Twin Builder® Modelica Tutorial

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

Release 2025 R2
July 2025

Table of Contents

Table of Contents **Contents-1**

1 - Introduction to Modelica in Twin Builder **1-1**

General Description of Contents 1-2

Twin Builder Documentation 1-2

2 - Working with Modelica in Twin Builder **2-1**

Overview of the Twin Builder Interface 2-1

Overview of the Twin Builder Modelica Environment 2-3

Starting Twin Builder and Creating a New Project 2-4

The Twin Builder Component 2-7

Related Topics 2-9

Creating a Modelica Component in Twin Builder 2-9

Importing a Modelica Model File 2-9

Using the Diagram Editor to Create a Modelica Model 2-12

Using a Modelica Component in a Twin Builder Design 2-17

Analyzing a Design and Plotting Results 2-17

Workflow for Creating, Compiling, and Simulating a Modelica Model in Twin Builder 2-19

Adding a New Project and Modelica Model in Twin Builder 2-20

Placing Modelica Components for a New Model 2-21

Connecting Modelica PID Controller Components 2-22

Adding an Initial Equation and Variable Declaration to the PID Controller 2-23

Initial Equation 2-23

Variable Declaration 2-25

Setting Up Model Properties for the PID Controller 2-26

Compiling and Updating the PID Controller Project 2-28

Adding the PID Controller Component to a Twin Builder Schematic 2-31

Defining Outputs and Creating Reports for the PID Controller 2-33

Setting Up and Running a Transient Analysis for the PID Controller 2-34
3 - Case Studies **3-1**
Mechanics Example: Heat Losses 3-1
Heat Losses Example Workflow 3-2
Adding a New Project and Modelica Model in Twin Builder 3-2
Placing Modelica Components for the Heat Losses Model 3-3
Connecting Modelica Heat Losses Components 3-4
Setting Up Model Properties for the Heat Losses Example 3-6
Compiling and Updating the Heat Losses Project 3-10
Adding the Heat Losses Component to a Twin Builder Schematic 3-13
Defining Outputs and Creating Reports for the Heat Losses Design 3-15
Setting Up and Running a Transient Analysis for the Heat Losses Design 3-16
Fluids Example: Controlled Tanks 3-19
Adding a New Project and Modelica Model in Twin Builder 3-20
Placing Modelica Components for the Controlled Tanks Model 3-21
Connecting Modelica Controlled Tanks Components 3-22
Declaring a Fluid Medium Variable for the Controlled Tanks Model 3-25
Setting Up Model Properties for the Controlled Tanks Example 3-27
Compiling and Updating the Controlled Tanks Project 3-29
Adding the Controlled Tanks Component to a Twin Builder Schematic 3-33
Defining Outputs and Creating Reports for the Controlled Tanks Design 3-35
Setting Up and Running a Transient Analysis for the Controlled Tanks Design 3-36
Thermal Example: Parallel Pump Drop Out 3-37
Adding a New Project and Modelica Model in Twin Builder 3-39
Placing Modelica Components for the Parallel Pump Drop Out Model 3-39
Connecting Modelica Parallel Pump Drop Out Components 3-41
Setting Up Model Properties for the Parallel Pump Drop Out Example 3-42
Compiling and Updating the Parallel Pump Drop Out Project 3-45

Adding the Parallel Pump Drop Out Component to a Twin Builder Schematic 3-48
Defining Outputs and Creating Reports for the Parallel Pump Drop Out Design 3-50
Setting Up and Running a Transient Analysis for the Parallel Pump Drop Out Design 3-51
Magnetics Example: SMPM Inverter MultiPhase 3-53
Adding a New Project and Modelica Model in Twin Builder 3-55
Placing Modelica Components for the SMPM Inverter MultiPhase Model 3-56
Connecting Modelica SMPM Inverter MultiPhase Components 3-58
Adding an Initial Equation and Variable Declaration to the SMPM Inverter MultiPhase Design 3-60
Setting Up Model Properties for the SMPM Inverter MultiPhase Example 3-64
Compiling and Updating the SMPM Inverter MultiPhase Project 3-69
Adding the SMPM Inverter MultiPhase Component to a Twin Builder Schematic 3-73
Defining Outputs and Creating Reports for the SMPM Inverter MultiPhase Design 3-75
Setting Up and Running a Transient Analysis for the SMPM Inverter MultiPhase Design 3-77
Electrical Example: AIMC with Losses 3-79
Adding a New Project and Modelica Model in Twin Builder 3-80
Placing Modelica Components for the AIMC Model 3-81
Connecting Modelica AIMC Components 3-82
Adding an Initial Equation and Variable Declaration to the AIMC Design 3-84
Initial Equation 3-84
Setting Up Model Properties for the AIMC Example 3-90
Compiling and Updating the AIMC Project 3-94
Adding the AIMC Component to a Twin Builder Schematic 3-98
Defining Outputs and Creating Reports for the AIMC Design 3-100
Setting Up and Running a Transient Analysis for the AIMC Design 3-102
Example: Simple Power Train 3-103
Adding a New Project and Modelica Model in Twin Builder 3-105

Placing Modelica Components for the Simple Power Train Model 3-106
Connecting Modelica Power Train Components 3-107
Setting Up Model Properties for the Modelica Power Train Example 3-109
Compiling and Updating the Simple Power Train Project 3-113
Adding the Simple Power Train Component to a Twin Builder Schematic 3-116
Placing and Connecting Twin Builder Components for the Simple Power Train Design 3-117
Setting Up Properties for the Simple Power Train Twin Builder Components 3-119
Defining Outputs and Creating Reports for the Simple Power Train Design 3-121
Setting Up and Running a Transient Analysis for the Simple Power Train Design 3-124
4 - Index **127**

1 - Introduction to Modelica in Twin Builder

Twin Builder is a software package used to design and analyze complex technical systems. Simulation models created with Twin Builder can contain circuit components from different physical domains, block elements, and state machine structures modeled in Modelica.

Twin Builder's simple graphical interface makes even complex models easy to design. Fast and stable simulation algorithms reduce simulation time and provide reliable results.

The various tools used for modeling, simulating, and analyzing are integrated within the Twin Builder application which manages the project files, sets options for both simulation and program environment, runs analyses, and generates reports.

Modelica is an open standard non-proprietary, object-oriented, equation-based language to conveniently model complex physical systems containing, for example, mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents. The language is standardized and developed by the Modelica Association. The association also supports a large open source library developed in Modelica called the Modelica Standard Library (MSL).

As a modeling language, Modelica has the following advantages for modeling engineering systems:

- It is a very capable modeling language, which is well designed considering all physical and signal domains. It allows user to focus on high-level mathematical descriptions of engineering component. While deep knowledge of complex solution enabling topics, such as differential-algebraic equations, symbolic manipulation, numeric solvers, etc., is not required. These solver-related aspects of simulation models are handled by Modelica compiler through complex algorithms behind the scenes.
- The language syntax supports both behavioral and structural definition of a model. For behavioral definition, it is an equation-based language, which enables easy high-level modeling of various engineering components (e.g. springs, resistors, clutches, etc.). And the structural definition supports convenient connection between components, so they can be easily combined into systems and architectures.
- It supports a wide range of modeling formalisms. It allows both causal (often used for control system design) and acausal (often used in creating schematic oriented physical designs) approaches within the same model. And it also supports continuous and discrete behaviors. As a result, both physical design and control design can be done using Modelica.
- It is designed to support effective library development and model exchange. It is built on object-oriented constructs, with JAVA like Syntax, to facilitate reuse of models. Being an

open language also makes it possible to import and export Modelica models between different tools that support the standard.

More information on Modelica can be found here: http://www.modelica.org.

General Description of Contents

Introduction

Provides an overview of the Twin Builder software package for use in designing and analyzing complex technical systems using components, including components created using the Modelica modeling language.

Working with Modelica in Twin Builder

Provides an overview of the Twin Builder environment, the use of the Modelica environment within Twin Builder to create and use Modelica components in Twin Builder designs, analyzing designs and obtaining results, and includes an example workflow for creating, compiling, and simulating a Modelica model in Twin Builder.

Case Studies

Uses six case study examples to provide an in-depth understanding of how to use Modelica models in Twin Builder designs.

Twin Builder Documentation

Use the following guides and manuals to quickly find help while working with Twin Builder.

Table 1: PDF Manuals

Getting Started	Program functionality at a glance, new functions, step-by-step simulation examples, program conventions. Click **Help** > **Twin Builder Getting Started Guides** or **Help** > **Twin Builder PDFs** > **Twin Builder Getting Started Guides** > **Twin Builder Getting Started Guide**.
Modelica Tutorial	Detailed descriptions and examples of Twin Builder's Modelica functionality (this guide). Listed under **Help** > **Twin Builder PDFs** > **Twin Builder Getting Started Guides** > **Modelica Tutorial**
Installation Guide	Refer to the Ansys EM Suite Installation topic. This topic provides links to the Windows *Installation Guide* and the *Platform Support* website.

Table 2: Help

Help	Click **Help** > **Twin Builder Help** to access the Help with index and search capabilities available for the Twin Builder programs and installed models.

	Help also has a description of additional Twin Builder modules such as Interface and Coupling elements, C-interface, and optimization algorithms. Click **Help** > **Twin Builder PDFs** > **Twin Builder Help** to access the PDF of the Twin Builder Help.
Examples	Examples are available for installed Twin Builder models. Open the example by right-clicking on the model in the Project Manager and selecting **Load Example**. If Twin Builder was installed in the default location, Twin Builder application examples are available in **"C:\Program Files\ANSYS Inc\v252\AnsysEM\Examples\Twin Builder\Applications"** and the Modelica Tutorial examples are available in **"C:\Program Files\ANSYS Inc\v252\AnsysEM\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples"**

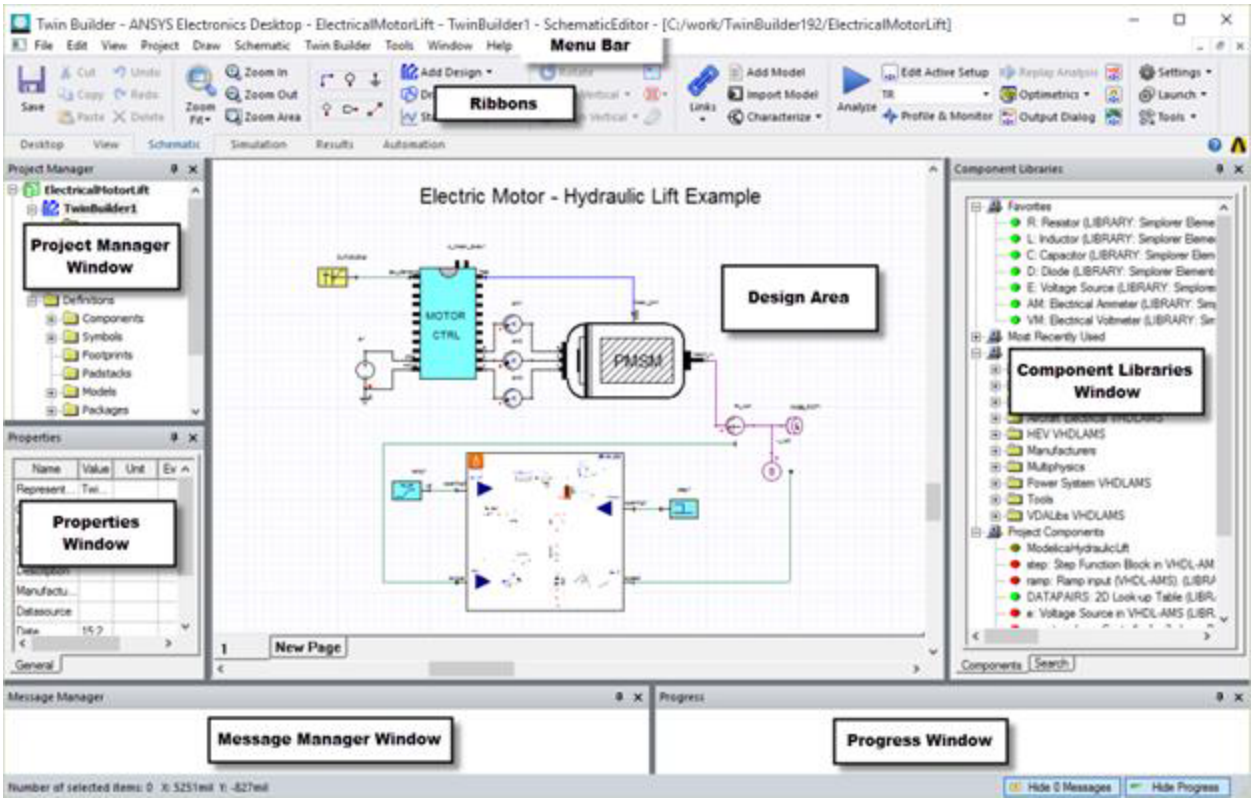
2 - Working with Modelica in Twin Builder

The topics in this section:

- Provide an overview of the Twin Builder user interface, Twin Builder projects and the Project Manager window, and Twin Builder components.
- Explain basic user operations: creating a Modelica component in Twin Builder, using a Modelica component in a Twin Builder design, analyzing designs, and plotting results.
- Provide an example workflow for creating, compiling, and simulating a Modelica model in Twin Builder.

Overview of the Twin Builder Interface

The figure below and the table following describe the major Twin Builder interface elements for an existing project with its associated schematics.



Project Manager	The Project Manager window shows all the components, models, and other elements of each design in the project. Each project has its own expandable

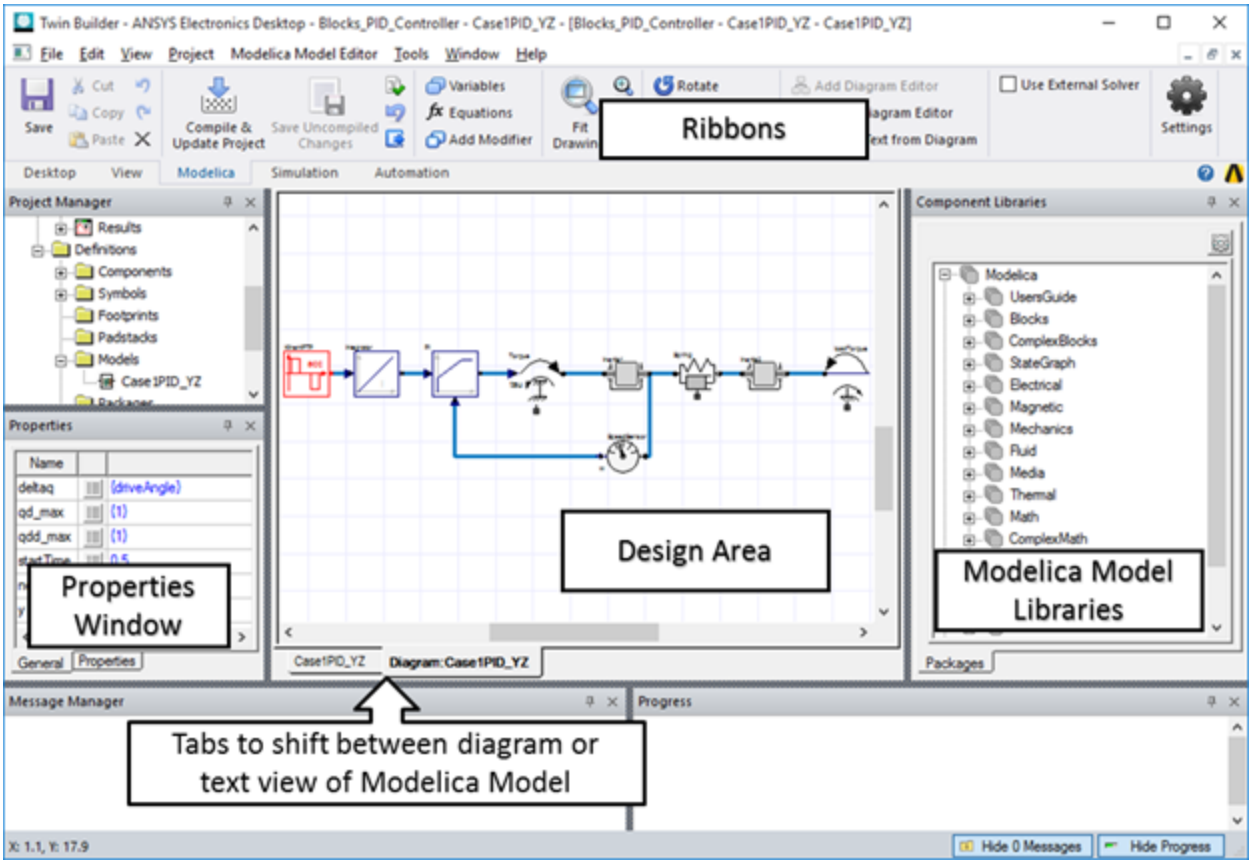
window and Project Tree	*Project Tree*. Many operations on the design elements can be performed directly from the Project Manager window.
Component Libraries window	Displays, on the Components tab, the component categories, including Favorites, Most Recently Used, Simplorer Elements, and Project Components. The Project Components section lists the elements that are active in your projects. It also provides a search feature on the Search tab. Note that the Modelica libraries are not displayed in this window, because Modelica models cannot be directly used in Twin Builder Design. Only after a Modelica model is compiled in a project and a component is generated, will it be displayed in this window under Project Component. The Modelica libraries are displayed only in the Twin Builder Modelica Environment.
Message Manager window	Displays error, informational, and warning messages for the active project.
Progress window	Displays solution progress information.
Properties window	Displays the attributes of a selected object in the active model, such as the object's name, electrical or other associated physical quantities, orientation, and color. Also displays information about a selected command that has been carried out. For example, if a circle was drawn, its command information would include the command's name, the circle's center position coordinates, and the size of its radius.
Design area window	Displays one or more editor windows such as the Schematic Editor, model editors, and symbol editor. It also displays various report windows.
Menu bar	Provides various menus that enable you to perform Twin Builder tasks, such as managing project files, designs, and libraries; customizing desktop components; drawing objects; and setting and modifying project parameters and options.
Ribbons	Provides tabs containing icons that act as shortcuts for executing various commands.
Status bar	Shows current actions and provides instructions.

Note	The screen layout may be different from what is shown in the picture above, depending on the **View** menu settings.

For additional information, please refer to the help and the *Twin Builder Getting Started Guide*.

Overview of the Twin Builder Modelica Environment

When adding a Modelica model, or editing an existing one, a Twin Builder Modelica environment dedicated for Modelica modeling is presented for the user. This environment resembles the Twin Builder schematic editor environment. The following figure and table describe the areas that differ from the schematic editor environment when the Twin Builder Modelica environment is active.



Ribbons	Provides a Modelica tab containing icons that act as shortcuts for Modelica special commands.
Component Libraries window	Displays only the Modelica libraries, which can be used in the Modelica Environment. Library file supported here is the Modelica package file, which is formatted differently than the main Twin Builder libraries.
Properties window	Displays Modelica model properties of a selected instance in the diagram editor. User can also edit property values here. Also displays information about a selected command that has been carried out. For example, if a circle was drawn, its command information would include the

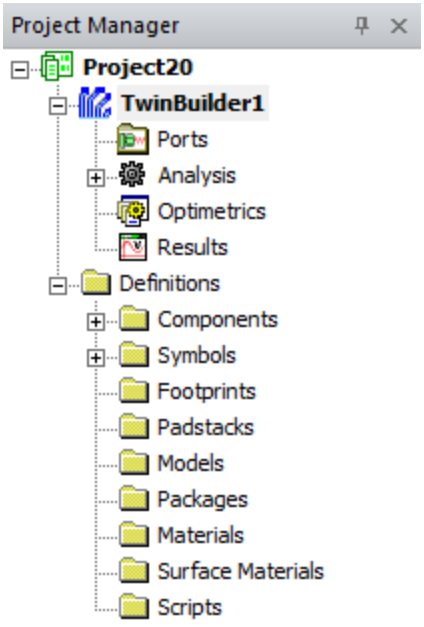
	command's name, the circle's center position coordinates, and the size of its radius.
Design area window	Provide two views for a Modelica model: text, and diagram. You can switch between the two views using the tabs at the bottom of this window. You will be asked whether to overwrite from diagram when switching to the text view, or when compiling. This is because a model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated.

Starting Twin Builder and Creating a New Project

Twin Builder can be started like all Windows applications - by using the **Start** menu on the Windows task bar.

1. Click **Start**, and select **Programs** > **Ansys EM Suite** *version* > **Ansys Twin Builder** *version*.

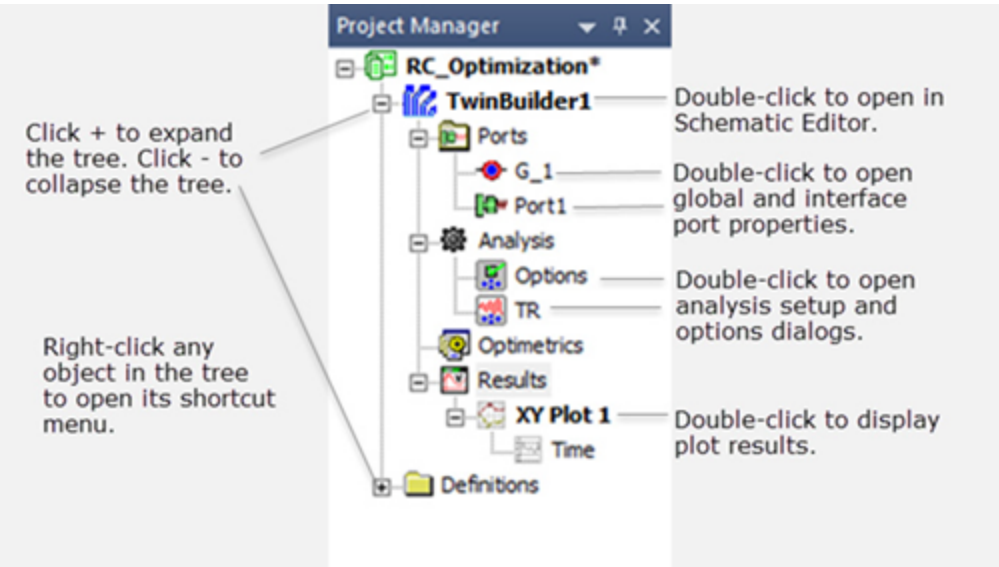
   By default, opening Twin Builder creates a new project named **Project***n* and inserts a new design named **TwinBuilder***n*, where *n* is the order in which each was added to the current session. A project is a collection of one or more designs saved in a single ***.aedt** file.



