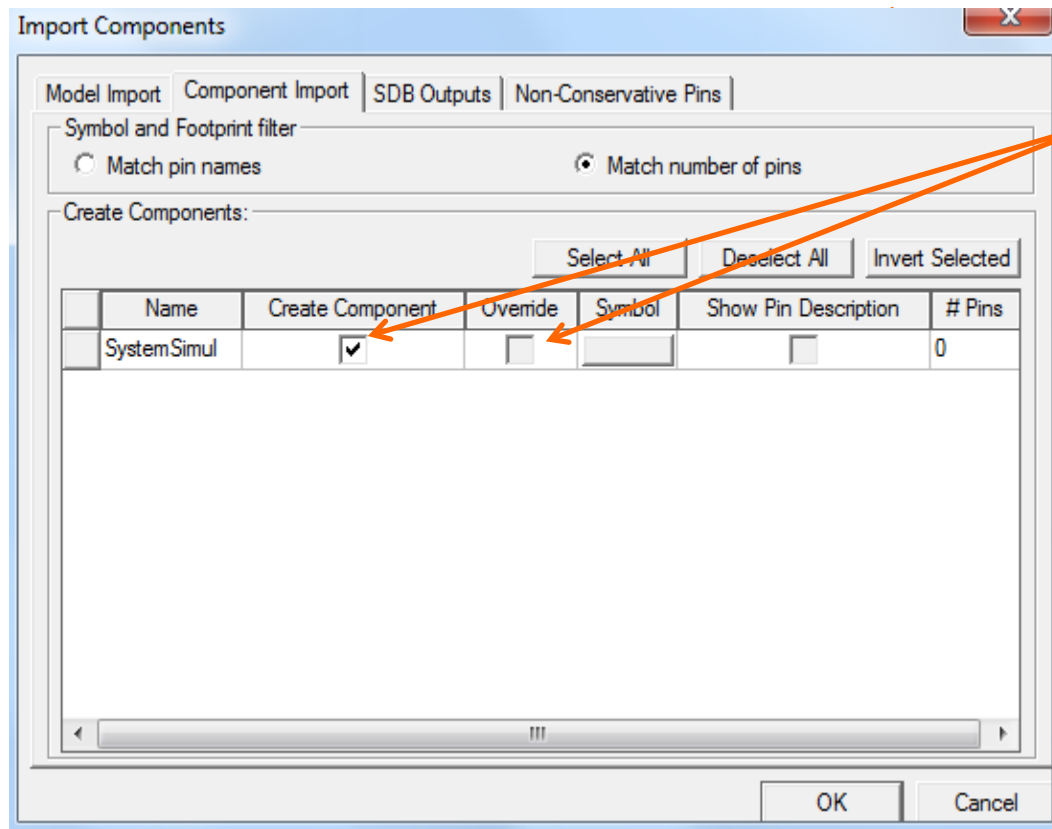


Model Import tab:

A table lists the names of models that will be imported. If a model name already exists, the Override checkbox (checked by default) allows you to override the current model definition

Import of DLL in the Simplorer Environment

The “Import Components” dialog box is displayed



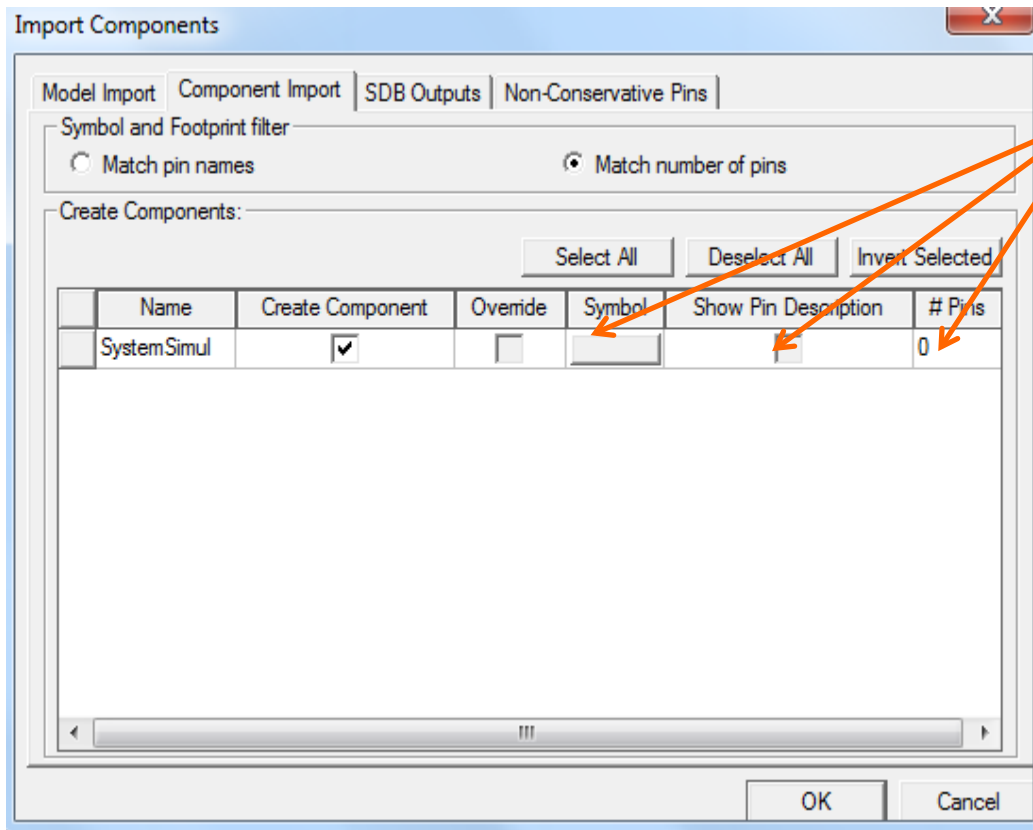
“Component Import” tab:

A table

- lists the models available for import by name
- provides a checkbox for selecting whether or not to create a component from the model (generally checked for the first import else the Override checkbox enables the imported component to override an existing component of the same name)