   The new project, which appears in the **Project Manager** window, contains a file structure that organizes design elements such as Ports, Analysis, Optimetrics, and Results. Project

Definitions such as Components, Symbols, Models, Packages, Materials, and Scripts are also listed.

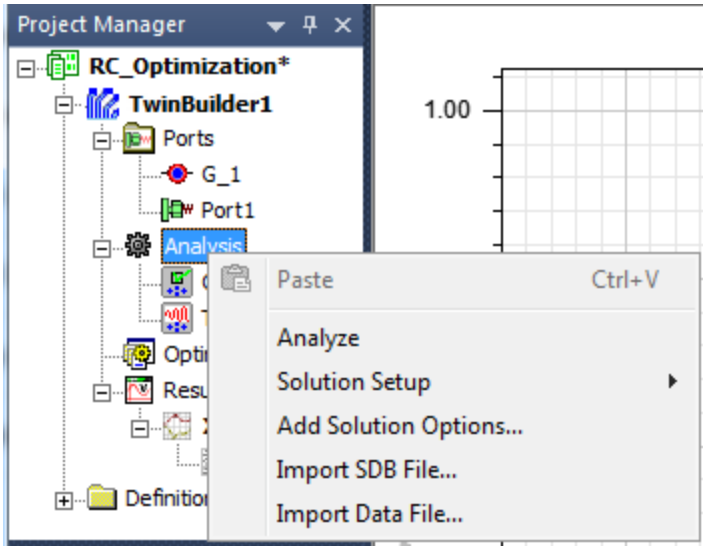
The Project Manager window shows the projects loaded into Twin Builder. Each project may consist of one or more designs. The Project Manager window contains details about all open Twin Builder projects in a form known as the project tree, as shown below:



The top node listed in the project tree is the project name. Expand the project icon to view all the Twin Builder design information and definitions for the project. If there are multiple designs in a project, each design's ports, analysis setup, and results are displayed as entries in a separate sub-tree.

The label of the selected Project Tree element that is active (that is, has focus) may be rendered in bold text; or a small "window" icon may display next to the selected Project Tree element. The icon will be gray if the editor or plot window associated with the selected element is closed, or not in focus. Clicking the gray icon will open the window and bring it into focus.

Right-clicking a folder or item opens a shortcut menu that allows you to perform various operations. For example, right-clicking the analysis icon for a design displays a menu similar to this:



Virtually all project editing and management can be done from the **Project Manager** window using the right mouse button shortcut menus. Many of these menus are also available from the main menu bar in the **Twin Builder** menu.

Once you insert a Twin Builder design into a project, it is listed as the second node in the project tree. It is named TwinBuilder*n* by default, where *n* is the order in which the design was added to the project. Expand the design icon in the project tree to view all of the specific data about the model, including its solution and post-processing information.

The TwinBuilder*n* node contains the following project details:

Ports	Displays the *interface* and *global* ports added to a Twin Builder design.
Analysis	Displays the *solution setups* and *solution options* for a Twin Builder design. A solution setup specifies how Twin Builder will compute the solution. The solution options provide settings such as the integration formula, iterations, and maximum and minimum errors.
Optimetrics	Displays any Optimetrics setups added to an Twin Builder design.
Results	Displays any post-processing reports generated.

Refer to *The Project Manager Window* in the main Twin Builder help for more information.

2. To rename the design, right-click **Twin Builder***n* in the project tree, and choose **Rename** on the shortcut menu.
3. Type a name of your choosing, and then press **Enter** to complete the change.

Note	To use the example files provided for this Tutorial, choose **File** > **Open Examples** from the main menu and navigate to the example ***.aedt** files in the **Tutorial Examples** folder in **"C:\Program Files\ANSYS Inc\v252\AnsysEM\Examples\Twin Builder\Modelica Tutorial"**.

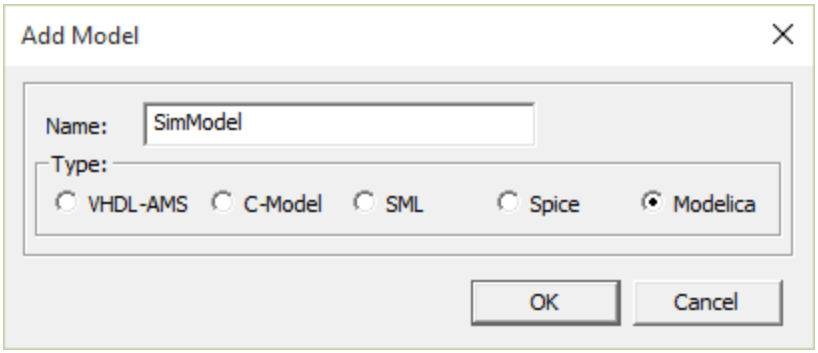
The Twin Builder Component

A Twin Builder component (for example, a resistor, a transistor, or a motor) has several definitions associated with it. It needs a **Symbol** definition to describe the appearance of its schematic symbol, a **Model** definition to define the characteristic behavior of the component, and a **Component** definition as a kind of high-level wrapper.

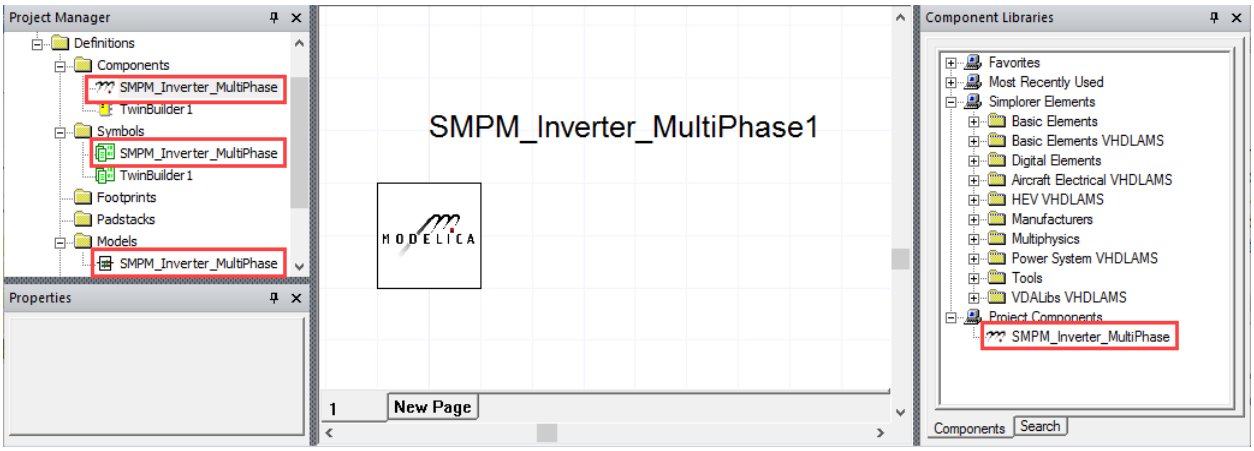
In Twin Builder projects, library component definitions are accessed once per component – when you place the first instance of a component onto a schematic. Once a component is placed, all the definitions for that component are transferred (copied) from the library to the project file.

Editing component definitions and updating instances on a schematic are then controlled from the project definitions, which are listed in the Definitions folder in the **Project Manager** window. In other words, you edit the various definitions associated with a component in the *project*, not in the library.

For a component's model definition, Twin Builder supports multiple modeling languages, as shown below in the **Add Model** dialog box. Modelica is one of the supported languages. Adding or editing a Modelica model is done using the Twin Builder Modelica Environment.

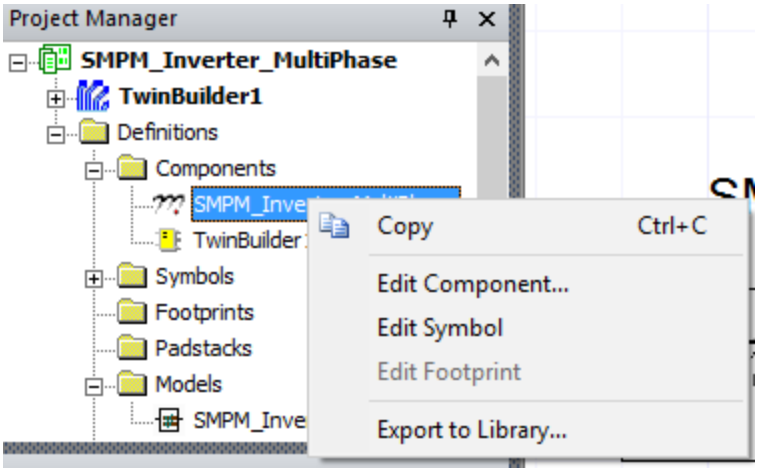


After a Modelica model is compiled, Twin Builder creates a component of the same name in the project definitions. The component, and its associated symbol and model, are local to the project. You can locate the component either in the project tree **Definitions** > **Components** in the **Project Manager**, or in the **Component Libraries** window, under **Project Components** as shown in the figure below. Within the project, you can drag-and-drop the Modelica component onto the schematic of a design, where it can be connected with other Twin Builder components, and simulated with the design.



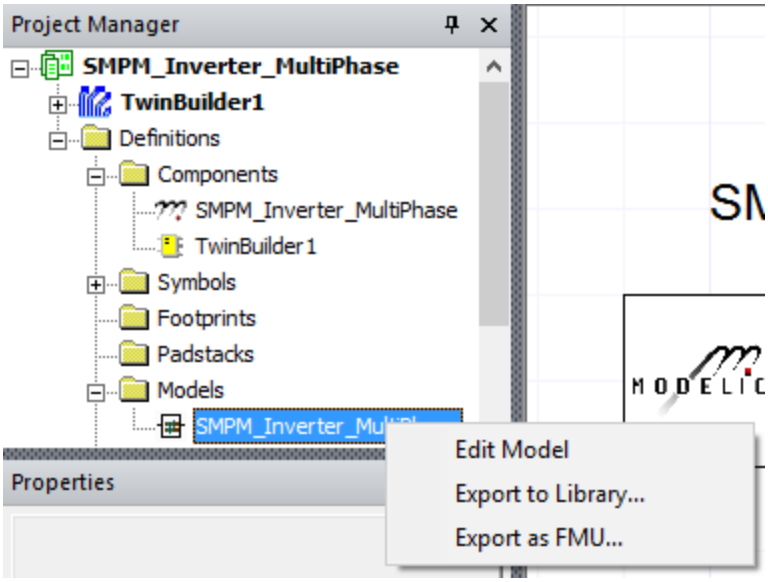
You can right-click the component in the project tree to:

- Edit the component's symbol, which changes its appearance on the schematic.
- Export the component to a user library (UserLib) or personal library (PersonalLib). (Refer to the Twin Builder help for information on *Working with Libraries*.)



You can also right-click the model to:

- Edit the model, which opens the Modelica environment for editing.
- Export the compiled Modelica model to a user or personal library.
- Export the model as a Functional Mock-up Unit (FMU).



Related Topics

"Overview of the Twin Builder Modelica Environment" on page 2-3

Creating a Modelica Component in Twin Builder

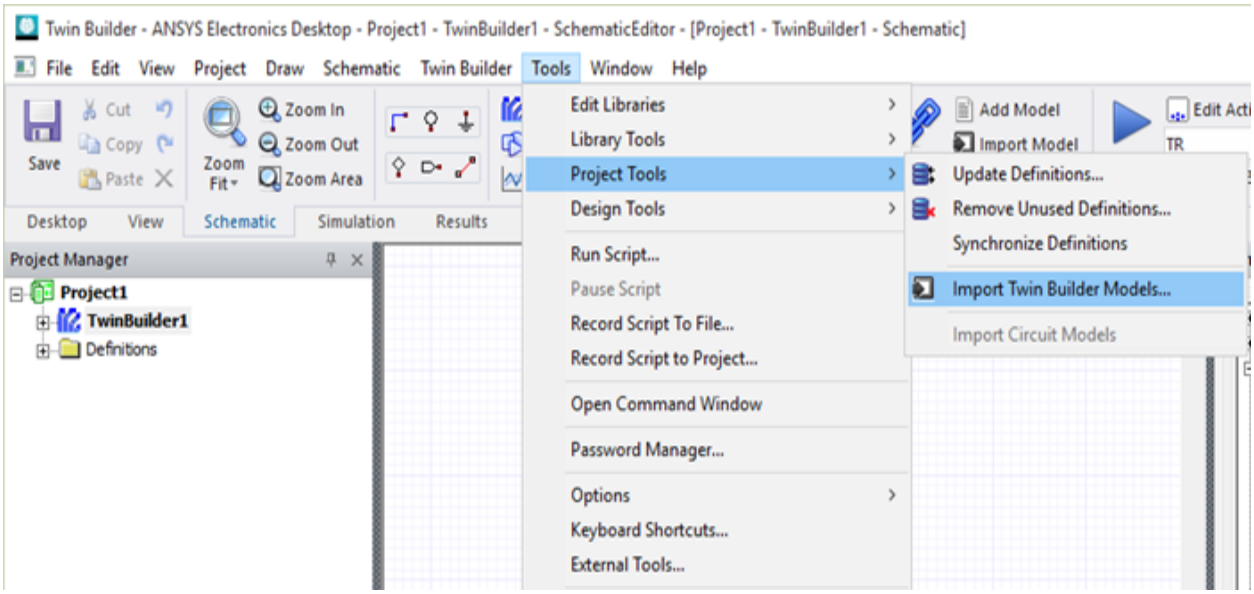
You can create a Modelica component in Twin Builder in either of two ways:

- Import an existing Modelica model **.mo** file.
- Use the Twin Builder Modelica diagram editor.

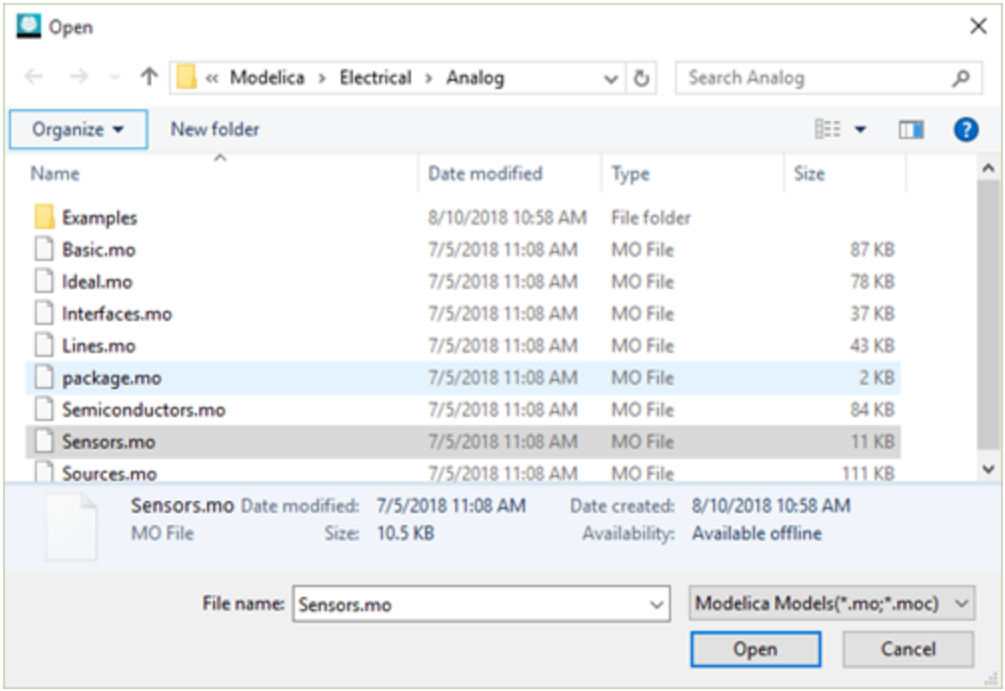
Importing a Modelica Model File

To create a Modelica component in Twin Builder by importing a Modelica model **.mo** file:

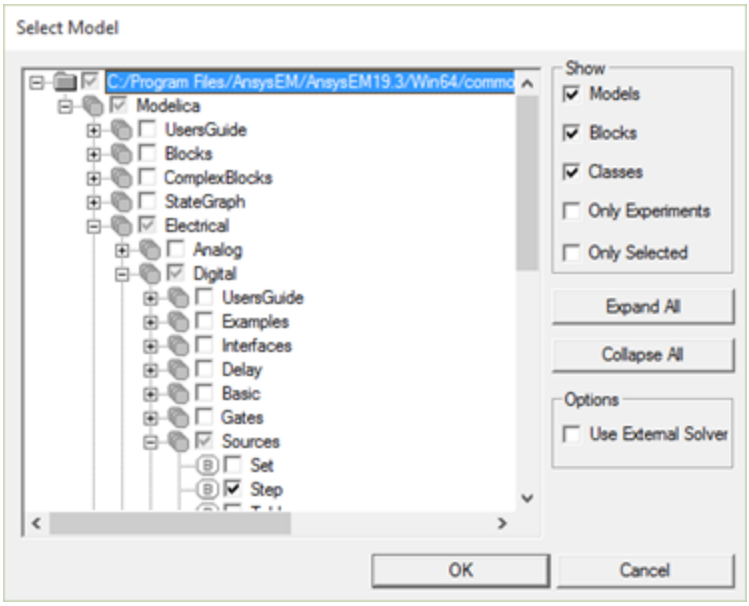
1. In a Twin Builder project, select: **Tools** > **Project Tools** > **Import Twin Builder Models**.



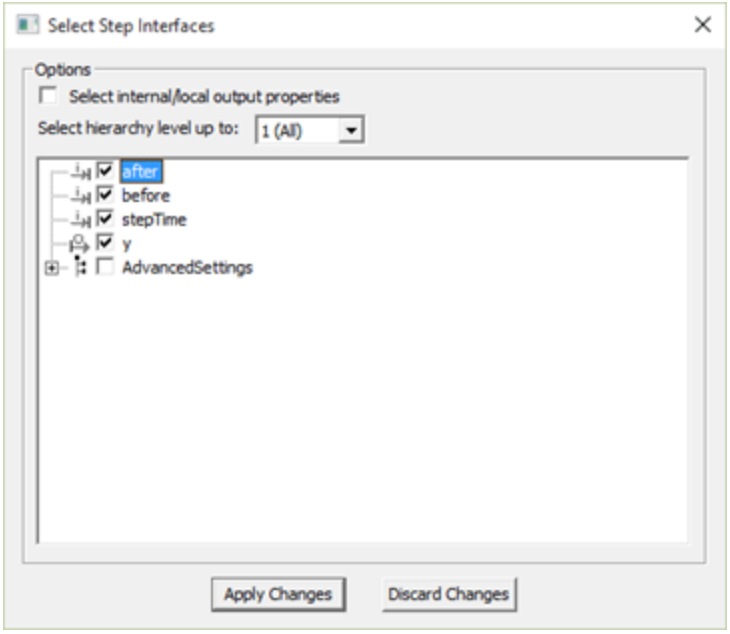
2. Navigate to the Modelica model you want to import, and click **Open**.



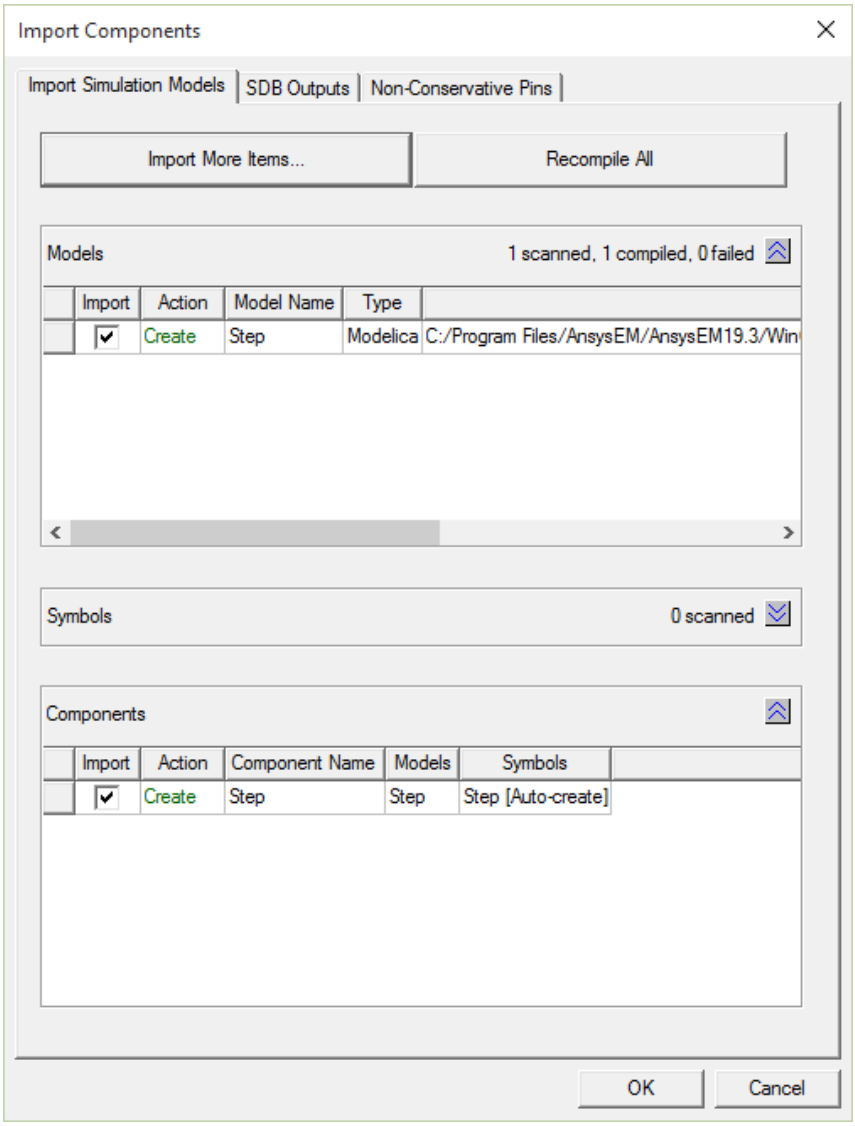
3. In the **Select Model** dialog box, select the check boxes to make sure the desired model's information is transferred into Twin Builder, then click **OK**.



4. Select interface and other properties for the imported Modelica model.



5. After the Modelica model, function, or package has been successfully transferred into Twin Builder, the following dialog box appears.



6. Click **OK** and the entire model/function/package is included in one component in the Project Manager tree in the **Definitions** > **Components** folder. You can now select a component and place it on the schematic.

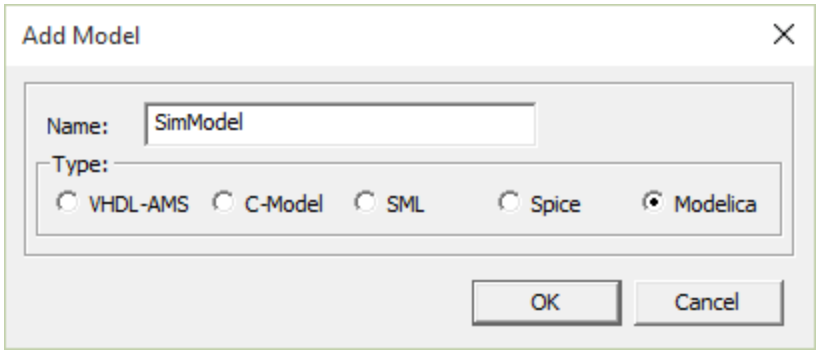
Using the Diagram Editor to Create a Modelica Model

To create a Modelica component in Twin Builder using the Modelica Model diagram editor:

1. In a Twin Builder project, do one of the following to create a Modelica model:

   - Click **Twin Builder** > **Add Model**.
   - Right-click the **Models** folder under **Definitions** in the **Project Manager** pane and select **Add Definition**.
   - Press Ctrl+Shift+M.

     The **Add Model** dialog box appears. Ensure that the Modelica type is selected.



2. Enter a name for the model and click **OK**. The model file opens in Twin Builder's Modelica model diagram editor in a separate Diagram tab.

Note	If an editable model has been imported, you must add a new Diagram Editor by clicking the **Add Diagram Editor** icon in the Modelica ribbon, or by clicking **Modelica Model Editor** > **Add Diagram Editor**.

3. From the right side of the Diagram Editor, drag and drop the component you want on the Diagram Editor. Press **Esc** to complete the action.

   - To move a component after placing it, select and drag it.
   - To rotate a component before placing it, press **R** before releasing the mouse click.
   - To rotate a component after placing it, select it and press **Ctrl +R**, which rotates it by 90°, or right-click the component and select the **Rotate** option.

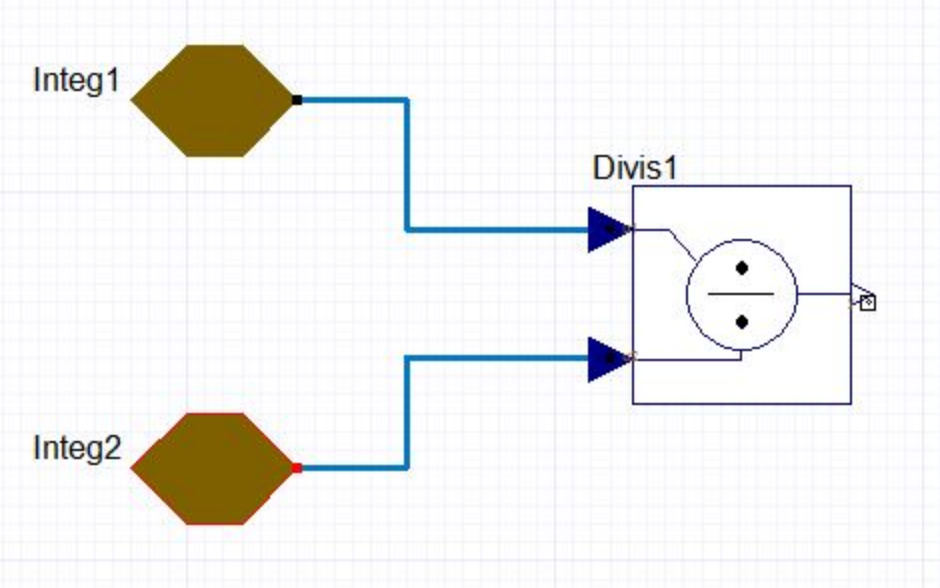
Note	When you rotate a block or component, the diagram editor removes wires connected to unselected components. You must reconnect the blocks and components.

- To delete a component, select it and press **Delete**, or right-click and select **Delete**.

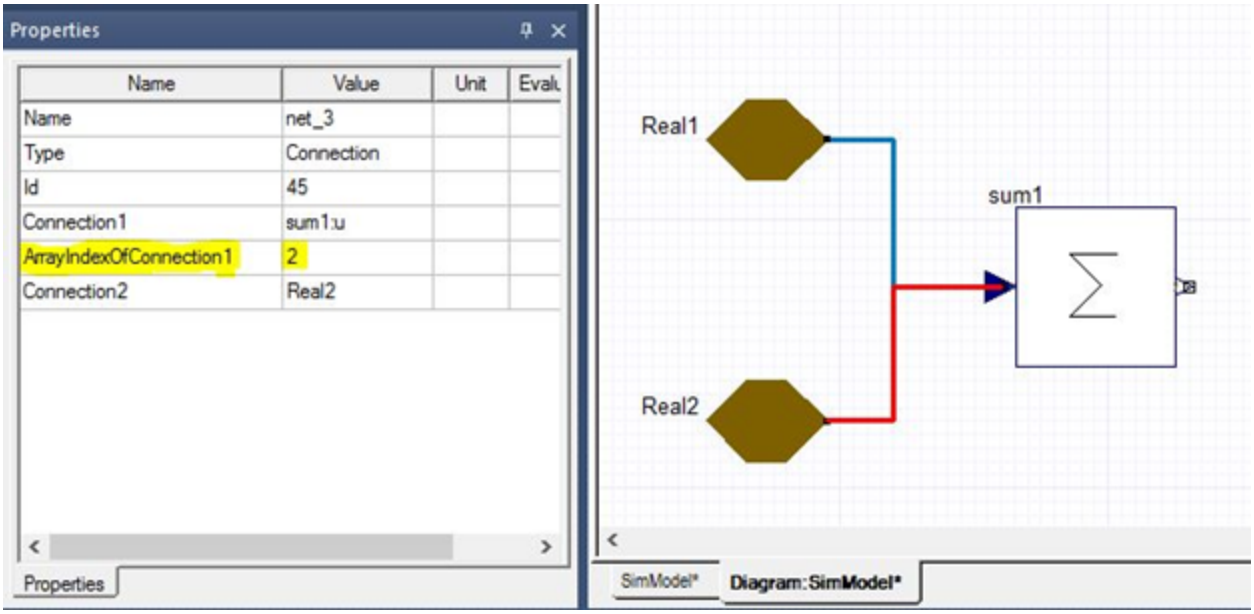
Note	When you delete a block or component, the Diagram Editor removes all connected wires.

For more information, refer to *Component Libraries* in the Twin Builder help.

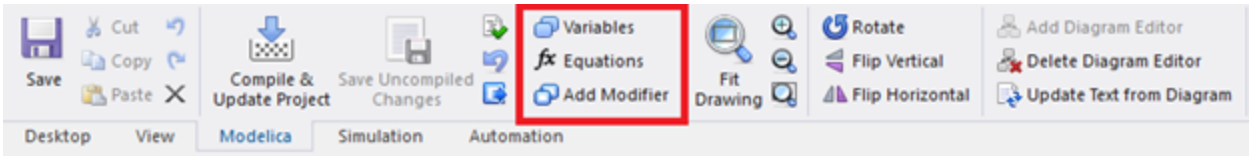
4. After arranging the components, use wires and connectors to connect the components. Click on the input/output pins of the component and connect them to the desired pins.



5. For Modelica models, if the component has an array interface port and you make a connection, you can select the connection and change the default value of 1 for **ArrayIndexOfConnectionX** in the Properties window.

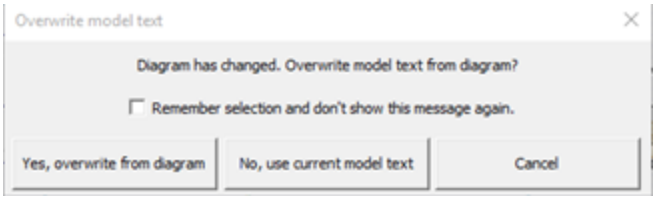


6. After you complete the diagram, edit the properties and parameters of the components in the Properties window.
7. Variable's attributes inside a Modelica instance cannot be directly changed in the Properties window. If you want to do so, you can add modifier to this instance by select it in the diagram, then click **Add Modifier** in Modelica ribbon.

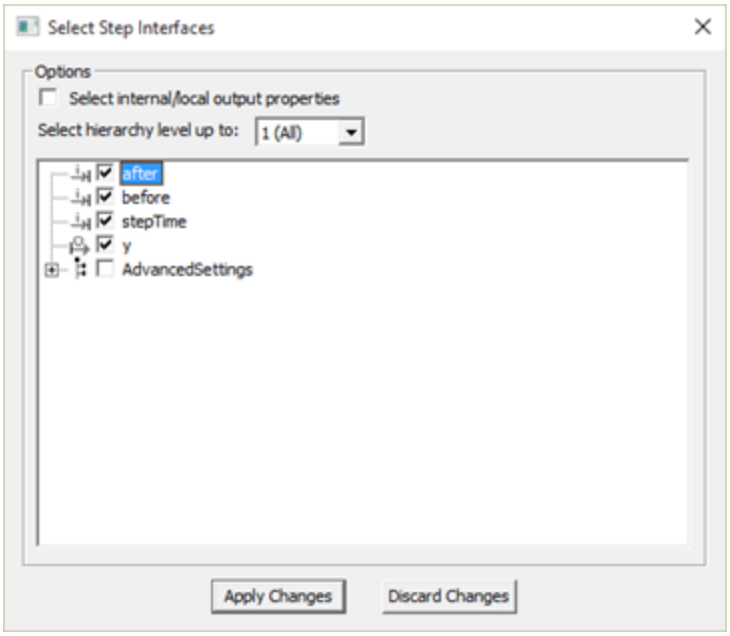


8. You can also declare variables or add equations by clicking the **Variables** and **Equations** icons in Modelica ribbon.
9. To update the model file text, click **Modelica Model Editor** > **Update Text from Diagram** or click the **Update Text from Diagram** icon on the Modelica ribbon. For more information, refer to *Updating Text from Diagram* in the Twin Builder help.
10. Now the Modelica model is complete, it is ready to be compiled and added to the Twin Builder project as a component. Click **Compile & Update Project** on the Modelica ribbon tab.
11. A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. Thus, after clicking **Compile & Update**

Project, an overwrite model text dialog will be displayed. The same dialog is also displayed whenever you switch tabs to SimModel to view Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text and not overwrite from the diagram. The latter choice won't apply any change to the diagram since the last save or compile.

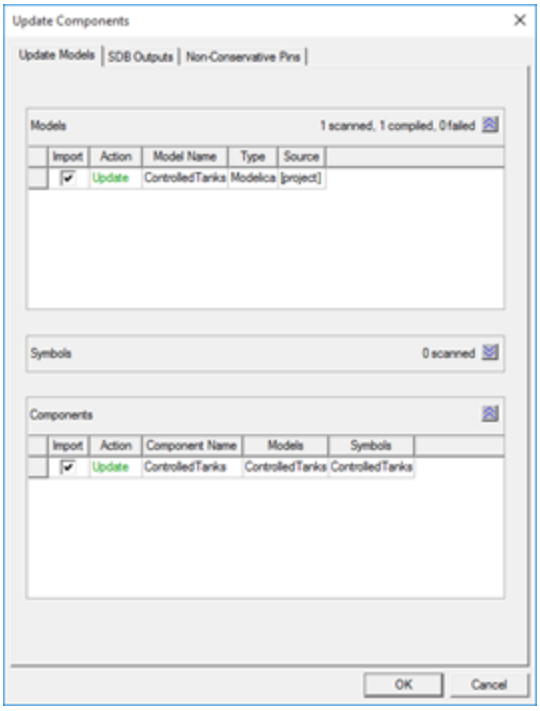


12. Following the compile operation, you can select interfaces for the component, which will be accessible in the higher-level Twin Builder design. Click **Apply Changes** to continue.



13. The resulting **Update Components** dialog box allows you to confirm whether to create orupdate an existing Twin Builder component.

    Click **OK** and the Modelica component is ready to use in a Twin Builder design.



You can find the newly created Modelica component in the Project Manager's **Definitions** > **Components** folder.

Using a Modelica Component in a Twin Builder Design

After a Modelica component is created in a project, you can locate it in the project tree under **Definitions>Components**, and drag and drop it in a Twin Builder design schematic.

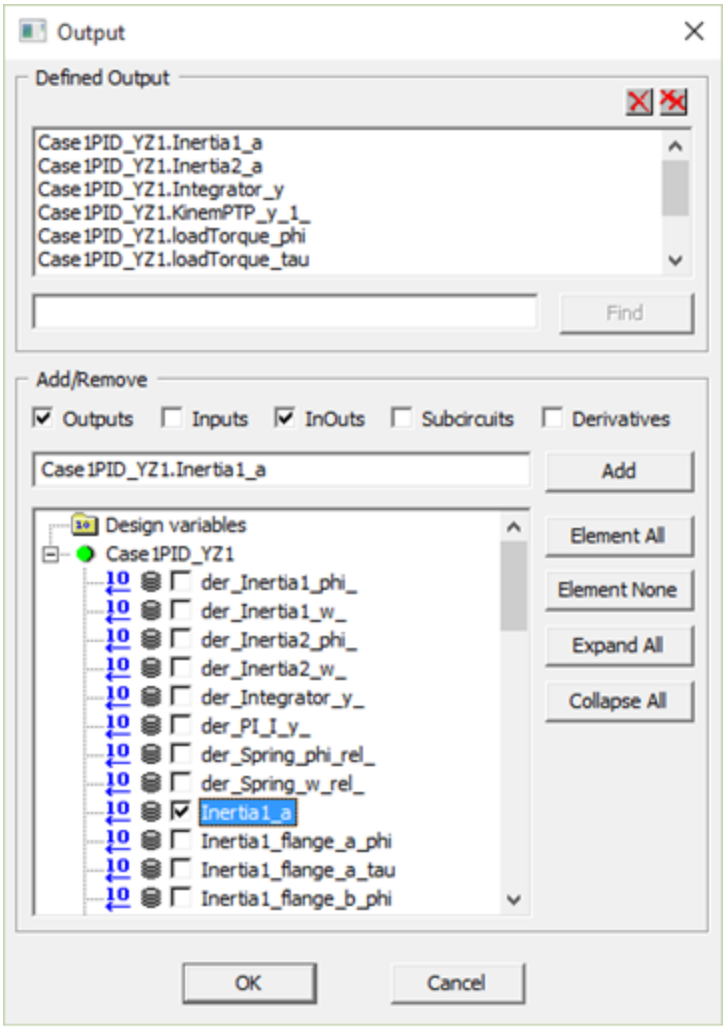
If a complete system is built inside the Modelica component, it is ready to simulate and plot results as described in ***Analyzing a Design and Plotting Results***.

You also have the option to implement a subsystem inside a Modelica component, then connect the Modelica component with other Twin Builder components in a design schematic. For an example of this workflow, refer to the case study: ***Simple Motor Power Train***.

Analyzing a Design and Plotting Results

Twin Builder enables you to set up and run analyses (simulations) on the designs you construct in the Schematic Editor, using the simulator engine in the background for running the analyses. For a given simulation, you typically have an interest in looking at the results for specific conditions in the schematic. To create reports or plots of simulation results, you must first define outputs in the **Output** dialog box to make them available for plotting. Running an analysis then provides results for any reports and plots you create.

1. To define outputs for plotting, click on **Output Dialog** on the **Schematic** tab of the ribbon.

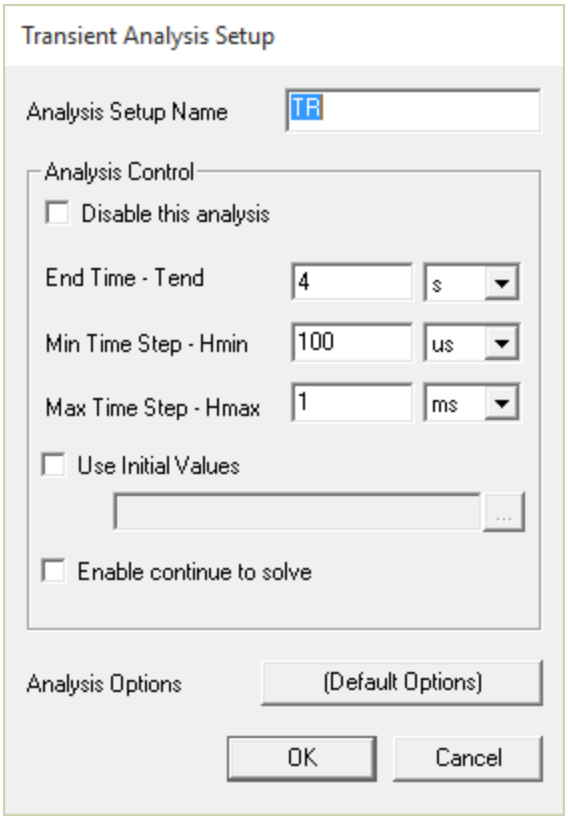


2. Select the outputs that are needed for your plots. (Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.)

3. In the Project Manager Results context menu, you can then select from a list of report types under **Create Standard Report** for the design to be simulated. The defined outputs will be available for plotting. For details of viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder Help.

4. By default, a transient analysis **TR** is created for each design. You can find it in the Project Manager under **Analysis** for the design. A transient analysis simulates the time-domain response of the system. Double-click **TR** to set up the end time and desired time step for the simulation. For more details of transient analysis, see *Transient Analysis Setup* in the

Twin Builder help.



5. To start the transient simulation, do one of the following:
   - On the ribbon Schematic tab, click the blue arrow **Analyze** icon.
   - In Project Manager, right-click on TR of the design, and select **Analyze**.
6. When simulation is complete, you can review reports under **Results** in Project Manager.

Workflow for Creating, Compiling, and Simulating a Modelica Model in Twin Builder

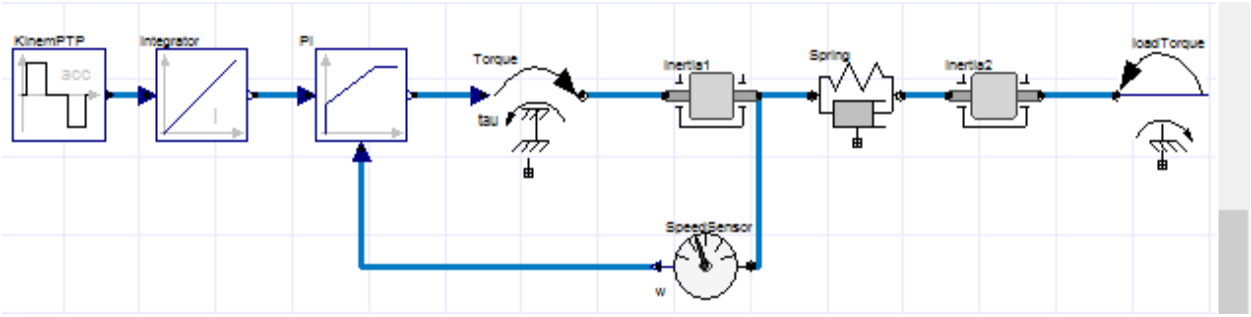
Modelica Example: PID Controller

This example illustrates the workflow for creating, compiling, and simulating a Modelica model of a simple drive train controlled by a PID controller. The Modelica system diagram is shown below. The model consists of a causal connection (including Torque, Inertias, Spring, load Torque) and an acausal control loop (including KinemPTP, Integrator, PI, and SpeedSensor). The two blocks **kinematic_PTP** and **integrator** are used to generate the reference speed. To check whether the system starts in steady state, the reference speed is zero until time = 0.5 s, and then follows the sketched trajectory. The **PI** block is a PI controller in which several practical important

properties, such as anti-windup-compensation, have been added. The output of the controller is a torque that drives a motor inertia, **inertia1**. The load inertia, **inertia2** is attached using a compliant spring/damper component. A constant external torque of 10 Nm is acting on the load inertia.