Import of DLL in the Simplorer Environment

The “Import Components” dialog box is displayed



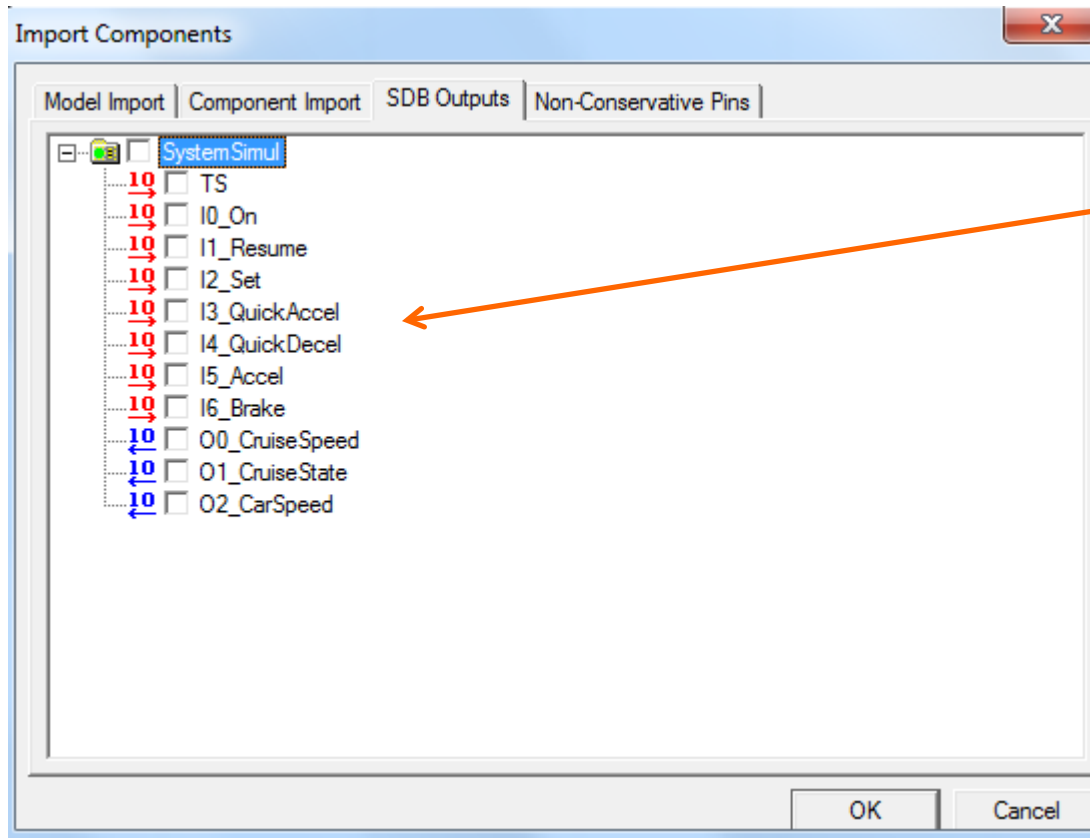
“Component Import” tab:

This table provides

- a button for choosing a symbol for the new component from the symbol library
- a checkbox that allows you to choose whether or not to show pin descriptions
- a pin count for the component (reflects only the number of conservative pins)

Import of DLL in the Simplorer Environment

The “Import Components” dialog box is displayed

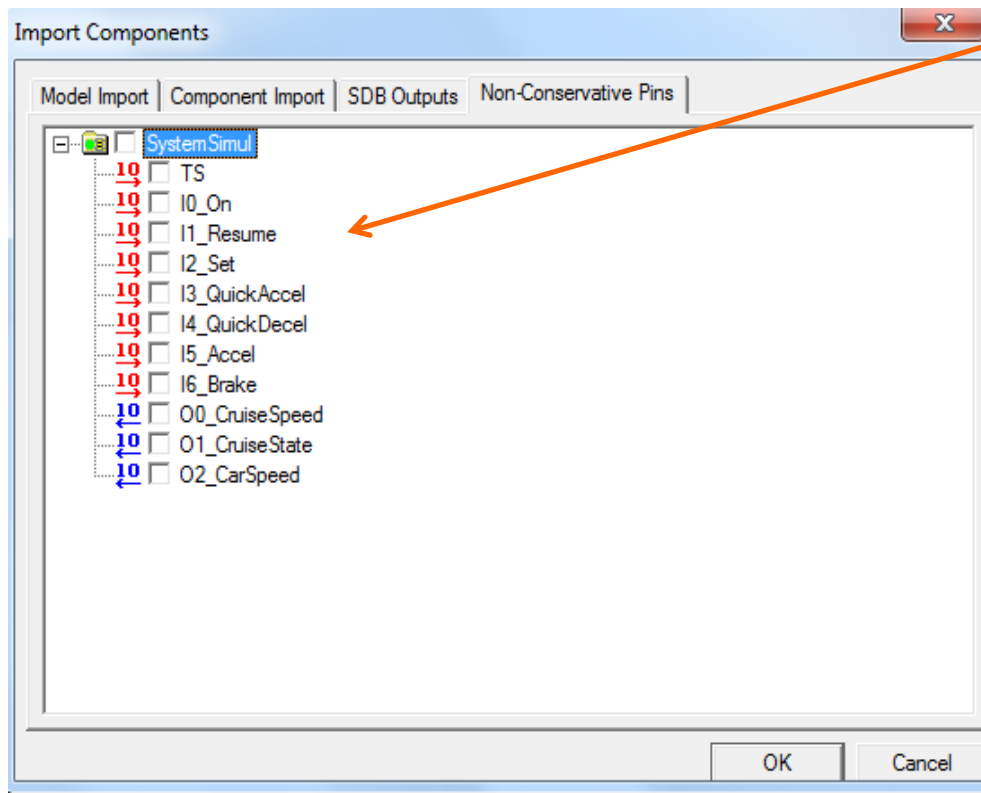


“SDB Outputs” tab:

Select the quantities and signals you want to set as default outputs for each created component

Import of DLL in the Simplorer Environment

The “Import Components” dialog box is displayed



“Non-Conservative Pins” tab:
Select the non-conservative quantities and signals for which you want pins included on the new component symbol.
The number of component pins on the “Component Import tab” reflects your choices

Tip#2 Conservative pins belong to components of physical domains (electrical, ...etc). They have a nature type and no direction attribute

Tip#1 Non-conservative pins represent many different data types. They can have a direction attributes: IN, OUT, or IN/OUT

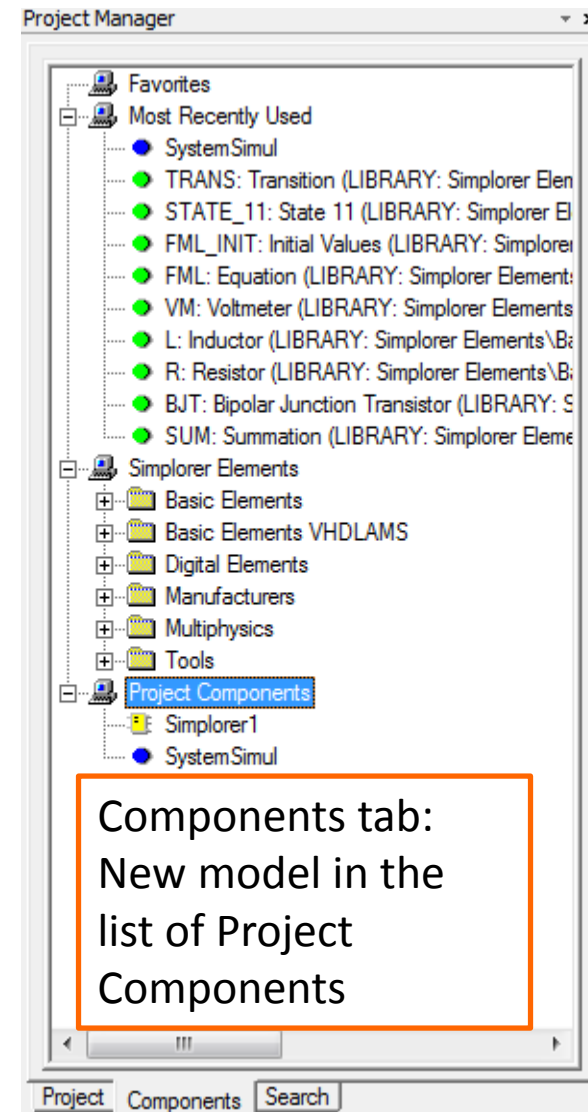
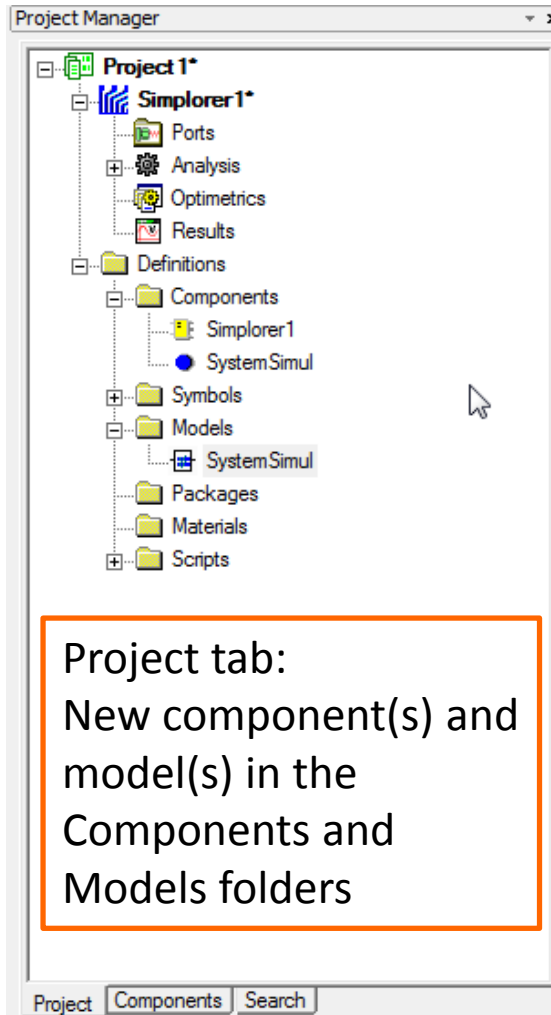
Import of DLL in the Simplorer Environment

Click either OK to validate or simply close the settings window to acknowledge the changes

The new component(s) and model(s) are added to the Components and Models folders within the current Definitions folder on the Project Manager Project tab

The new model is also added to the list of Project Components on the Project Manager Components tab where it can be placed onto a schematic for simulation