This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\Blocks_PID_Controller.aedt



PID Controller Example Workflow

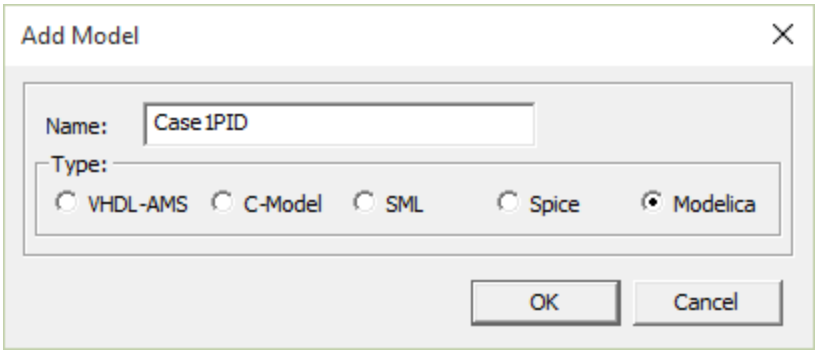
In this example, we will create a Modelica PID P2P model, and analyze its performance in the Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the new model into the **Diagram Editor**.
- Connect the Modelica components.
- Provide an initial equation and variable declarations.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Define outputs and Create Reports for the simulation results.
- Set up and run a Transient analysis.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.
2. Create a new Modelica model by doing one of the following:

- Select **Twin Builder** > **Add Model**.
- Press Ctrl+Shift+M.
- In the Program Manager **Definitions** folder, right-click **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

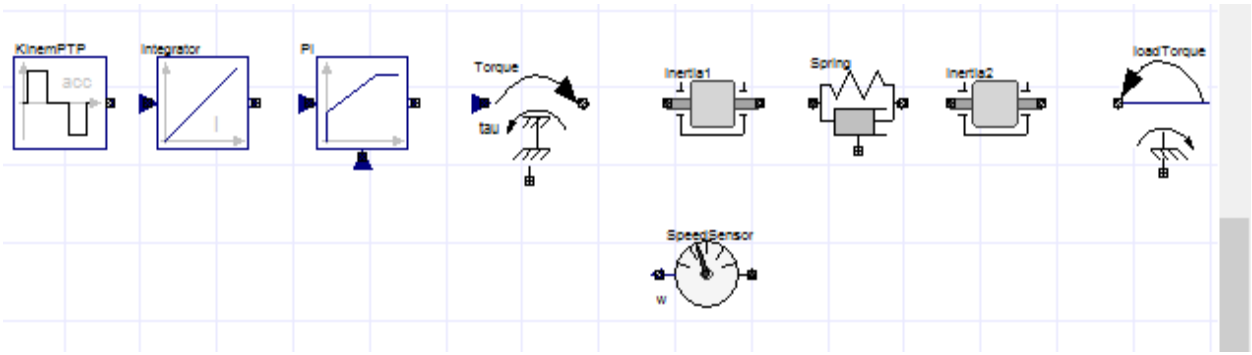
Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

Placing Modelica Components for a New Model

All of the required components for the PID Controller example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Modelica Library
KinemPTP	Modelica/Blocks/Sources/KinematicPTP
Integrator	Modelica/Blocks/Continuous/Integrator
PI	Modelica/Blocks/Continuous/LimPID
Torque	Modelica/Mechanics/Rotational/Sources/Torque
Inertia1, Inertia2	Modelica/Mechanics/Rotational/Components/Inertia
Spring	Modelica/Mechanics/Rotational/Components/SpringDamper
loadTorque	Modelica/Mechanics/Rotational/Sources/ConstantTorque
SpeedSensor	Modelica/Mechanics/Rotational/Sensors/SpeedSensor

1. Locate each of the components in the library.
2. Drag and drop the components into Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. (Connections will be added later.)

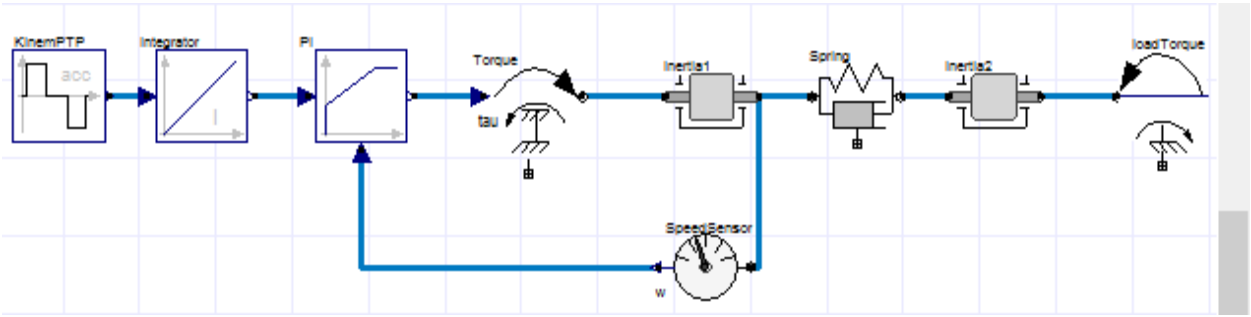


3. You can change the instance names for the components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press **Esc** to complete each action.

Next, you will add connections between the various components.

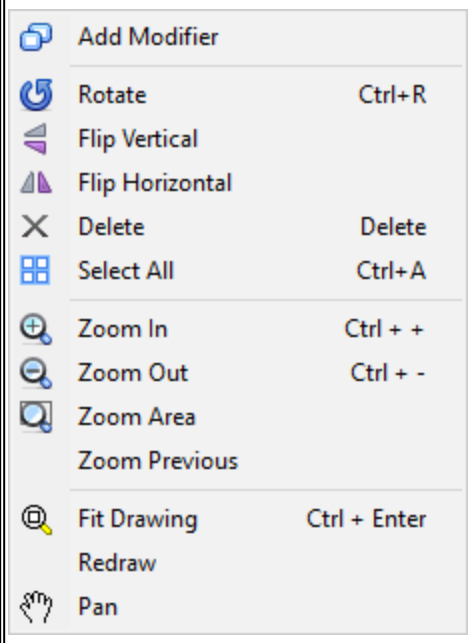
Connecting Modelica PID Controller Components

To connect the components for the PID Controller as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will add an initial equation and declare a variable for the PID Controller model.

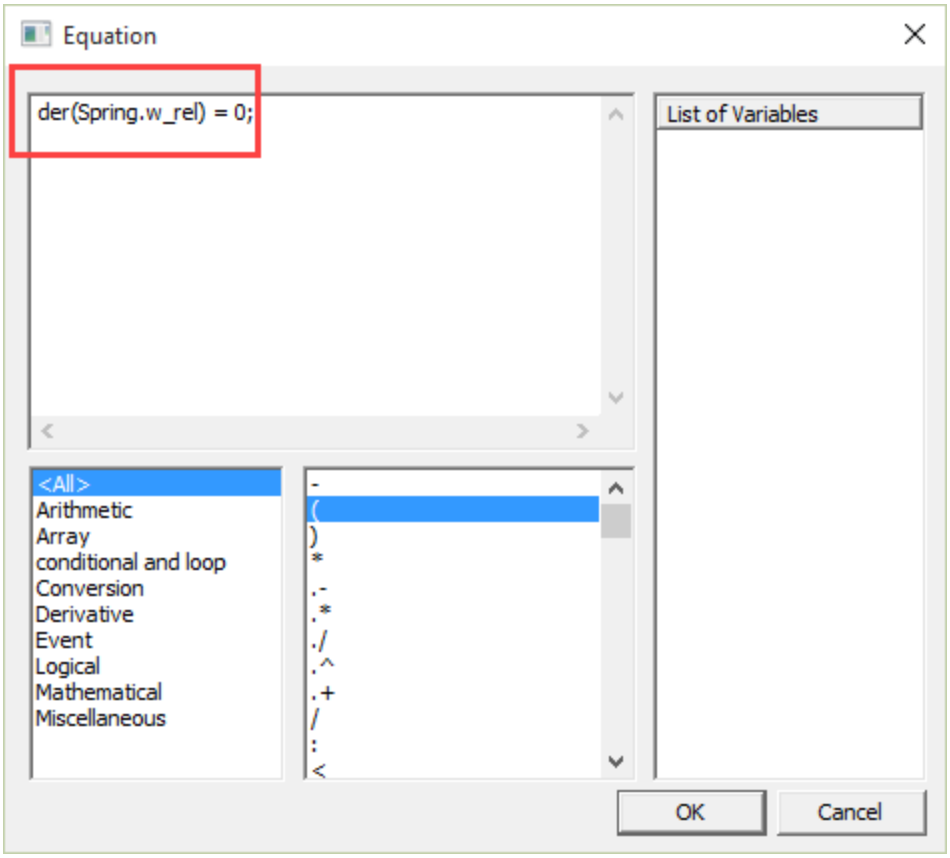
Adding an Initial Equation and Variable Declaration to the PID Controller

For this example, an initial equation is needed for spring initialization, and a drive angle variable declared.

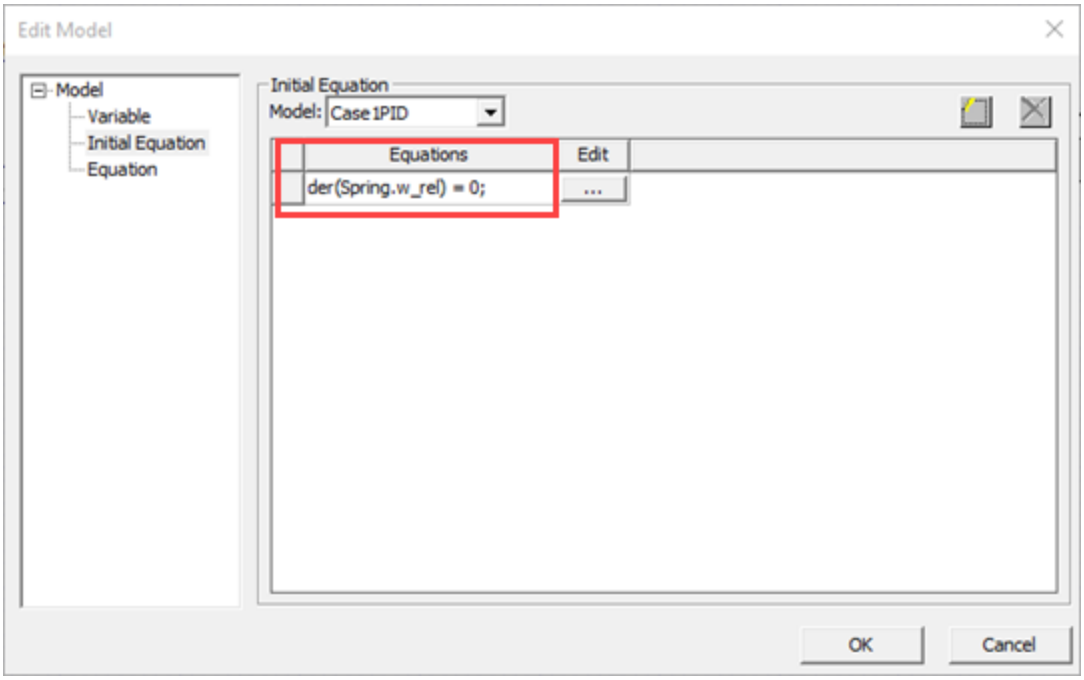
Initial Equation

1. For the initial equation, on the Twin Builder **Modelica** tab, click the Equations icon to open the **Edit Model** dialog box.
2. Select **Initial Equation**, then click the  button in the upper right of the dialog to open the **Equation** editor, and enter the equation:

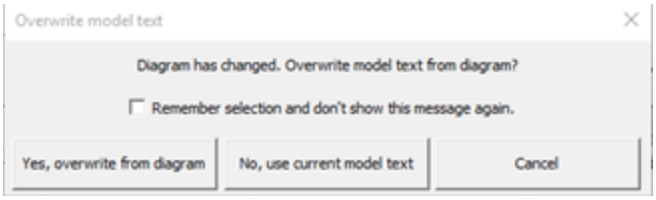
   der(Spring.w_rel) = 0;



3. Click **OK** to close the dialog box, adding the equation to the **Edit Model** dialog box.



4. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog boxdisplays.



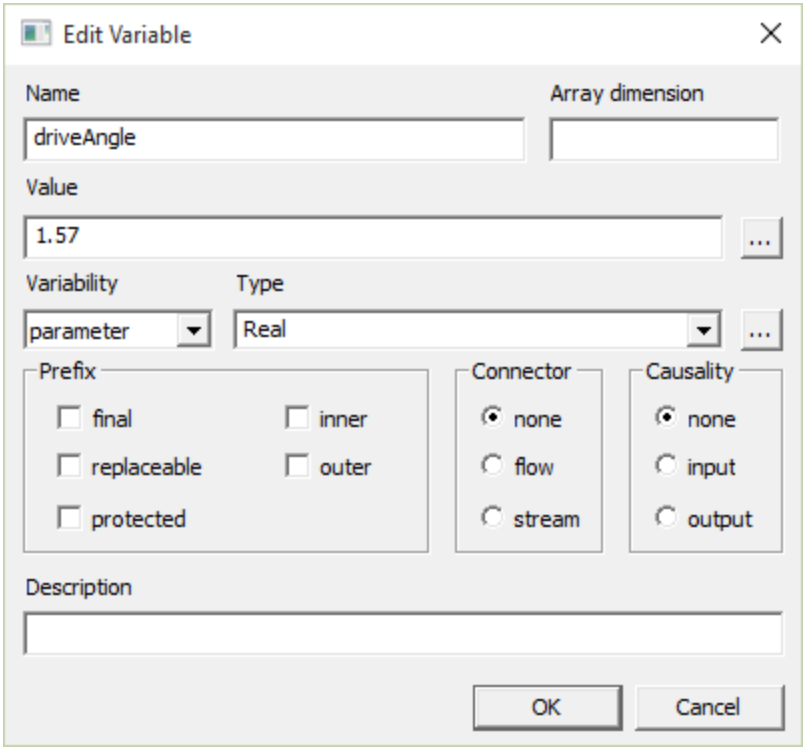
5. Click **Yes, overwrite from diagram** to generate the equation and add it to the model text.

```
initial equation
    der(Spring.w_rel) = 0;
```

Variable Declaration

1. To create a new variable for the real parameter **driveAngle**, on the Twin Builder **Modelica** tab, click the **Variables** icon to open the **Edit Model** dialog box.

2. Click the button in the upper right of the dialog to open the **Edit Variable** dialog box, and enter the settings as shown below.



3. Click **OK** to close the dialog box, adding the variable to the **Edit Model** dialog box.
4. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog box displays.
5. Click **Yes, overwrite from diagram** to add the variable declaration to the model text.

```
//Declaration(s)
parameter Real driveAngle = 1.57;
```

Next, you will set up properties for the PID Controller model.

Setting Up Model Properties for the PID Controller

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
KinemPTP	deltaq = {driveAngle}

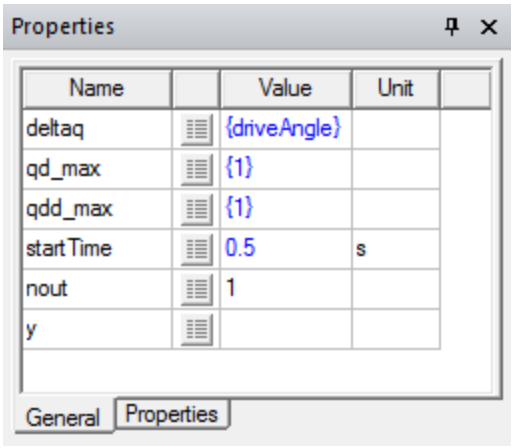
Component Instance	Properties
	qd_max = {1} qdd_max = {1} startTime = 0.5
Integrator	Use the default values.
PI	controllerType = Modelica.Blocks.Types.SimpleController.PI Ti = 0.1 yMax = 12 wp = 1 k = 100 limitsAtInit = false Ni = 0.1 initType = Modelica.Blocks.Types.InitPID.SteadyState
Torque	Modelica/Mechanics/Rotational/Sources/Torque
Inertia1	J = 1 phi(fixed = true,start = 0) a(fixed = true,start = 0)
Inertia2	J = 2
Spring	c = 1e4 d = 100
loadTorque	tau_constant = 10
SpeedSensor	Use the default values.

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options as needed.

For example, to set properties for the **KinemPTP** component in the Diagram Editor:

1. Click on the **KinemPTP** component to display its Properties window.



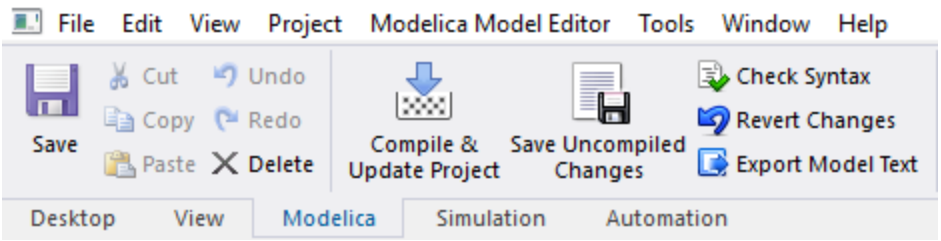
2. On the **General** tab, per the information in the above table, enter **{driveAngle}** for the **deltaq** property; **{1}** for the **qd_max** and **qdd_max** properties; and **0.5** for **startTime**. **nout** remains at its default value of **1**. No changes are needed on the Subcircuit tab. The Properties tab allows you to set the name of the component instance.

Next, you will compile and update the project.

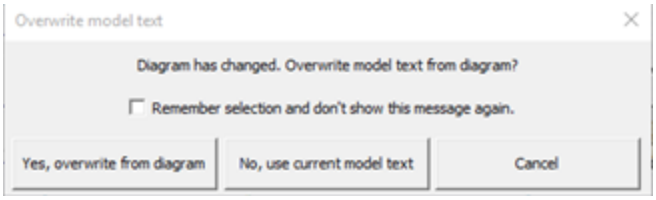
Compiling and Updating the PID Controller Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.



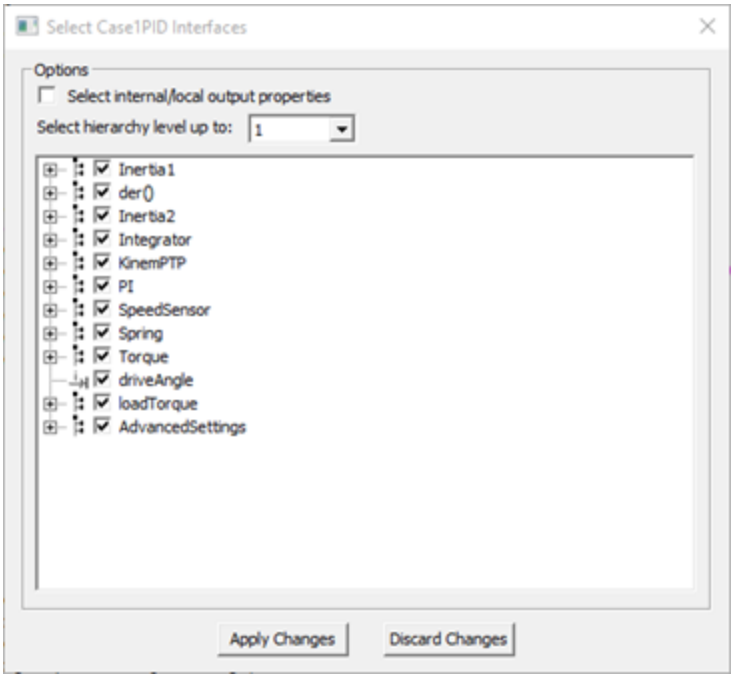
The **Overwrite model text** dialog box appears.



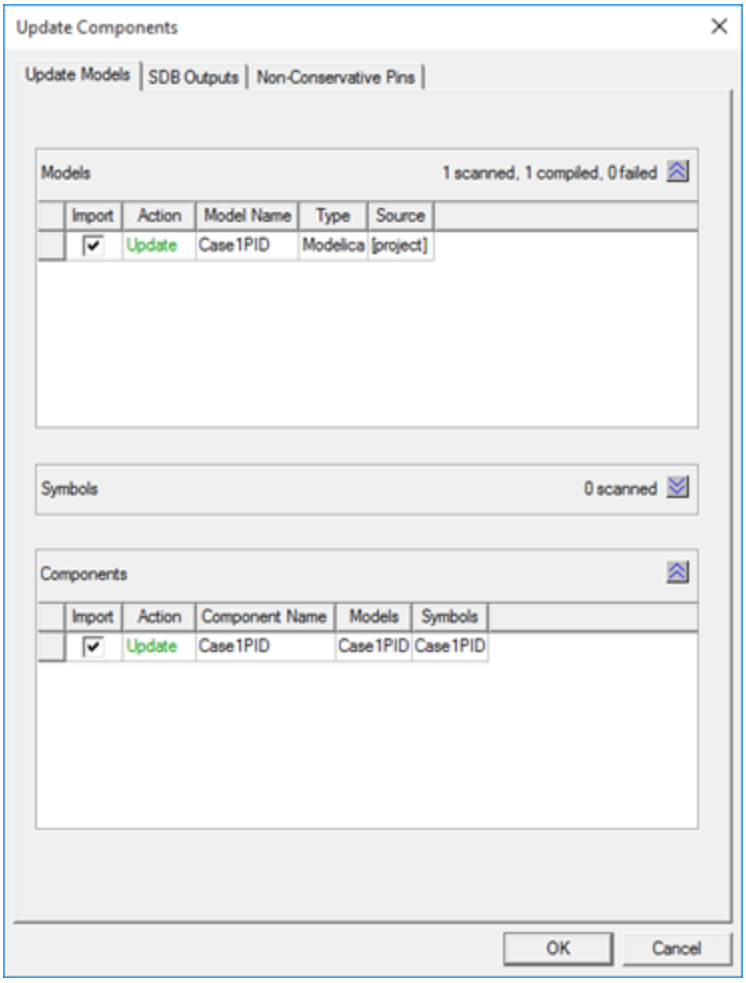
Note	This same dialog box appears when you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For the PID Controller example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. (This may take several seconds.)

3. Following the compile operation, you can select interfaces for the PID or just accept the default set of preselected interfaces.