Import of DLL in the Simplorer Environment



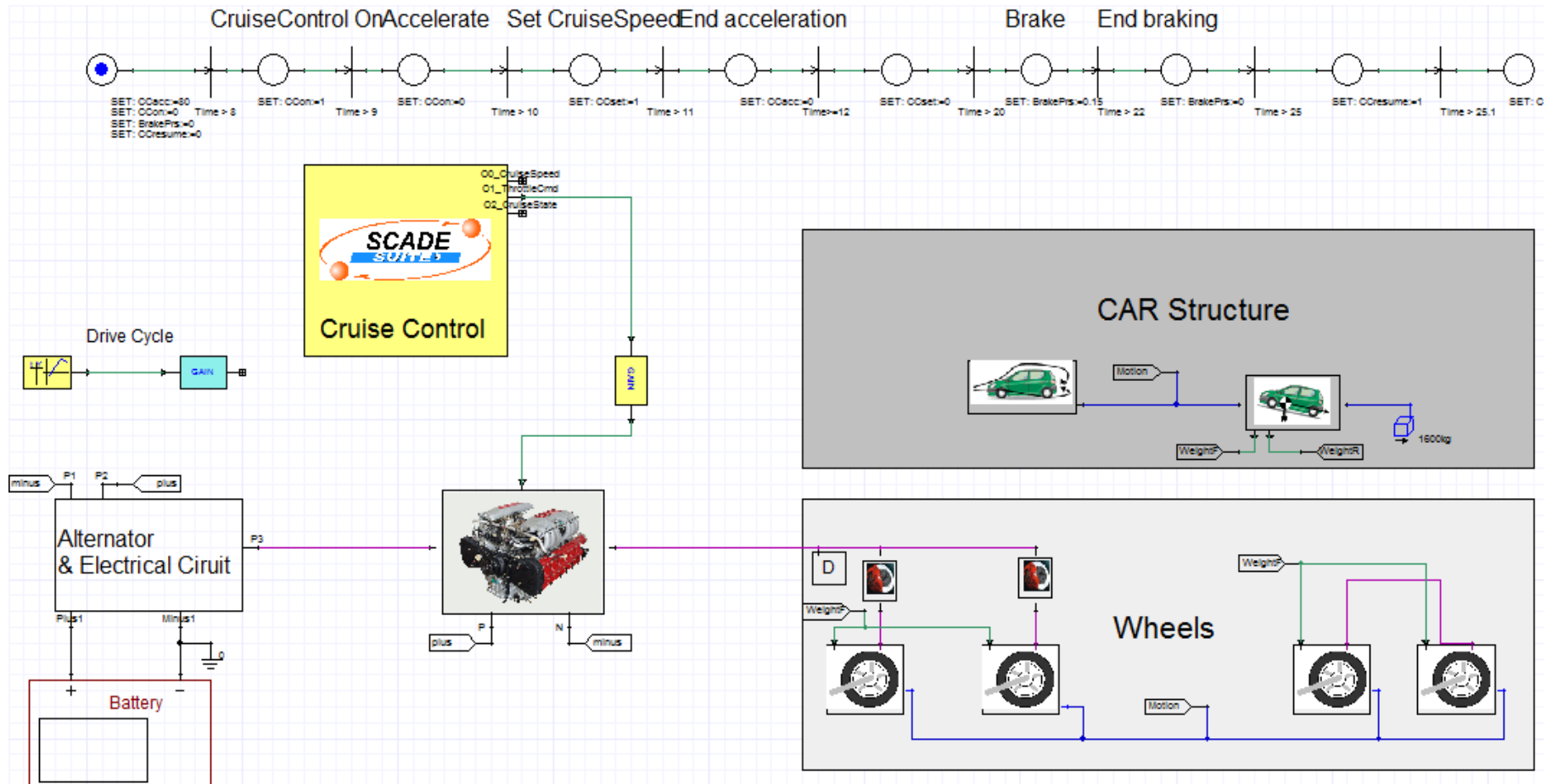
Import of DLL in the Simplorer Environment

Once the DLLs are imported, you can use the new SCADE Suite operators as any other Simplorer components in the schematic

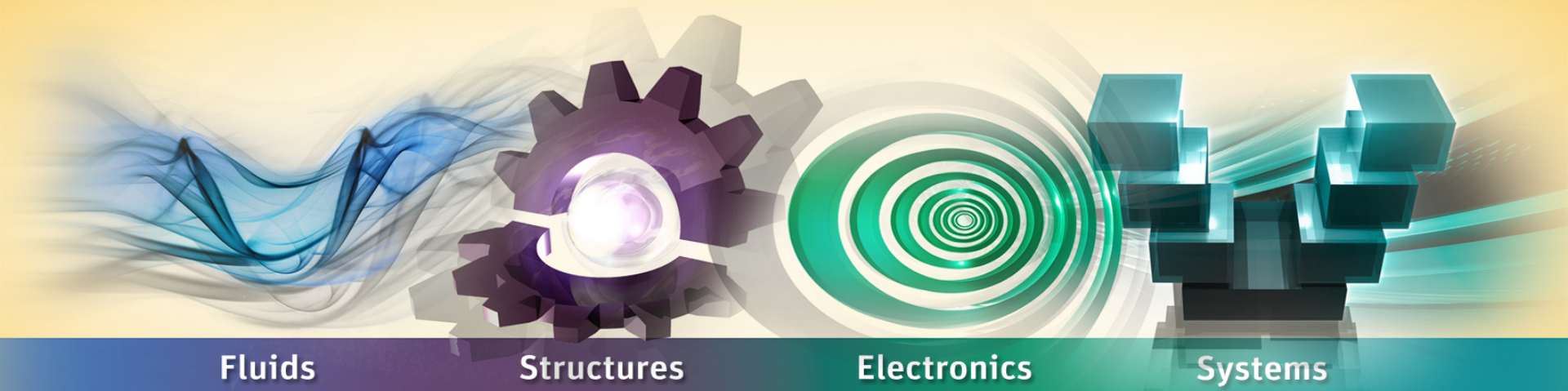
You start Black-Box or White-Box co-simulation in Simplorer depending on selected Simplorer settings chosen while building the DLL.

Import of DLL in the Simplorer Environment

Example of schematic:



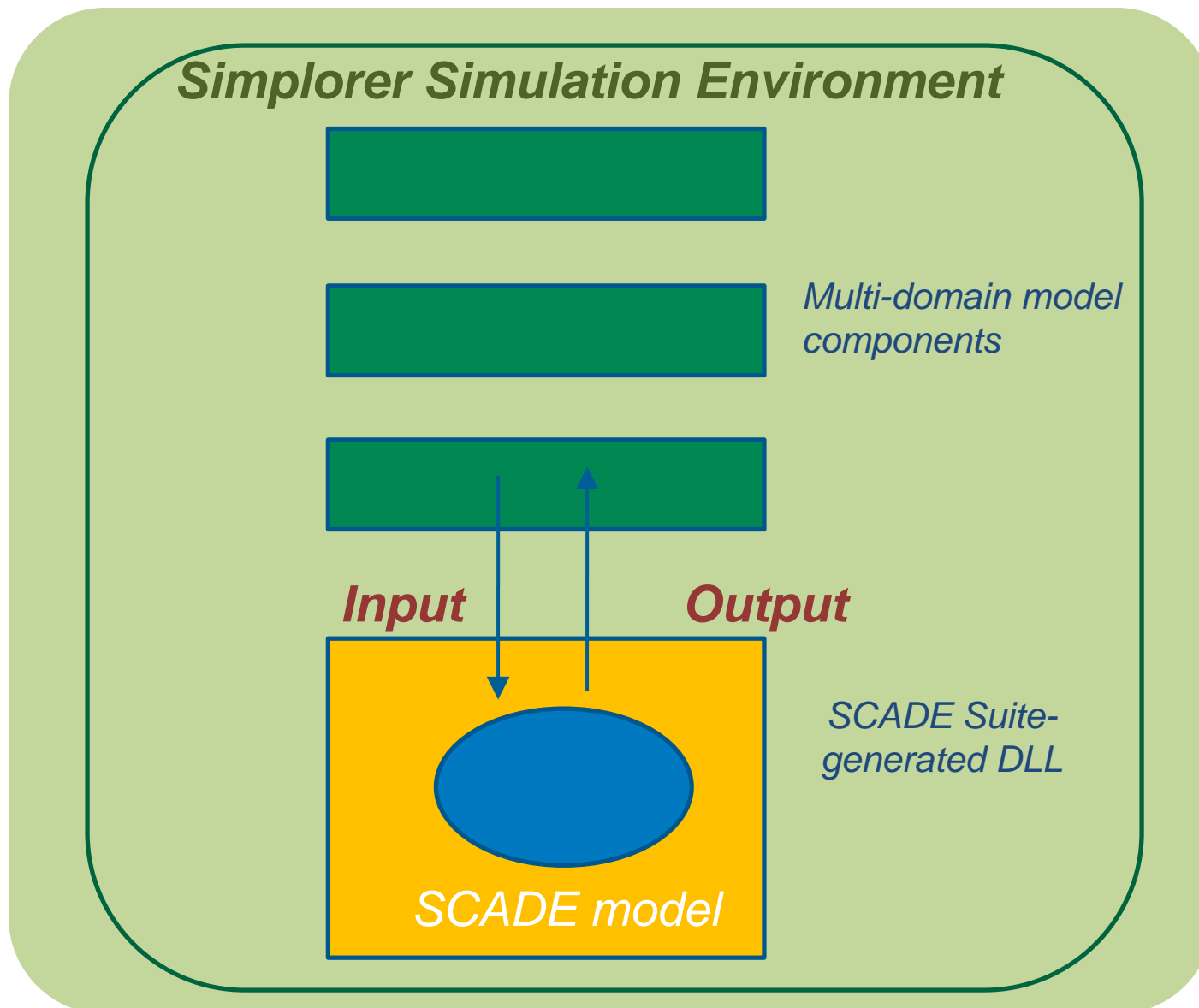
SCADE TRAINING



Black-Box co-simulation Example

KCG C code from the SCADE Suite model is directly embedded in the generated DLL for Simplorer.

SCADE Suite cyclic function of the root operator is called periodically at the user-specified period



The aim of this exercise is

- To generate the expected Simplorer DLL in SCADE Suite for the CruiseControl model to activate the Black-Box mode (exercise 1-1)
- To import this DLL and simulate it in Black-Box mode in the Simplorer Environment for Car_Scade_Cruise_Control schematic (exercise 1-2)

To work on this exercise, use a virtual drive named “N”

- Launch a Dos cmd
- Specifies the virtual drive “N:” to which you want to assign the path of the User Exercises folder:
 - That is to obtain the following virtual path: N:/Exercises/...
 - subst N: <Exercises path>
 - Example:
 - Physical address: C:/Users/Exercises
 - subst N: "C:/Users"
 - => Virtual path: N:/Exercises

Load the SCADE Suite project named CruiseControl.etp located in the “Prerequisites/Exercise 1/CruiseControl” folder

Set the Simplorer settings in a custom configuration

- Select Project – Configurations in SCADE Suite menu to open the Configurations window
 - Add a new configuration from “KCG” one
 - Rename it, “SimplorerBB”

Exercise 1-1



- Select this “SimplorerBB” configuration
- Click on the Settings icon from the Code Generator toolbar or select Project - Code Generator – Settings in SCADE Suite menu to open the Settings window
 - In the “General” tab
 - Select the KCG 6.1.3 Code Generator version
 - Select the “CruiseControl::CruiseControl” root node
 - In the “Code Integration” tab
 - Select in the list of targets and check “Simplorer” target
 - Set the periodicity from root node

Exercise 1-1



- In the “Compiler” tab
 - Select the Visual C version (VC6.0)
- In the “Simplorer” tab
 - Select the version (11. 0)
 - Uncheck if need be “Record scenario”
 - Uncheck if need be “Enable white box”
- Validate these choices clicking on the OK button

Generate the C code

- Click on the Generate icon from the Code Generator toolbar or select Project - Code Generator – Generate Node CruiseControl in SCADE Suite menu to generate C code
- Check if no error occurs and the Code generation is completed in the “Build” output tab

Build the DLL

- Click on the Build icon from the Code Generator toolbar or select Project - Code Generator – Build Node CruiseControl in SCADE Suite menu to generate the DLL
- Check if no error occurs and the Build generation is completed in the “Build” output tab

Load the Simplorer project named
Car_Scade_Cruise_Control_to_complete.asmp located in the
“Prerequisites/Exercise 1” folder

Import the generated DLL as a Simulation model

- Select Tools – Import Simulation models... in the Simplorer menu
 - From the “Import SML, VHDL, C-Model Dll or SPICE file” file selection dialog, load the DLL from the target directory

Exercise 1-2



- From the “Import Components” dialog box
 - Model Import tab: verify if “CruiseControl” is available and check “Override”

Check that the CruiseControl component and model are available into the Components and Models folders