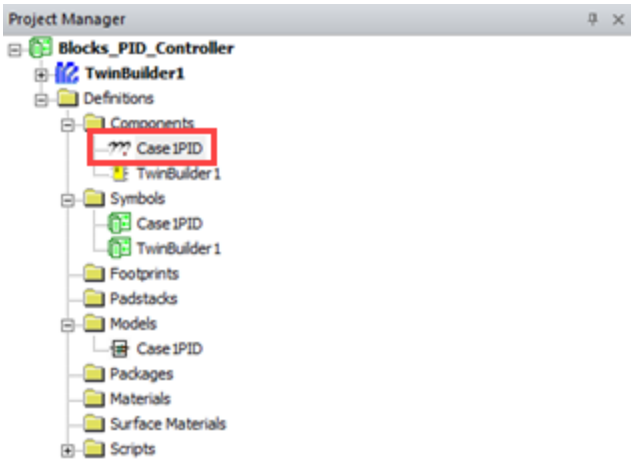


4. When finished selecting interfaces, click **Apply Changes** to close the dialog box and apply the changes to the PID Controller model. The **Update Components** dialog box appears with both the PID model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the PID Controller model and component into Twin Builder.

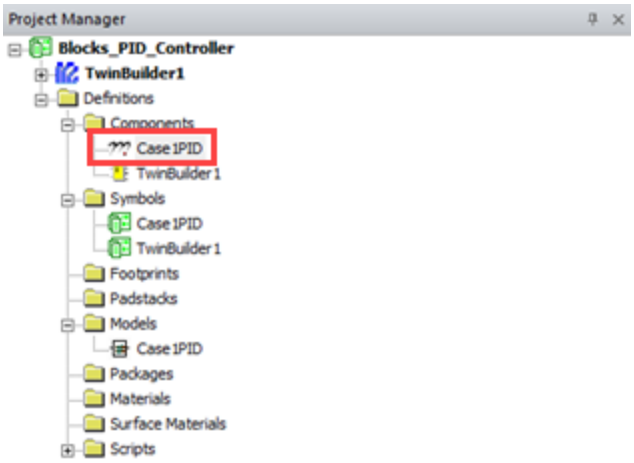
The Modelica PID Controller component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



Next, you will add the component to a Twin Builder schematic.

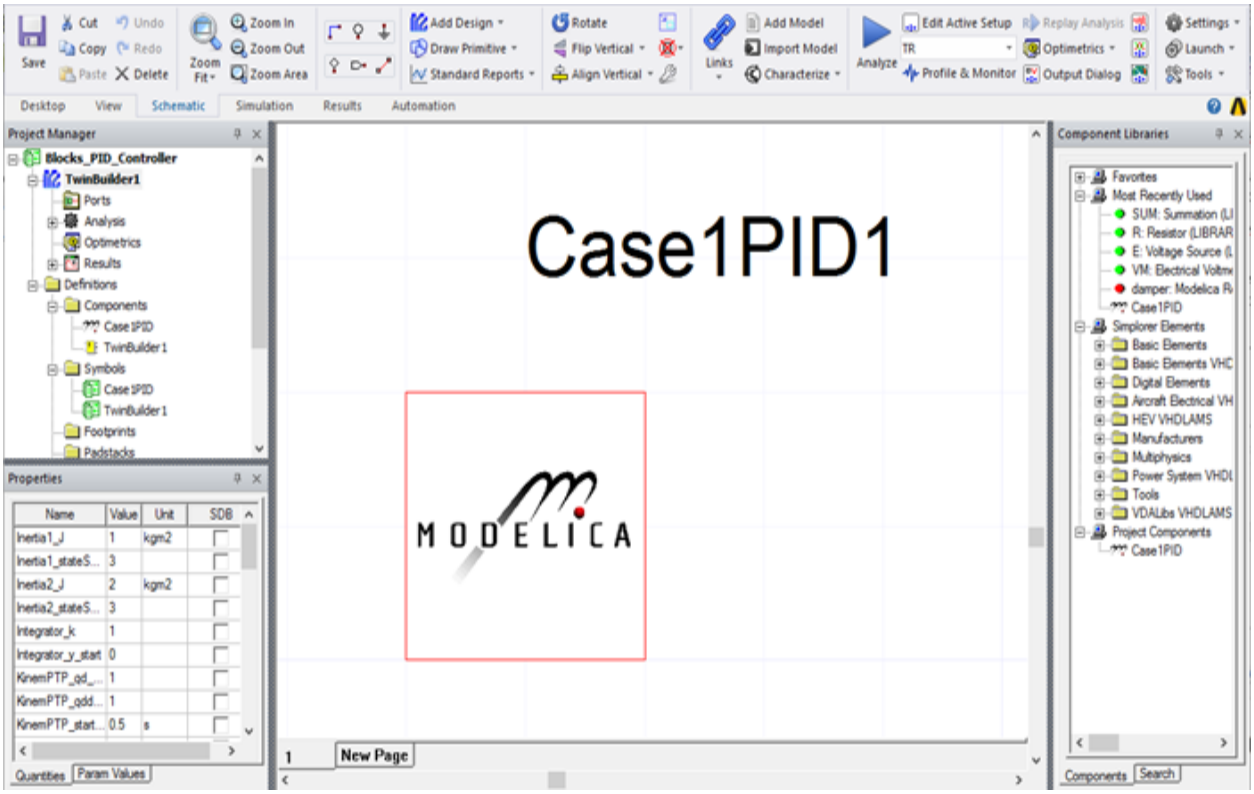
Adding the PID Controller Component to a Twin Builder Schematic

When Modelica PID Controller component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.

2. Locate the Modelica PID controller component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Because the entire PID controller system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

Next, you will define outputs and create reports for simulation results.

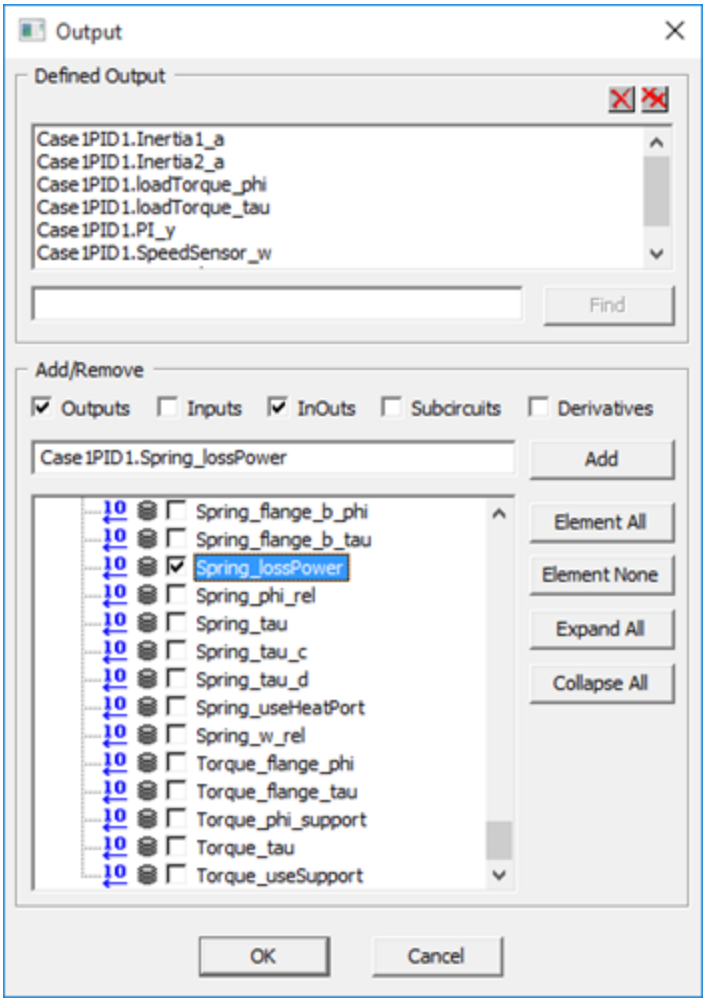
Defining Outputs and Creating Reports for the PID Controller

When Modelica PID Controller component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder.

1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, the following outputs are selected as shown below:

   - Case1PID1.Inertia1_a
   - Case1PID1.Inertia1_b
   - Case1PID1.loadTorque_phi
   - Case1PID1.loadTorque_tau
   - Case1PID1.PI_y
   - Case1PID1.SpeedSensor_w
   - Case1PID1.Spring_lossPower
   - Case1PID1.KinemPTP_y_1_

   Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.



2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

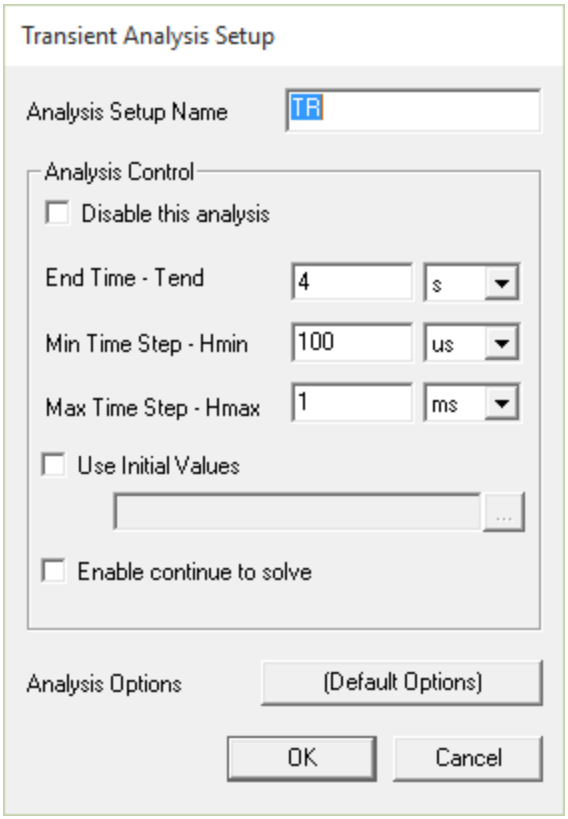
   The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate PID Controller simulation results.

Setting Up and Running a Transient Analysis for the PID Controller

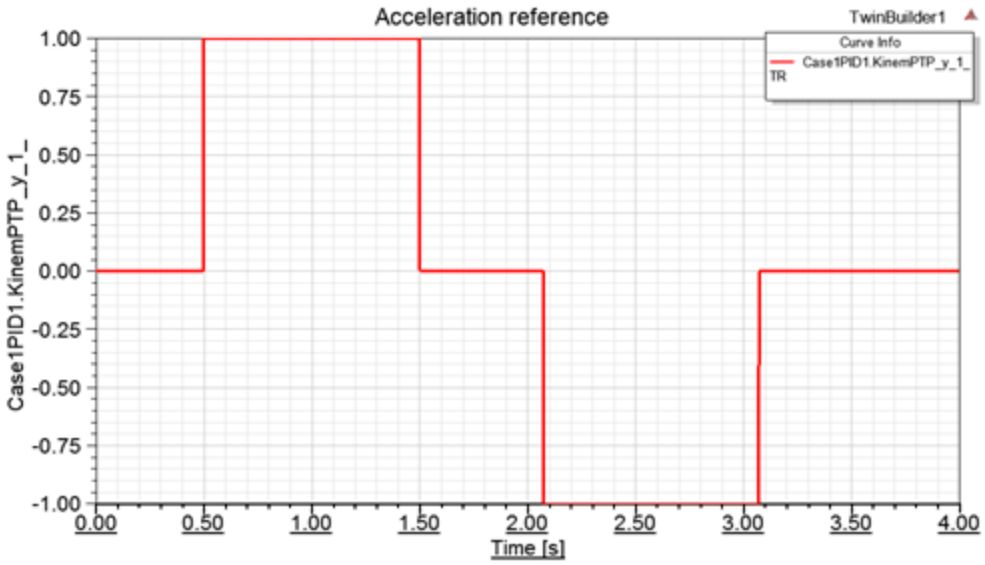
In this example, we would like to simulate the time-domain response of the PID Controller system. This requires that a transient analysis needs to be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

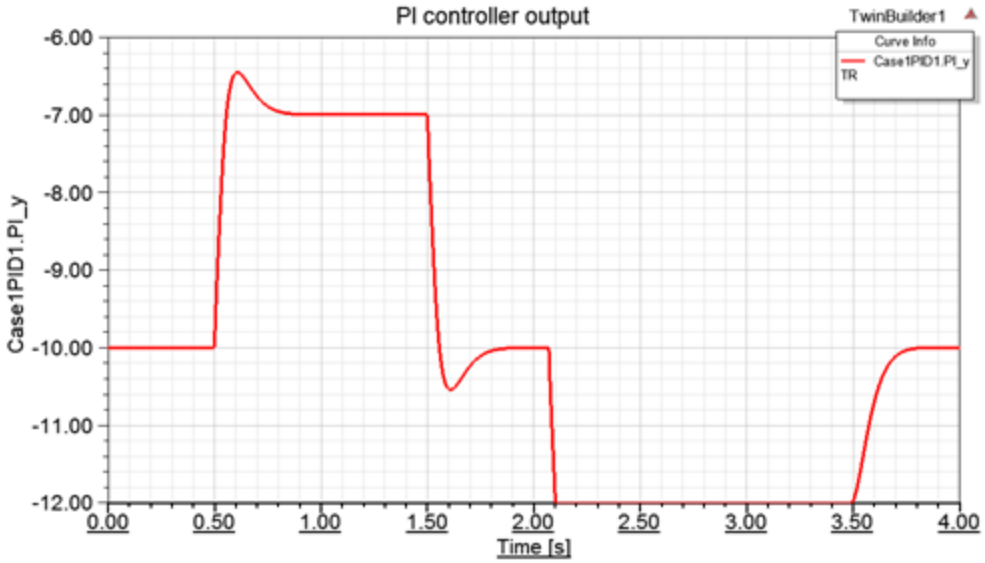
1. Double-click on **TR** to set up end time and the desired time step for the simulation.

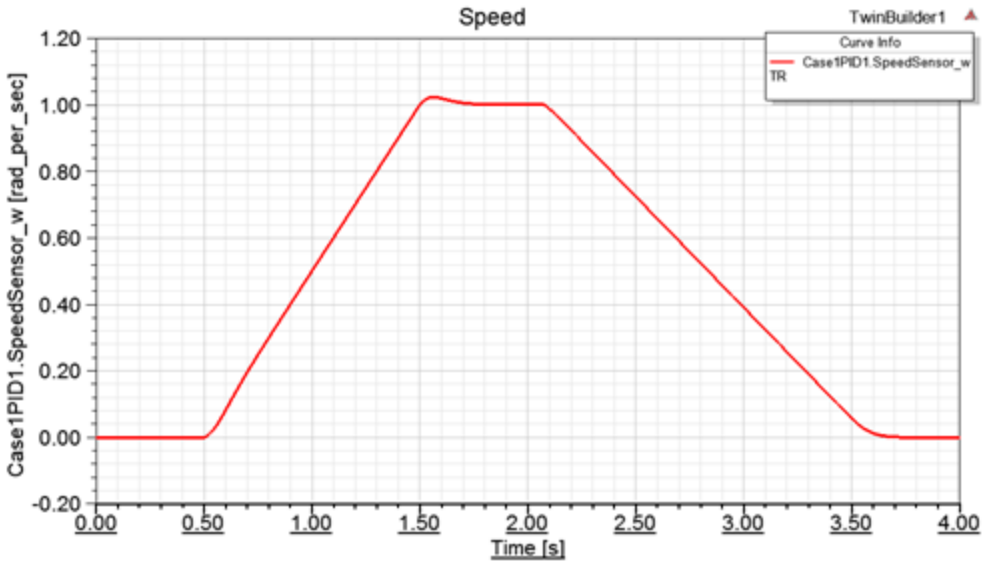


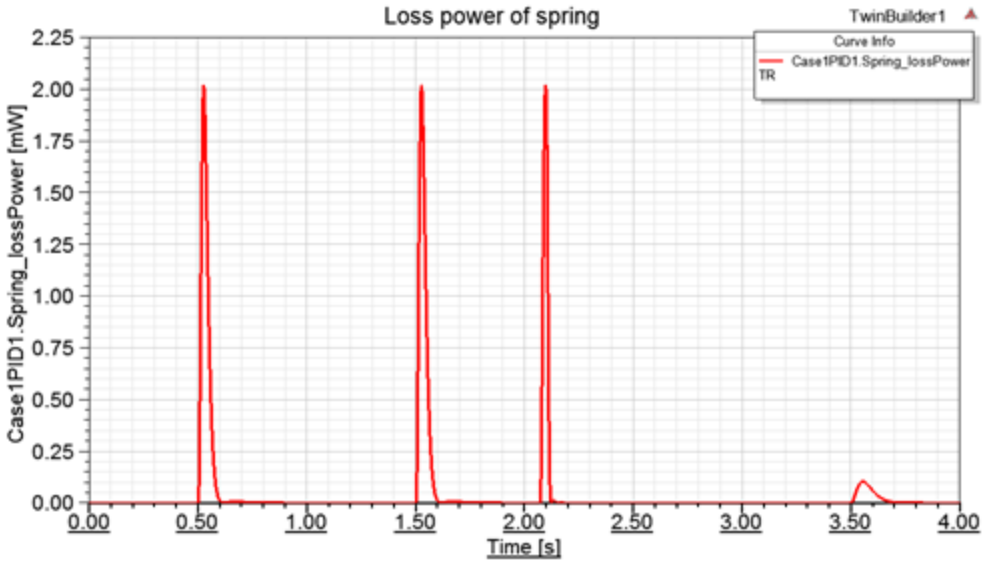
2. For this example, set **End Time** to 4s, **Min Time Step** to 100us, and keep Max Step Time as the default 1 ms. For more details about transient analysis, see *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
   - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
   - In the Project Manager, right click on **TR** of the design, then select **Analyze**.

Below are sample reports of the simulation results for the design.









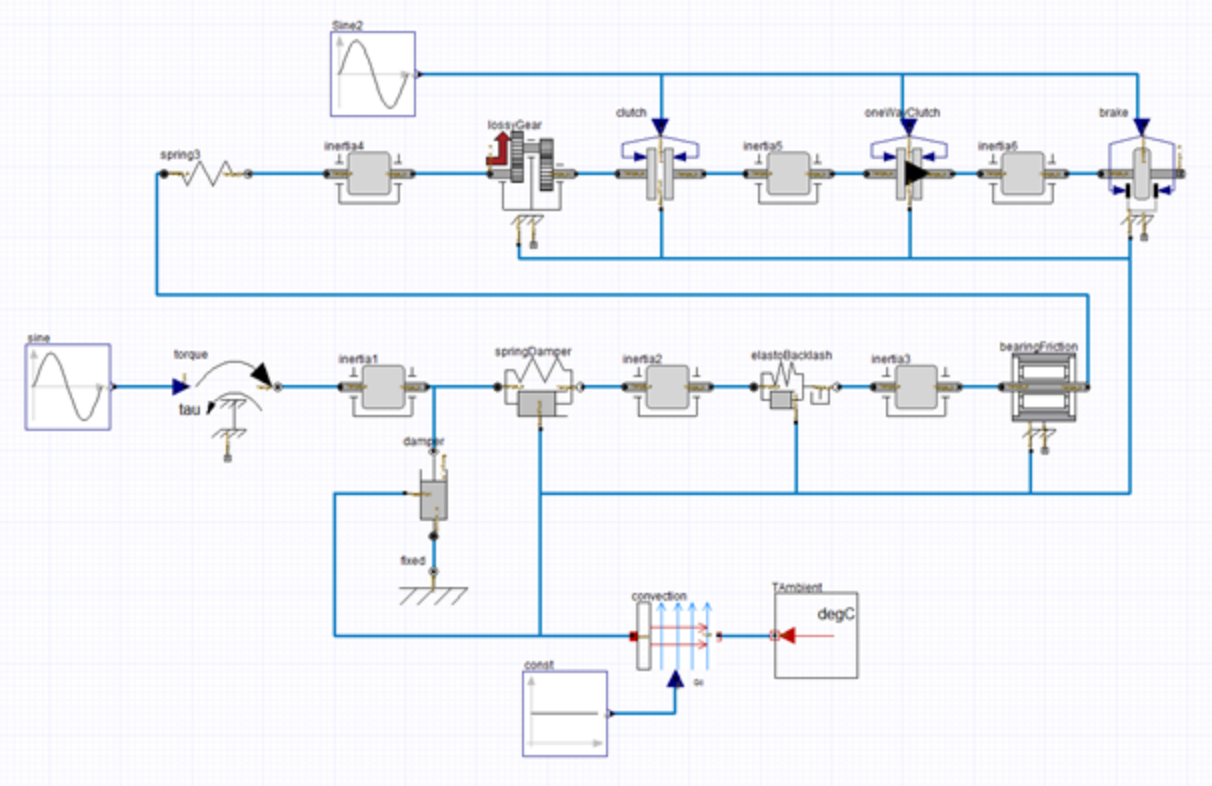
3 - Case Studies

The following examples describe the use of multiple Modelica model architectures for different levels of complexity, different modeling styles, and different behavior.

- Mechanics Example: Heat Losses
- Fluids Example: Controlled Tanks
- Thermal Example: Parallel Pump Drop Out
- Magnetic Example: SMPM Inverter MultiPhase
- Electrical Example: AIMC with Losses
- Simple Motor Power Train

Mechanics Example: Heat Losses

This model demonstrates how to model the dissipated power of a drive train. In this example, the heat Ports of all components are enabled and connected via a convection element to the environment. The total heat flow generated by the elements of the drive train, and transported to the environment, is present in variable **convection.fluid**.