Drag and drop this component to complete the schematic

Edit CruiseControl Properties Box

- From the Quantities tab
 - Uncheck “Show Pin” from all inputs
 - Modify the following default values:
 - I0_On: Ccon
 - I2_Resume: CCresume
 - I3_Set: CCset
 - I6_Accel: CCacc
 - I7_Brake: BrakePrs*100
 - I8_Speed: VehicleMass.V*3.6

Drag and drop the GAIN Continuous Block to link it to CruiseControl (GAIN input) and CombustionEngine (GAIN output) components in the schematic

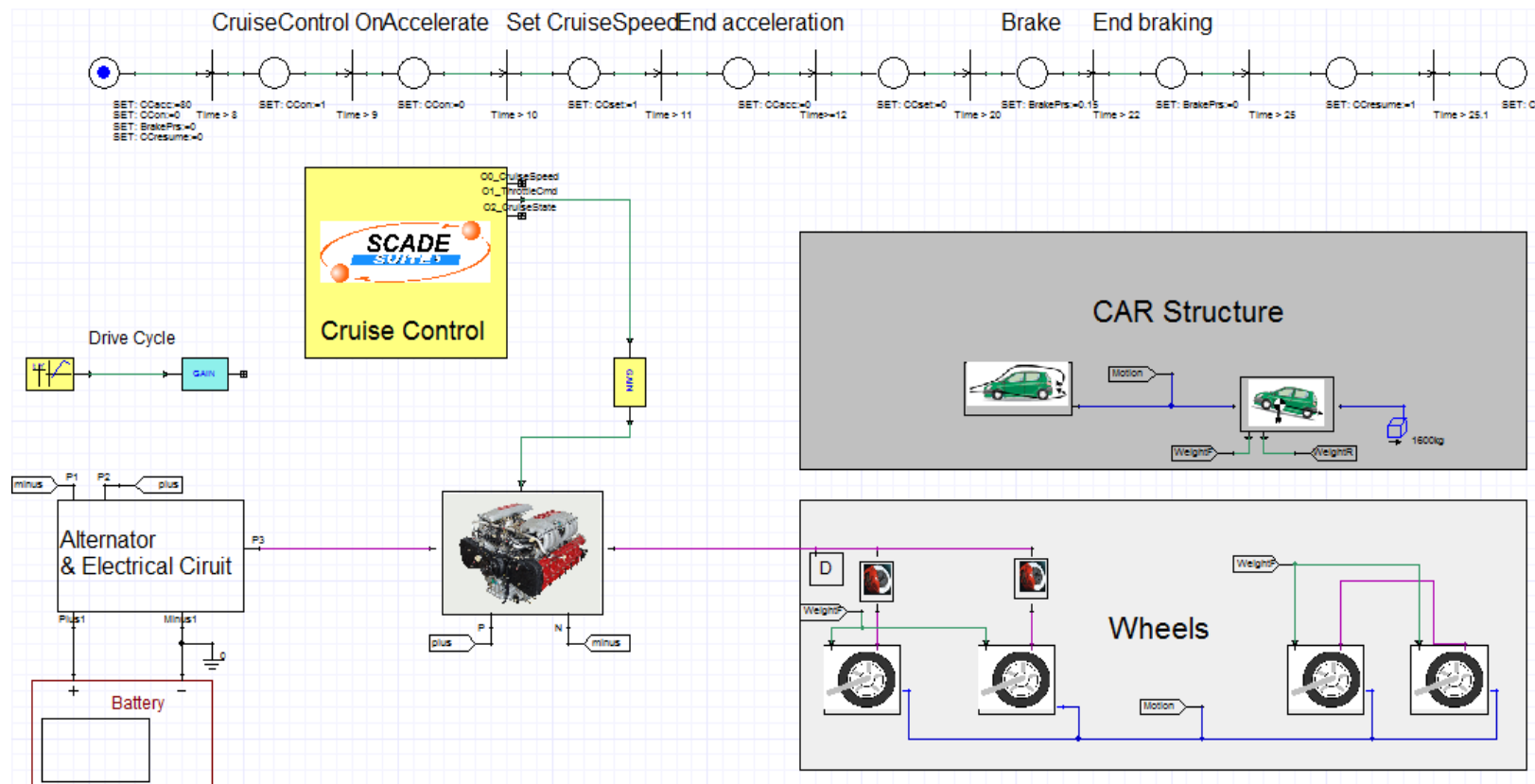
Edit Gain Properties Box

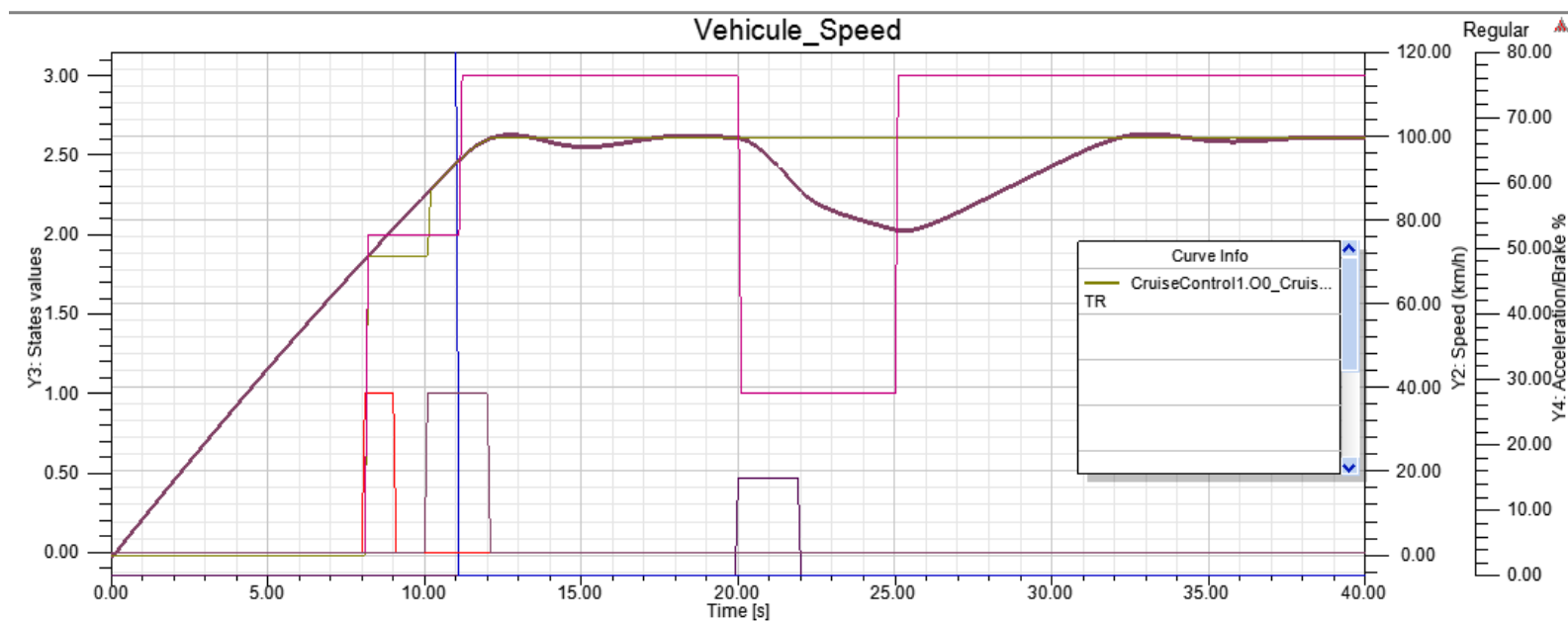
- From the Quantities tab
 - Uncheck “Show Pin” from the input
 - Modify the following default value:
 - KP: 1/10/100