This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\Heat_losses.aedt

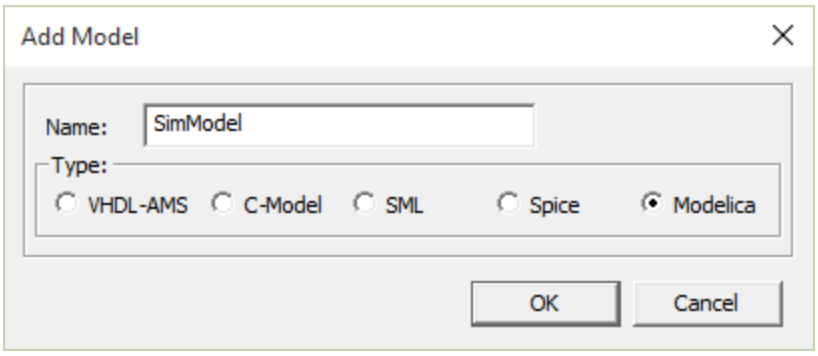
Heat Losses Example Workflow

In this example, we will create a Modelica heat loss model and analyze its performance in Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the Heat Losses model into the **Diagram Editor**.
- Connect the Modelica Heat Losses components.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Define outputs and Create Reports for the Heat Losses simulation results.
- Set up and run a Transient analysis for the Heat Losses design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.
2. Create a new Modelica model by doing one of the following:
   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

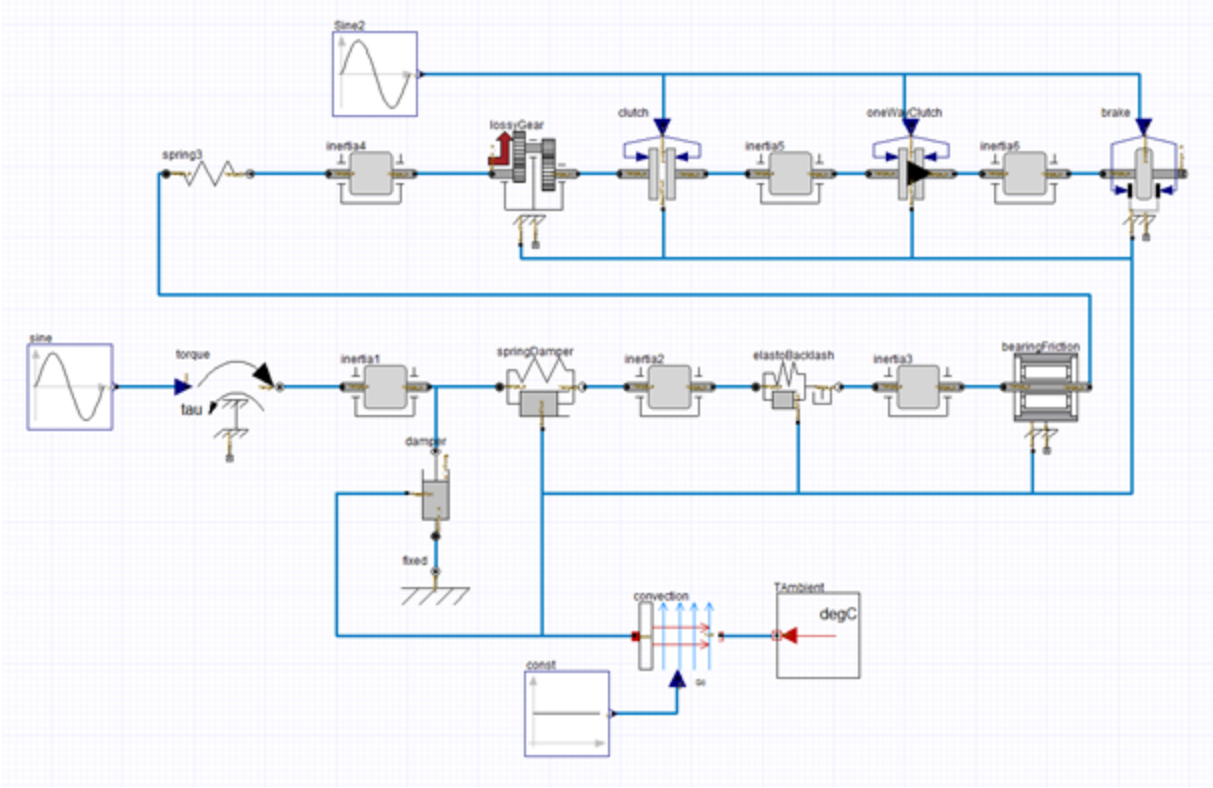
Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

Placing Modelica Components for the Heat Losses Model

All of the required components for the Heat Losses example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
sine, sine2	Modelica/Blocks/Sources/Sine
const	Modelica/Blocks/Sources/Constant
spring3	Modelica/Mechanics/Rotational/Components/Spring
inertia 1-6	Modelica/Mechanics/Rotational/Components/Inertia
torque	Modelica/Mechanics/Rotational/Sources/Torque
springDamper	Modelica/Mechanics/Rotational/Components/SpringDamper
elastoBacklash	Modelica/Mechanics/Rotational/Components/ElastoBacklash
bearingFriction	Modelica/Mechanics/Rotational/Components/BearingFriction
damper	Modelica/Mechanics/Rotational/Components/Damper
fixed	Modelica/Mechanics/Rotational/Components/Fixed
lossyGear	Modelica/Mechanics/Rotational/Components/LossyGear
clutch	Modelica/Mechanics/Rotational/Components/Clutch
oneWayClutch	Modelica/Mechanics/Rotational/Components/OneWayClutch
brake	Modelica/Mechanics/Rotational/Components/Brake
convection	Modelica/Thermal/HeatTransfer/Components/Convection
TAmbient	Modelica/Thermal/HeatTransfer/Celsius/FixedTemperature

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press Esc after placing each component to stop the placement action. (Connections will be added later.)

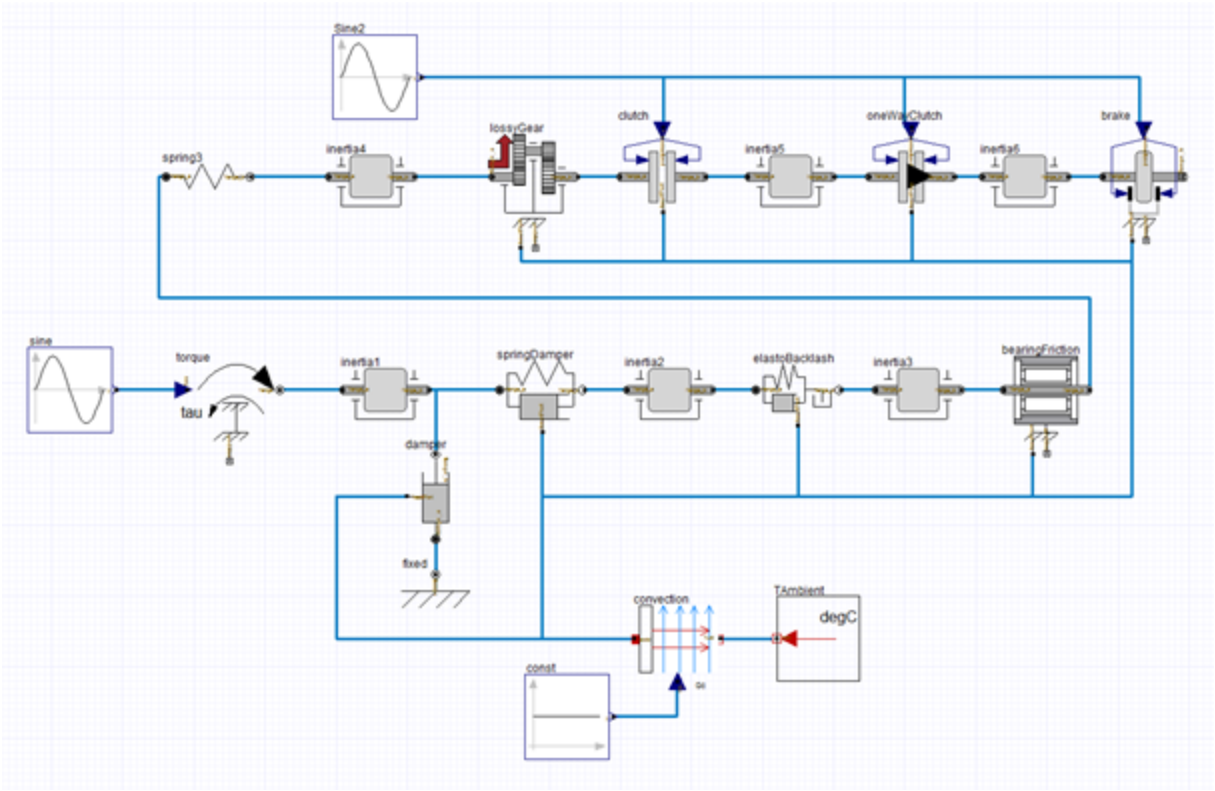


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press Esc to complete each action.

Next, you will add connections between the various Heat Losses components.

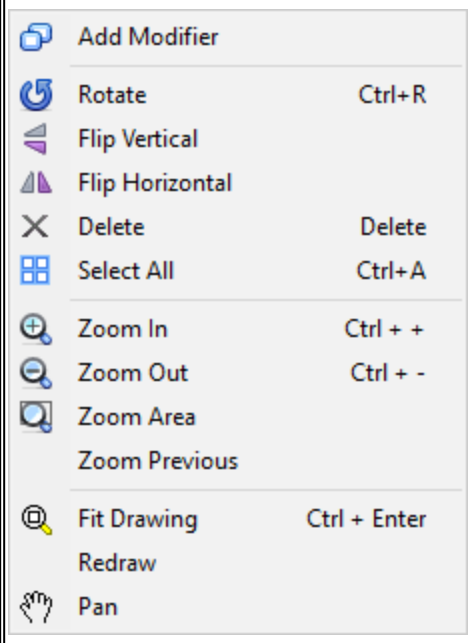
Connecting Modelica Heat Losses Components

To connect the components for the Heat Losses example as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will set up model properties for the Heat Losses example.

Setting Up Model Properties for the Heat Losses Example

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
sine	amplitude = 20 freqHz = 5
sine2	amplitude = 1 freqHz = 0.2 phase = 0 offset = 0 startTime = 0
const	k = 20
torque	
convection	
fixed	
TAmbient	T = 25
inertia 1	J = 2

Component Instance	Properties
	phi(start = 0) w(start = 0)
inertia 2 - 4	J = 2 phi(fixed = true,start = 0) w(fixed = true,start = 0)
inertia 5 6	J = 2
damper	d = 10 phi_nominal = 1e-4 useHeatPort = true
springDamper	c = 1e4 d = 20 phi_rel0 = 0 phi_nominal = 1e-4 useHeatPort = true phi_rel(start = 0) w_rel(start = 0) a_rel(start = 0)
elastoBacklash	c = 1e5 d = 100 b = 0.001 phi_rel0 = 0 useHeatPort = true phi_nominal = 1e-4 w_rel(start = 0)
bearingFriction	tau_pos = [0, 1] peak = 1 useHeatPort = true w_small = 1e10

Component Instance	Properties
spring3	c = 1e4 phi_rel0 = 0
lossyGear	ratio = 2 lossTable = [0, 0.8, 0.8, 1, 1; 1, 0.7, 0.7, 2, 2] useHeatPort = true Unknown = 3 Free = 2 Forward = 1 Stuck = 0 Backward = -1
clutch	mue_pos = [0, 0.5] peak = 1 cgeo =1 fn_max = 10 phi_nominal = 1e-4 useHeatPort = true w_small = 1e10 phi_rel(fixed = true) w_rel(fixed = true)
oneWayClutch	mue_pos = [0, 0.5] peak = 1 cgeo = 1 fn_max = 1 useHeatPort = true phi_nominal = 1e-4 w_small = 1e10 phi_rel(fixed = true) w_rel(fixed = true,start = 0)

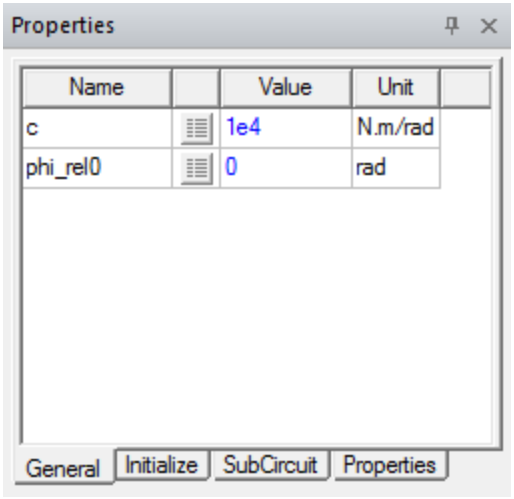
Component Instance	Properties
brake	mue_pos = [0, 0.5] peak = 1 cgeo = 1 fn_max = 2 w_small = 1e10 useHeatPort = true

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options as needed.

For example, to set properties for the **Spring3** component in the Diagram Editor:

1. Click on the **Spring3** component to display its Properties window.



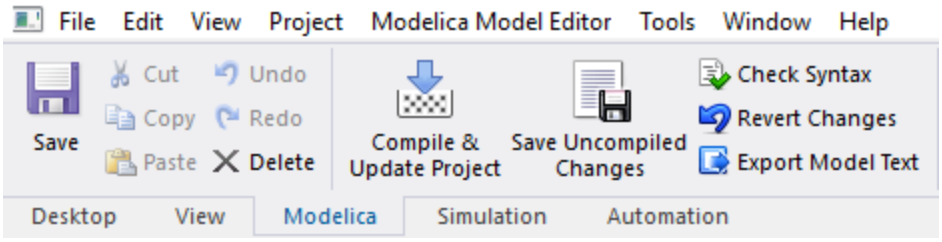
2. On the **General** tab, per the information in the above table, enter **1e4** for the **c**property; and **0** for **phi_rel0**. No changes are needed on the other tabs. The Properties tab allows you to set the name of the component instance.

Next, you will compile and update the project.

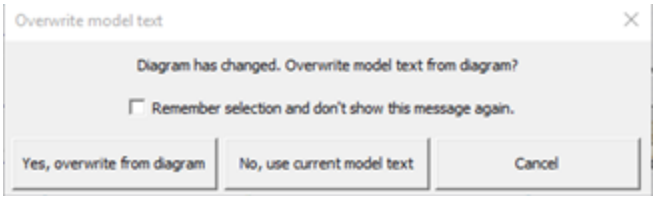
Compiling and Updating the Heat Losses Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.

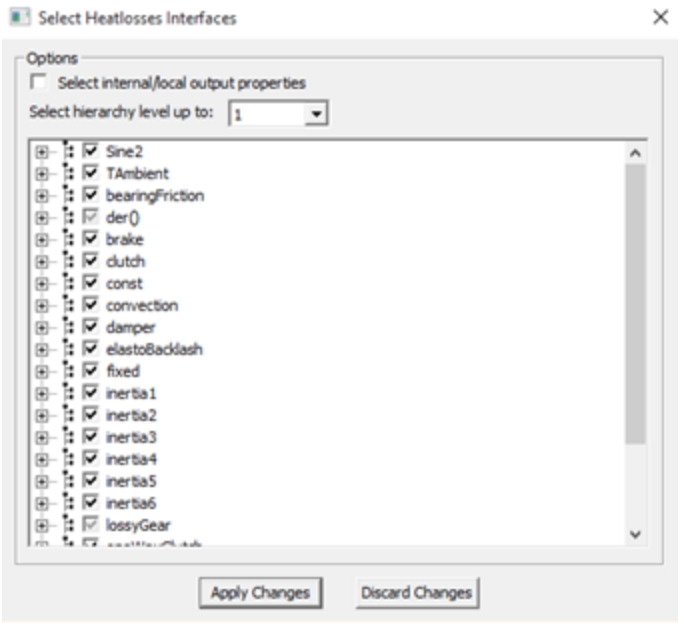


The **Overwrite model text** dialog is displayed.

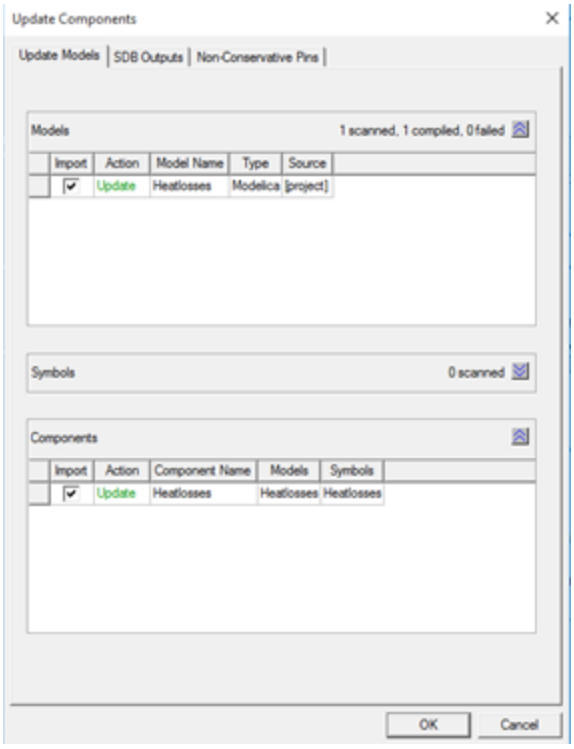


Note	This same dialog is also displayed whenever you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For Heat Losses example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. (This may take several seconds.)
3. Following the compile operation, you can select interfaces for the Heat Losses model, or just accept the default set of preselected interfaces.

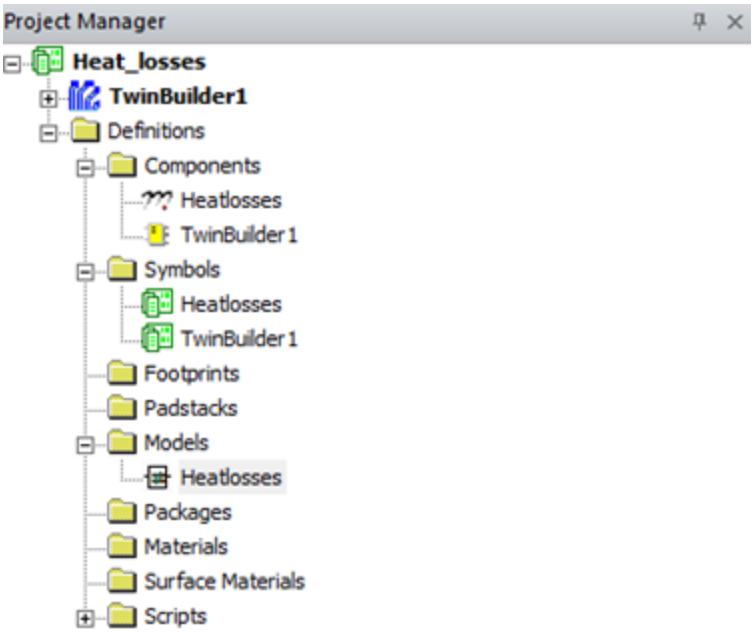


4. When finished selecting interfaces, click **Apply Changes** to close the dialog and apply the changes to the Heat Losses model. The **Update Components** dialog is displayed with both the Heat Losses model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the Heat Losses model and component into Twin Builder.

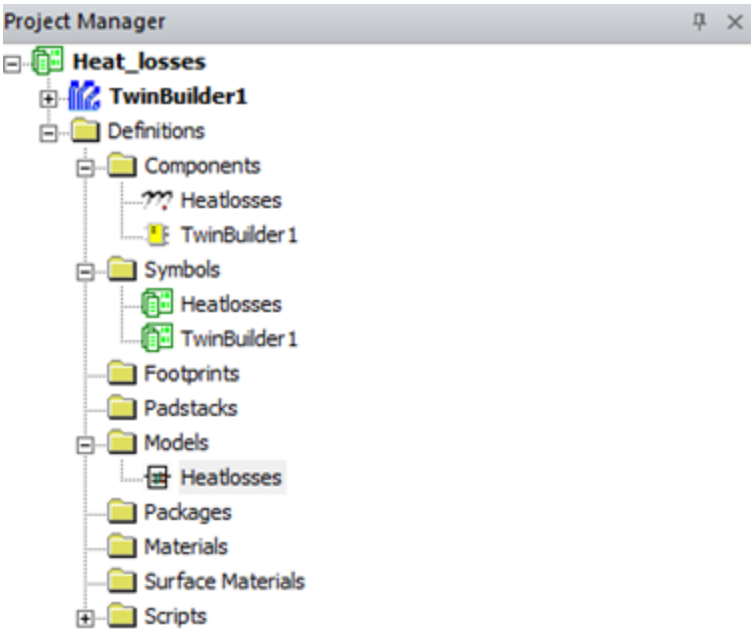
The Modelica Heat Losses component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



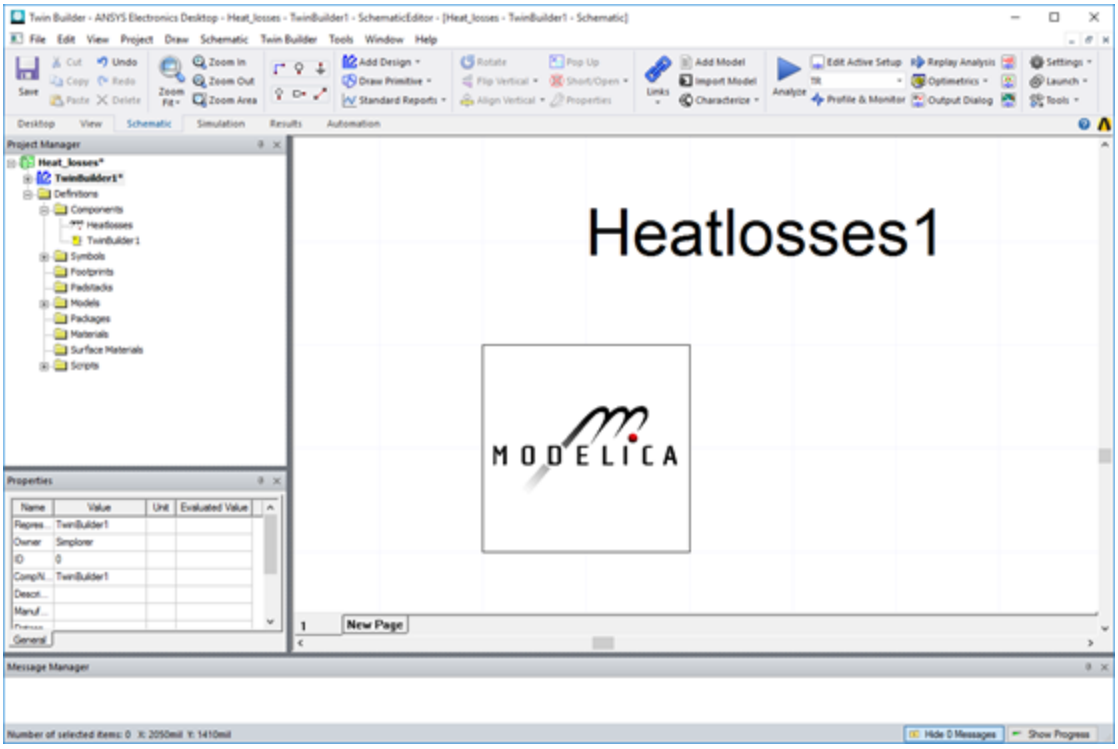
Next, you will add the component to a Twin Builder schematic.