Simulate it

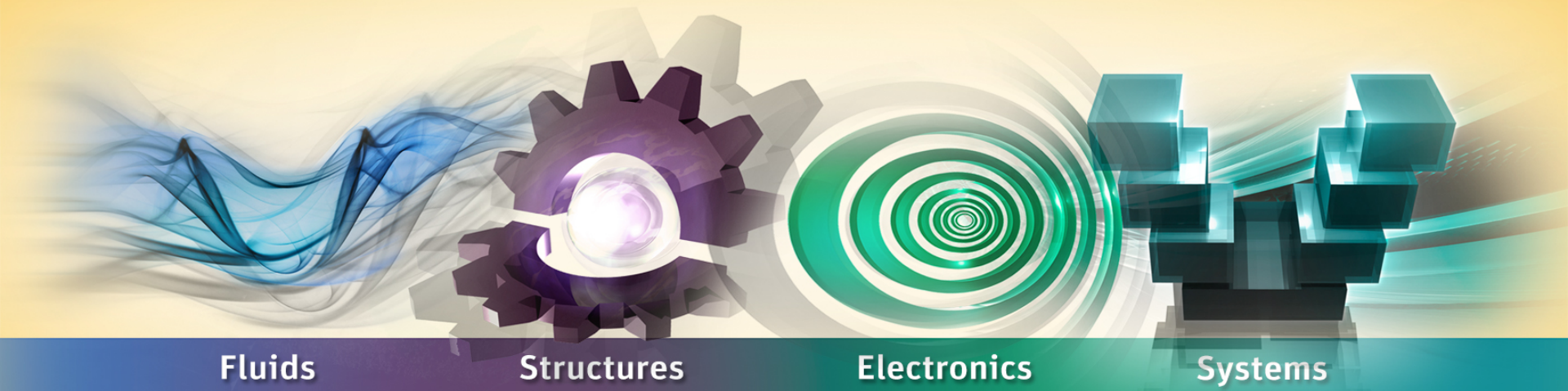
- Expand the Regular/Results tree in the Project View
- Launch the simulation (Simulate Setup icon on the menu)
- Verify that only the Simplorer simulation runs
- When the simulation finishes:
 - Select Vehicule_Speed report and click on “Open Report”
 - Check results on the graph

SOLUTION





SCADE TRAINING



White-Box co-simulation Example

SCADE Suite model is executed in SCADE Suite Simulator

Simplorer forwards the inputs to SCADE Suite Simulator through a network connection

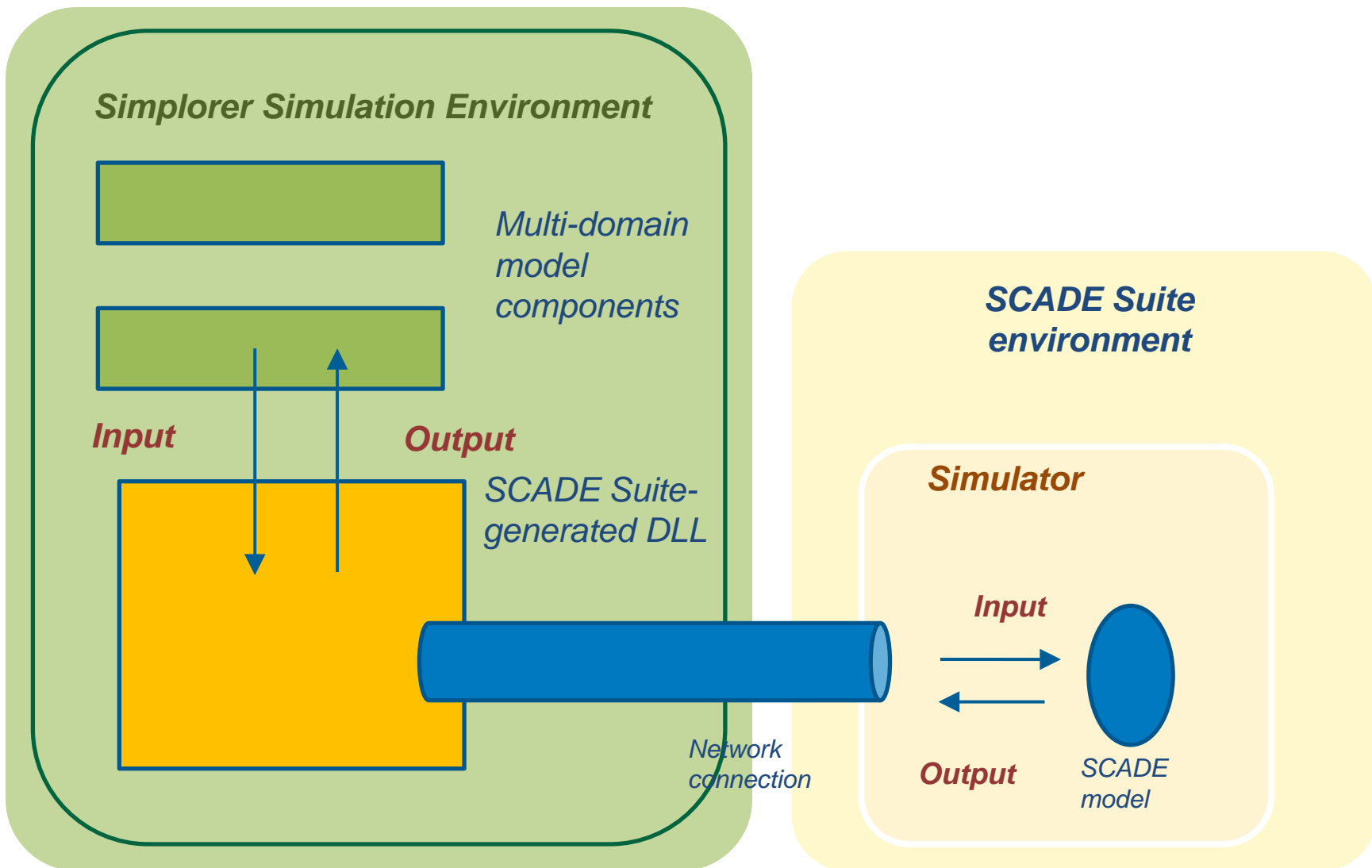
This mode enables model debugging with the full power of SCADE Suite simulation where you can observe model reactions graphically or set breakpoints and stop conditions.