Adding the Heat Losses Component to a Twin Builder Schematic

When Modelica Heat Losses component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.
2. Locate the Modelica Heatlosses component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

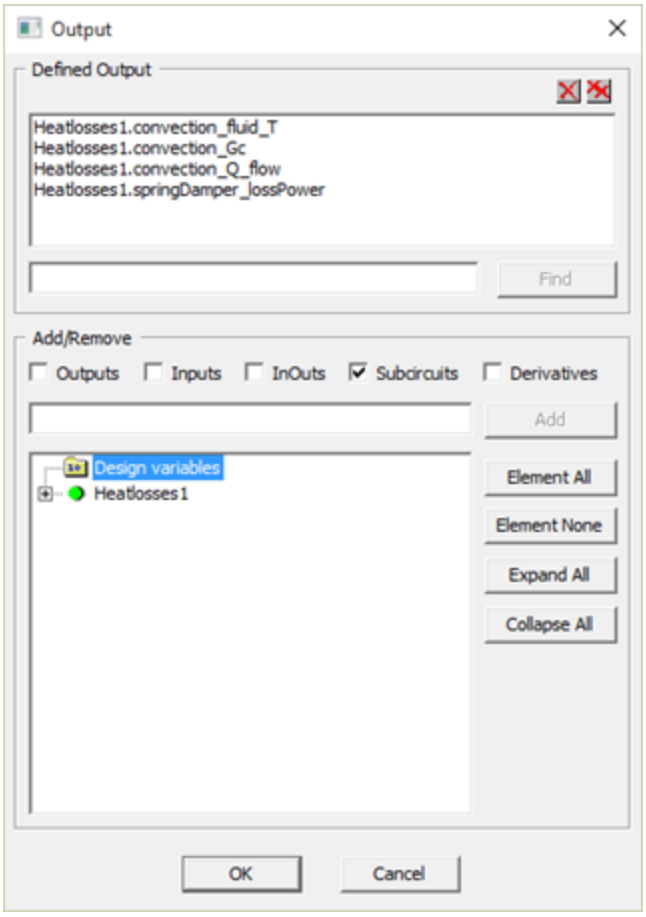
Because the entire PID controller system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

Next, you will define outputs and create reports for simulation results.

Defining Outputs and Creating Reports for the Heat Losses Design

When Modelica PID Controller component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.

1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.



2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

   The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate PID Controller simulation results.

Setting Up and Running a Transient Analysis for the Heat Losses Design

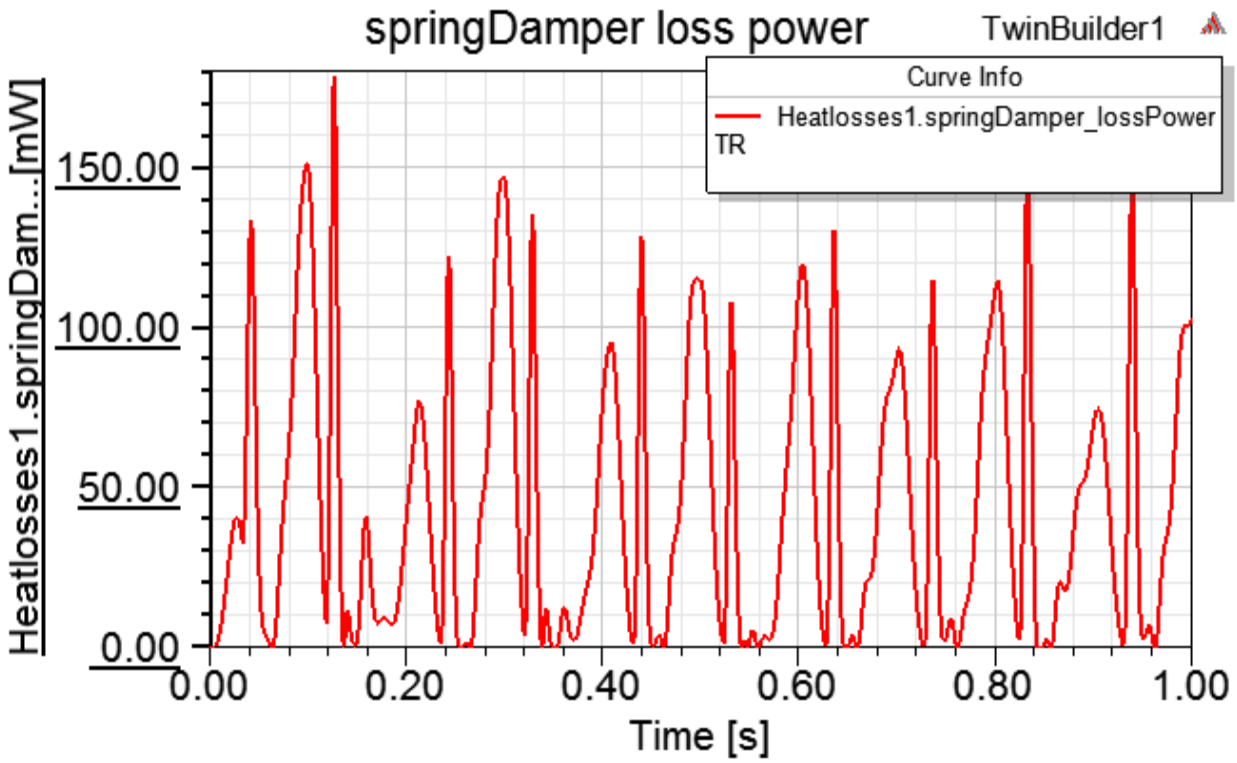
In this example, we would like to simulate the time-domain response of the Heat Losses system. This requires that a transient analysis needs to be performed on the design. By default, a

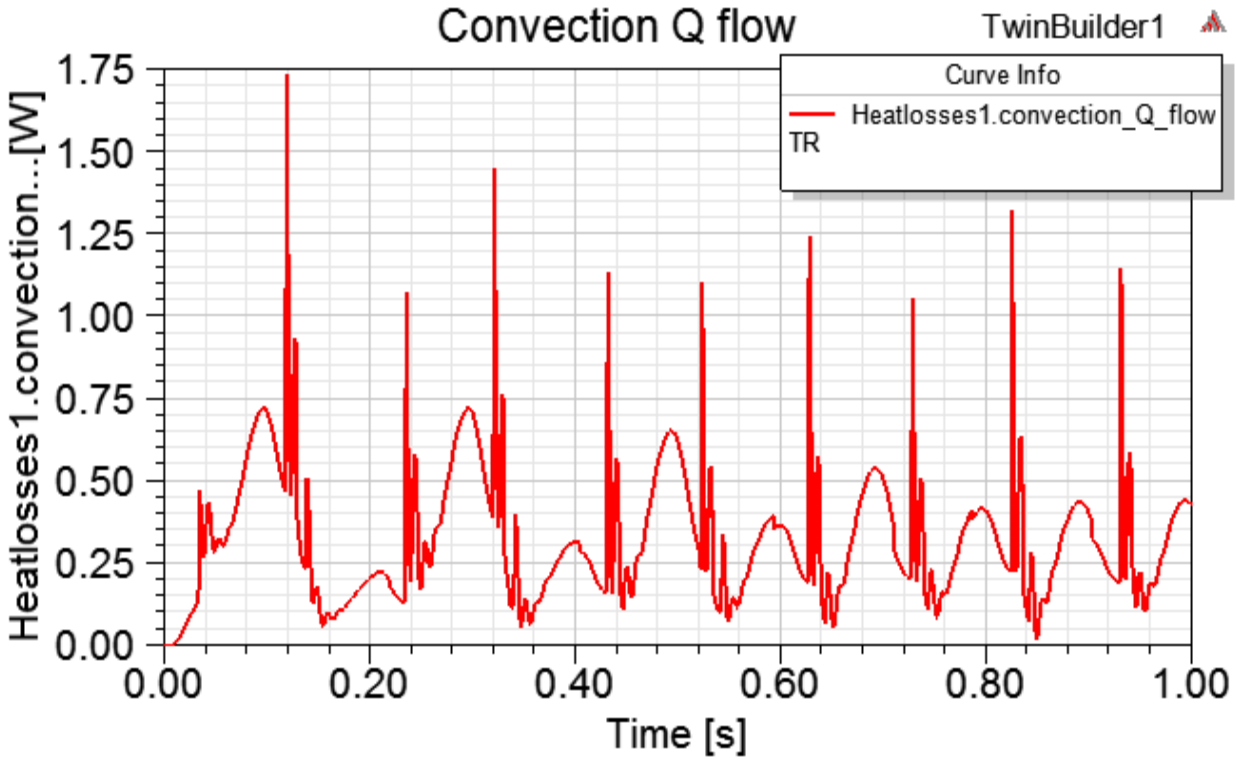
transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

1. Double-click on **TR** to set up end time and the desired time step for the simulation.

2. For this example, set **End Time** to 1s, **Min Time Step** to 10us, and keep Max Step Time as the default 1 ms. For more details about transient analysis, please refer to *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
    - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
    - In the Project Manager, right click on **TR** of the design, then select **Analyze**.

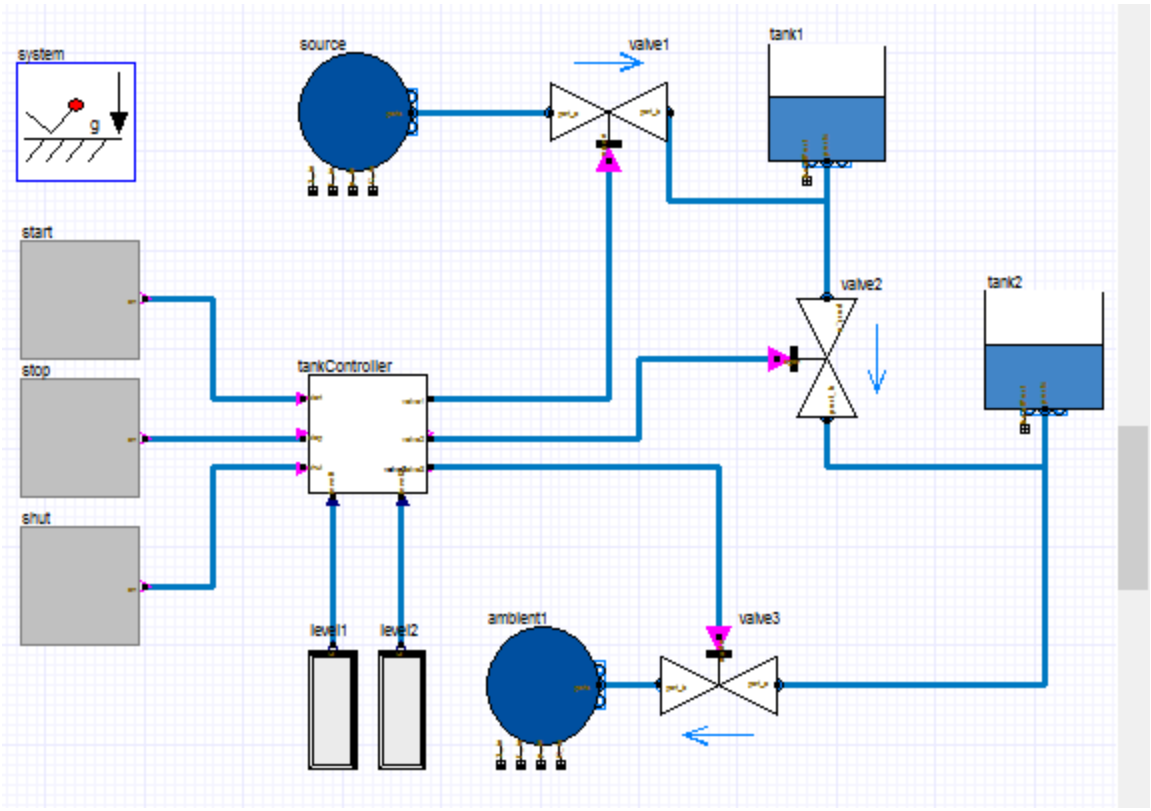
Below are sample reports of the simulation results.





Fluids Example: Controlled Tanks

This example demonstrates a tank filling/emptying system with a controller as shown below.



In this case, the two tanks are filled and emptied by following scheme:

1. Valve 1 is opened and tank 1 is filled.
2. When tank 1 reaches its fill level limit, valve 1 is closed.
3. After a waiting time, valve 2 is opened and the fluid flows from tank 1 into tank 2.
4. When tank 1 reaches its minimum level, valve 2 is closed.
5. After a waiting time, valve 3 is opened and the fluid flows out of tank 2.
6. When tank 2 reaches its minimum level, valve 3 is closed.

The above process can be influenced by three buttons:

- Button “start” starts the above process. When this button is pressed after a "stop" or "shut" operation, the process operation continues.

- Button “stop” stops the above process by closing all valves. Then, the controller waits for further input (either "start" or "shut" operation).
- 3)Button “shut” is used to shut down the process, by emptying at once both tanks by opening valve 2 and valve 3. When this is achieved, the process goes back to its start configuration where all 3 valves are closed. Clicking on "start", restarts the process.

The example uses the following button settings:

- Button start pressed at 20 s;
- Button stop pressed at 220 s;
- Button start pressed at 280 s;
- Button stop pressed at 650 s;
- Button shut pressed at 700 s;
- Simulate for 900 s.

This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\ControlledTanks.aedt

Controlled Tanks Example Workflow

In this example, we will create a Modelica controlled tanks model and analyze its performance in Twin Builder environment. The following steps outline the process for accomplishing these goals.

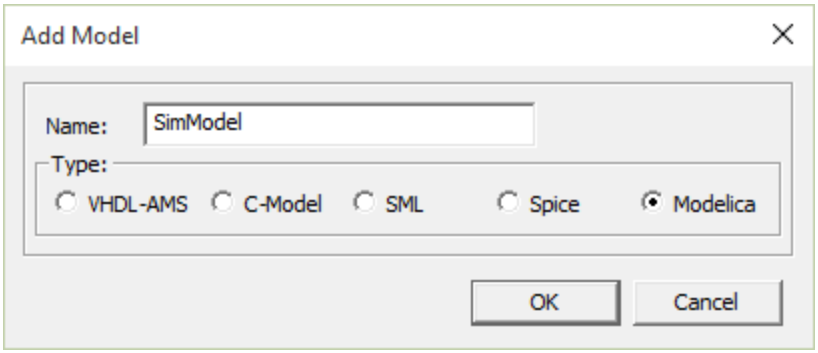
- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the Controlled Tanks model into the **Diagram Editor**.
- Connect the Modelica Controlled Tanks components.
- Declare a fluid medium for the Controlled Tanks model.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Define outputs and Create Reports for the Controlled Tanks simulation results.
- Set up and run a Transient analysis for the Controlled Tanks design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.

2. Create a new Modelica model by doing one of the following:

   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click on the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

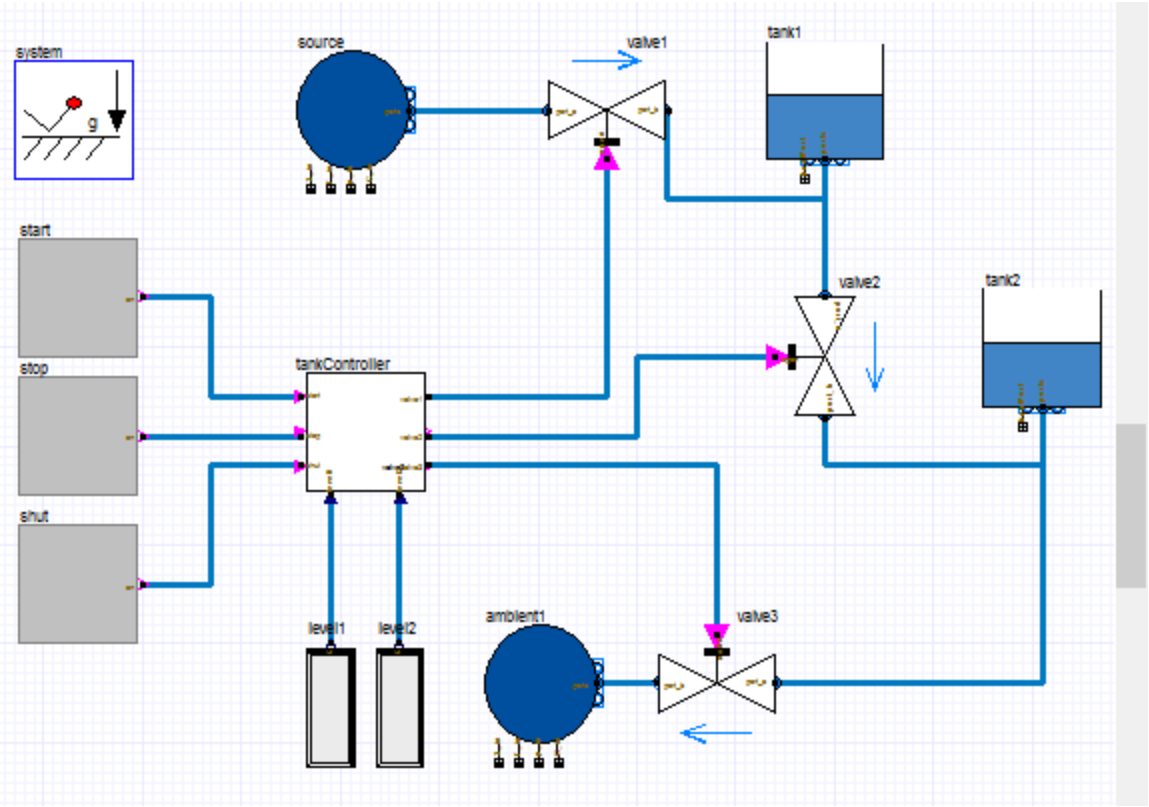
Placing Modelica Components for the Controlled Tanks Model

All of the required components for the Controlled Tanks example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
source	Modelica/Fluid/Sources/Boundary_pT
tank1, tank2	Modelica/Fluid/Vessels/OpenTank
valve1, valve2, valve3	Modelica/Fluid/Valve/ValveDiscrete
tankController	Modelica/Fluid/Examples/ControlledTankSystem/Utilities/TankController
ambient1	Modelica/Fluid/Sources/Boundary_pT
system	Modelica/Fluid/System
start	Modelica/Fluid/Examples/ControlledTankSystem/Utilities/RadioButton
stop	Modelica/Fluid/Examples/ControlledTankSystem/Utilities/RadioButton
shut	Modelica/Fluid/Examples/ControlledTankSystem/Utilities/RadioButton

Component Instance	Location in Library
level1, level2	Modelica/Blocks/Sources/RealExpression

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press **Esc** after placing each component to stop the placement action. Connections will be added later.

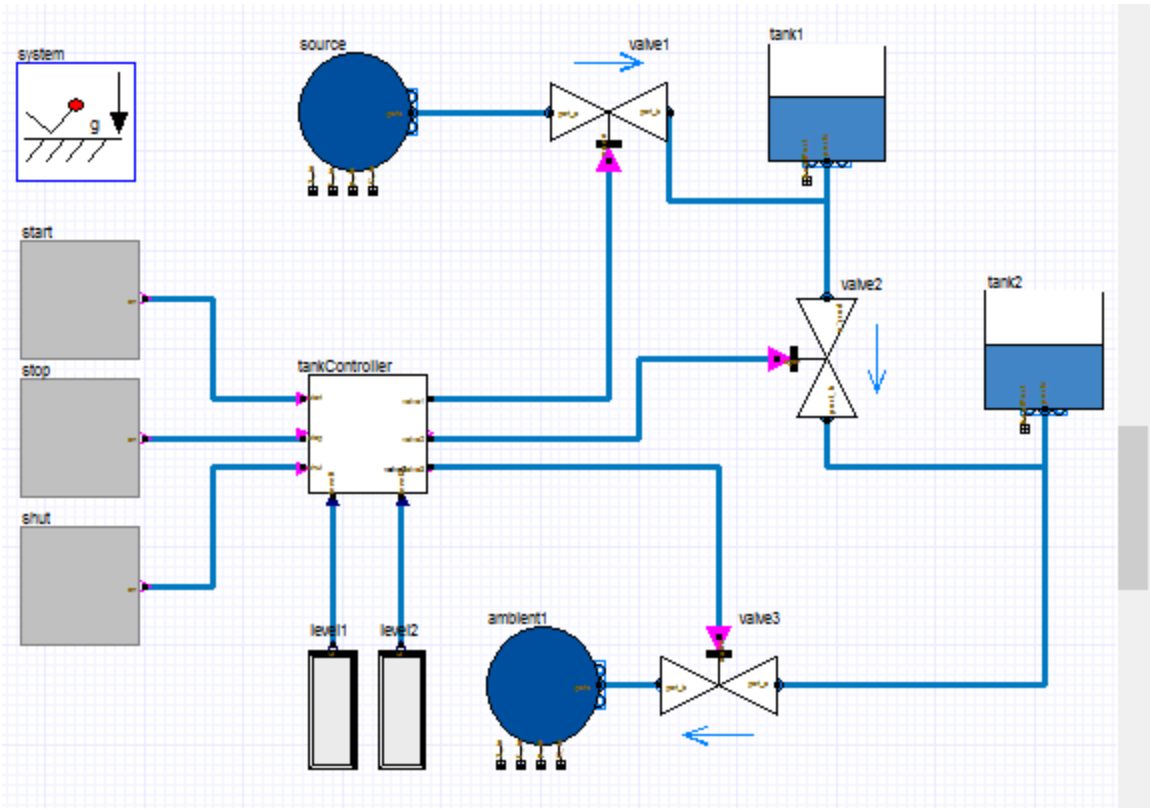


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab.
4. Press **Esc** to complete each action.