SCADE Suite cyclic function of the root operator is called periodically at the user-specified period

The white-box co-simulation is only possible for one instance of an operator in a given DLL

Multi-root operators are not allowed with this option

For white-box simulation, specify the IP address of the host in the Co-simulation host field where you want to execute SCADE Suite. Co-simulation with Simplorer can be launched from another host able to connect to the specified IP. Do not change the default value of the co-simulation host if you run co-simulation with SCADE Suite and Simplorer on the same local host.



The aim of this exercise is

- To generate the expected Simplorer DLL in SCADE Suite for the CruiseControl model to activate the White-Box mode (exercise 2-1)
- To import this DLL and simulate it in White-Box mode in the Simplorer Environment for Car_Scade_Cruise_Control schematic (exercise 2-2)

To work on this exercise, use a virtual drive named “N”

- Launch a Dos cmd
- Specifies the virtual drive “N:” to which you want to assign the path of the User Exercises folder:
 - That is to obtain the following virtual path: N:/Exercises/...
 - subst N: <Exercises path>
 - Example:
 - Physical address: C:/Users/Exercises
 - subst N: "C:/Users"
 - => Virtual path: N:/Exercises

Load the SCADE Suite project named CruiseControl.etp located in the “Prerequisites/Exercise 2/CruiseControl” folder

Set the Simplorer settings in a custom configuration

- Select Project – Configurations in SCADE Suite menu to open the Configurations window
 - Add a new configuration from “SimplorerBB” one
 - Rename it, “SimplorerWB”

Exercise 2-1



- Select this “SimplorerWB” configuration
- Click on the Settings icon from the Code Generator toolbar or select Project - Code Generator – Settings in SCADE Suite menu to open the Settings window
 - In the “General” tab
 - Verify that the KCG 6.1.3 Code Generator version is selected
 - Verify that the “CruiseControl::CruiseControl” root node is selected
 - In the “Code Integration” tab
 - Verify that “Simplorer” target is selected and checked
 - Set the periodicity from root node

Exercise 2-1

- In the “Compiler” tab
 - Verify that the Visual C compiler version is selected (VC6.0)
- In the “Simplorer” tab
 - Select the version (11.0)
 - Check “Record scenario”
 - Check “Enable white box”
- Validate these choices clicking on the OK button

Generate the C code

- Click on the Generate icon from the Code Generator toolbar or select Project - Code Generator – Generate Node CruiseControl in SCADE Suite menu to generate C code
- Check if no error occurs and the Code generation is completed in the “Build” output tab

Build the DLL

- Click on the Build icon from the Code Generator toolbar or select Project - Code Generator – Build Node CruiseControl in SCADE Suite menu to generate the DLL
- Check if no error occurs and the Build generation is completed in the “Build” output tab

Load the Simplorer project named Car_Scade_Cruise_Control.asmp located in the “Prerequisites/Exercise 2” folder

Import the generated DLL as a Simulation model

- Select Tools – Import Simulation models... in the Simplorer menu
 - From the “Import SML, VHDL, C-Model Dll or SPICE file” file selection dialog, load the DLL from the target directory

- From the “Import Components” dialog box
 - Model Import tab: verify if “CruiseControl” is available, check “Override” item

Simulate it

- Expand the Regular/Results tree in the Simplorer Project View
- Launch the Simplorer simulation (Simulate Setup icon on the menu)
- Check that the SCADE Suite Simulator raises
- Insert a breakpoint at the state
CruiseControl::CruiseControl/SM1:Enabled:SM2:Active:SM3:On: (SCADE Suite simulator)

- Run the SCADE Suite simulation (Go icon on the SCADE Suite Simulator toolbar)
- Verify that these two simulations stop when the breakpoint is reached
- Remove this breakpoint
- Re-launch the SCADE simulation (Go icon)
- Verify that the Simplorer simulation continues also
- Check results in Simplorer when both simulations finish
 - Select Vehicule_Speed report and click on “Open Report”
 - Check results on the graph

Exercise 2-2



- Check that the following scenario file (.in format) is saved in the \$(Configuration) target directory
 - scenario-\$(DATE)- \$(TIME)-\$(USE).in

SOLUTION