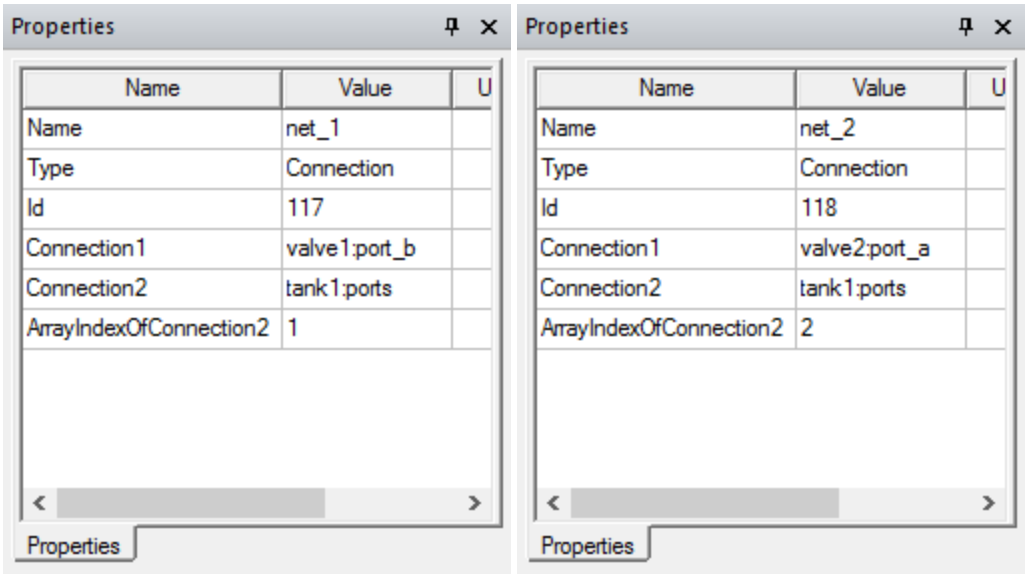
Next, you will add connections between the various Controlled Tanks components.

Connecting Modelica Controlled Tanks Components

To connect the components for the Controlled Tanks example as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.
3. An additional step is required to set the vector port indexes for tank1 and tank2.
    a. For tank1, we want to have tank1.ports[1] connect to valve1.port_b, and tank1.ports [2] connect to valve2.port_a. To specify the connection in diagram editor, click on connection line between **valve1** and **tank1**, and set **ArrayIndexOfConnection2** to **1** in the Properties window. Similarly, click on connection line between **valve2** and **tank1**, and set **ArrayIndexOfConnection2** to **2**.

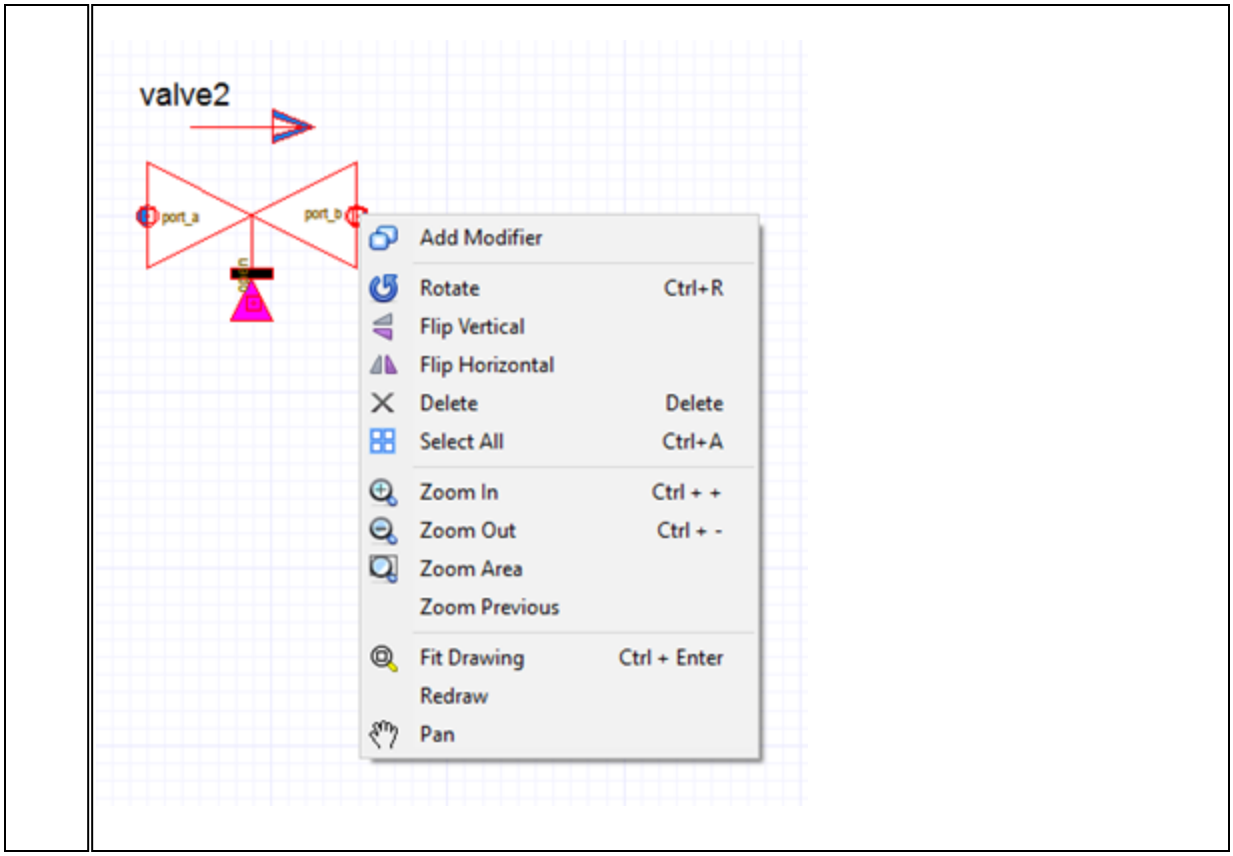


b. Similarly, for the connections between **valve2** and **tank2**, set **ArrayIndexOfConnection2** to **1**; and between **valve3** and **tank2**, set **ArrayIndexOfConnection2** to **2**.

   The above valve-to-tank settings generate the following lines in the Modelica model text code:

```
connect(valve1.port_b, tank1.ports[1]);

connect(valve2.port_a, tank1.ports[2]);

connect(valve2.port_b, tank2.ports[1]);

connect(valve3.port_a, tank2.ports[2]);
```

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



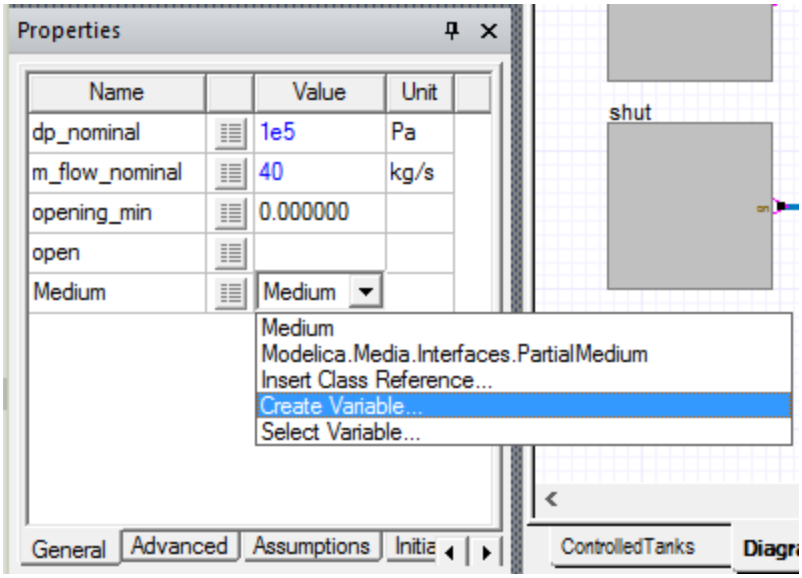
Next, you will add a declaration of a fluid medium variable for the model.

Declaring a Fluid Medium Variable for the Controlled Tanks Model

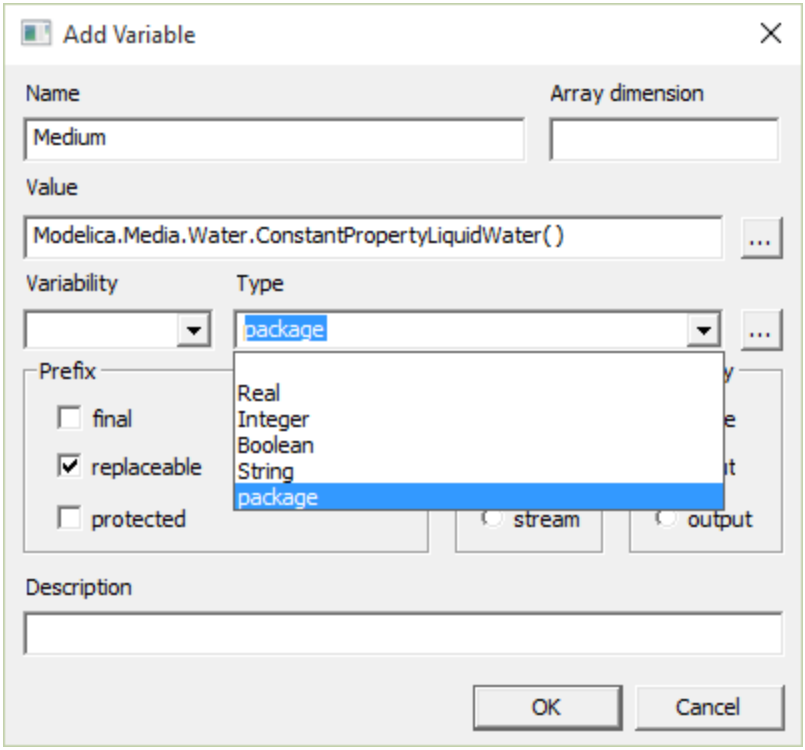
When modeling a fluid system in Modelica, the fluid medium is an important property to set for all models in the system. Models from the **Modelica.Fluid** library are implemented to use a medium defined in the **Modelica.Media** library, so you can declare a **Medium** package type variable from the **Modelica.Media** library, then reuse it for all **Medium** property settings.

For the Controlled Tanks example, the medium is water. Seven models in the example have a **Medium** property.

1. To declare the **Medium** to be *water*, select any one of the Controlled Tanks models that have the **Medium** property, for example, **valve1**.
2. In the **Properties** dialog box **General** tab, select **Create Variable** from the dropdown list in the **Value** column.



The **Add Variable** dialog box appears.



3. Click the **[...]** button for the **Value** field and select **ConstantPropertyLiquidWater** in the **Modelica** > **Media** > **Water** library. Select **package** in the **Type** drop-down menu.
4. Click **OK**. The declaration of package **Medium** is created in Modelica model text:

```
//Declaration(s)
 package Medium =
Modelica.Media.Water.ConstantPropertyLiquidWater( );
```

   Now that **Medium** is defined, it is available for selection in the remaining six models in the Controlled Tanks model that have the **Medium** property by choosing **Select Variable**, then choosing **Medium** in the **Select Variable** dialog box.

5. To change the fluid medium in the system, click **Modelica** > **Variables** on the ribbon. The **Edit Model** dialog box appears.
6. Click **Model** > **Variable** in the directory tree. Select the declared **Medium** value and edit as desired.

Next, you will set up properties for the Controlled Tanks model.

Setting Up Model Properties for the Controlled Tanks Example

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
source	p = 25*101325 nPorts = 1 redeclare package Medium = Medium
tank 1	height = 4 crossArea = 6 level_start = 0.05 Medium = Medium nPorts = 2 Re_turbulent = 100.000000 portsData = {Modelica.Fluid.Vessels.BaseClasses.VesselPortsData (diameter=0.2, height=4, zeta_out=0, zeta_ in=1),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData (diameter=0.2, height=0, zeta_out=0, zeta_in=1)} **Note**: **portsData** is an array of records that defines diameter and port level of the bottom and side connectors.

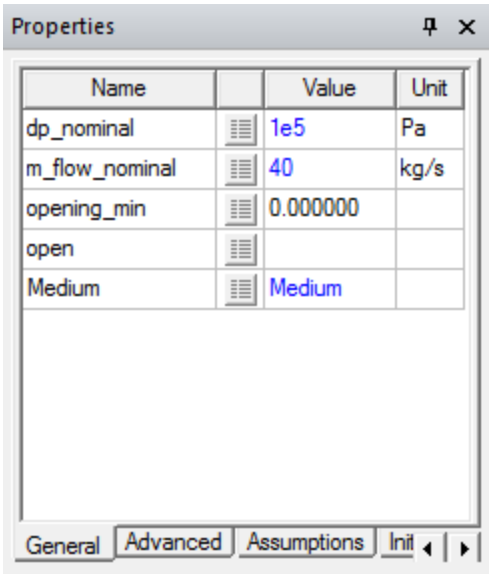
Component Instance	Properties
tank 2	height = 5 crossArea = 6 level_start = 0.05 Medium = Medium nPorts = 2 Re_turbulent = 100.000000 portsData = {Modelica.Fluid.Vessels.BaseClasses.VesselPortsData (diameter=0.2, height=5, zeta_out=0, zeta_ in=1),Modelica.Fluid.Vessels.BaseClasses.VesselPortsData (diameter=0.2, height=0, zeta_out=0,zeta_in=1)} **Note**: **portsData** is an array of records that defines diameter and port level of the bottom and side connectors.
valve 1	dp_noninal = 1e5 m_flow_nominal = 40 Medium = Medium
valve 2	dp_noninal = 1 m_flow_nominal = 100 Medium = Medium
valve 3	dp_noninal = 1 m_flow_nominal = 10 Medium = Medium
tankController	maxLevel = 0.9*4 minLevel = 0.01 waitTime = 50
ambient1	nPorts = 1 Medium = Medium
system	energyDynamics = Modelica.Fluid.Types.Dynamics.FixedInitial
start	button Time Table = {20, 280} reset = {stop.on, shut.on}
stop	button Time Table = {220, 650}

Component Instance	Properties
	reset = {start.on, shut.on}
shut	button Time Table = {700} reset = {start.on, stop.on}
level 1	y = tank1.level
level 2	y = tank2.level

To display the editable parameters and properties of a component, click the model to open its **Properties** dialog box.

For example, to set properties for the **valve1** component in the Diagram Editor:

1. Click on the **valve1** component to open its **Properties** dialog box.



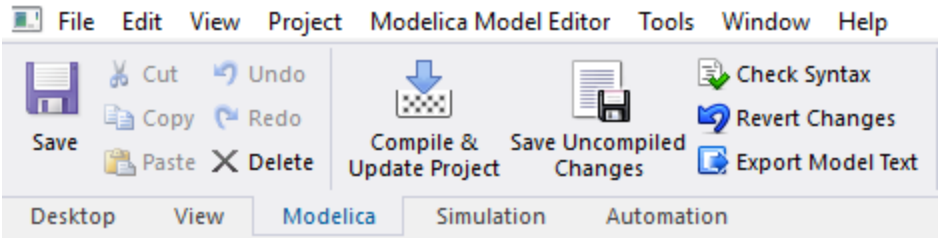
2. Click the **General** tab. Using the information in the above table, enter **1e5** for the **dp_nominal**property and **40** for **m_flow_nominal**. To edit the **Medium** property, choose **Select Variable** from the drop-down list. The **Select Variable** dialog opens. Select **Medium** for the Medium property. No changes are needed on the other tabs. Note that you can set the name of the component instance in the **Properties** tab allows.

Next, you will compile and update the project.

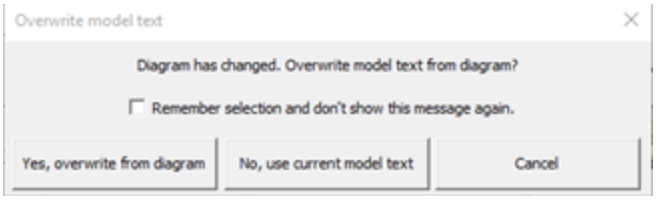
Compiling and Updating the Controlled Tanks Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.

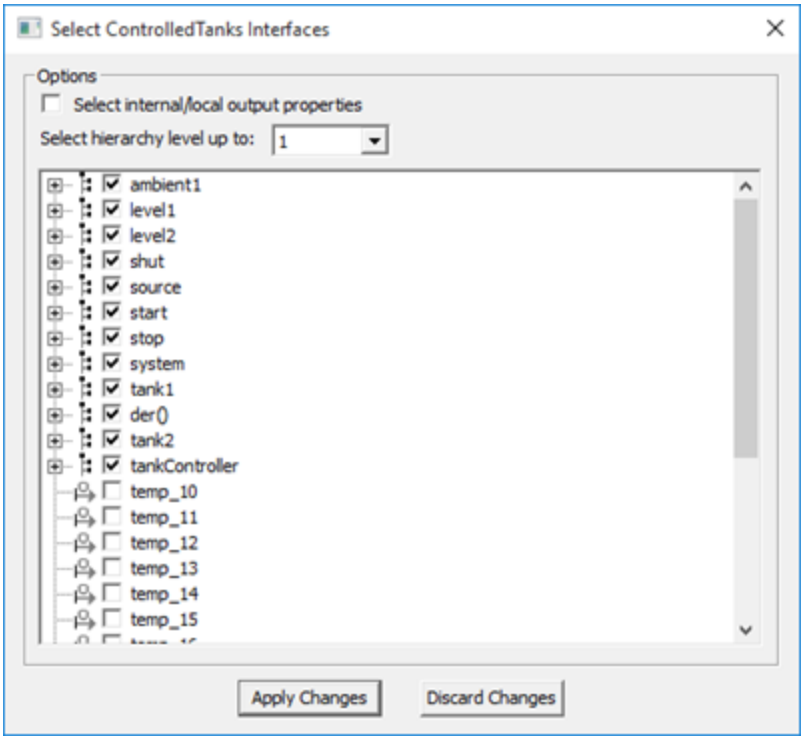


The **Overwrite model text** dialog box appears.

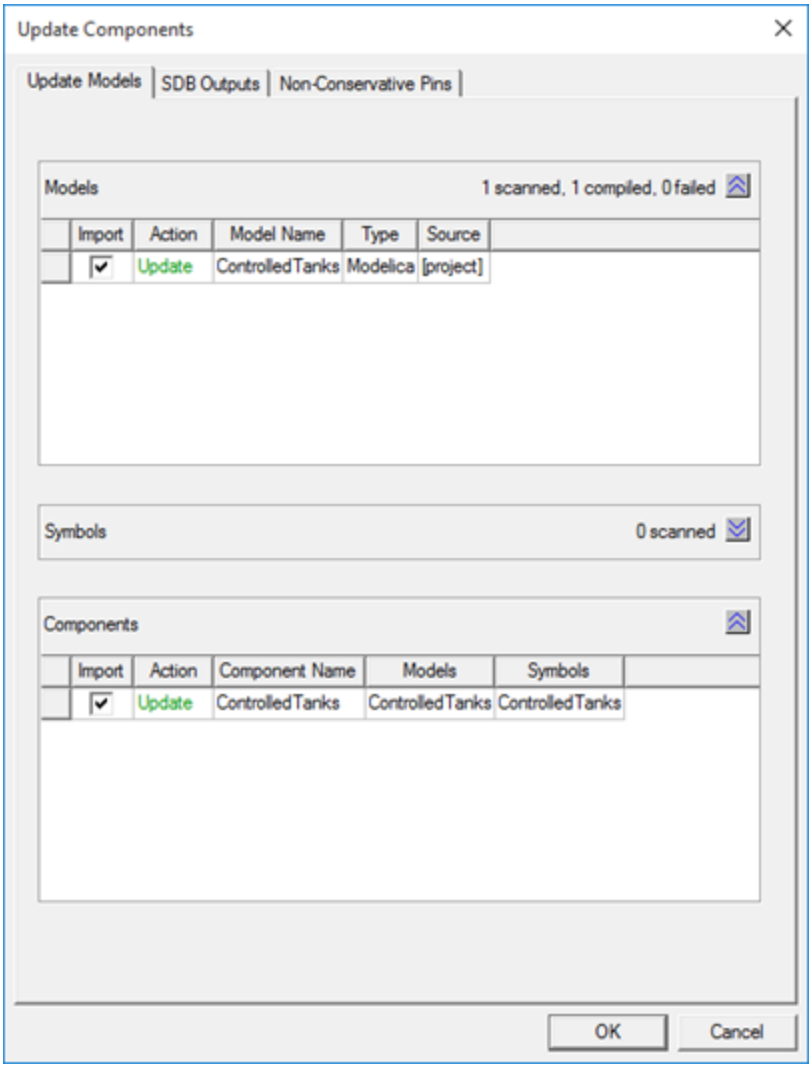


Note	This same dialog box appears when you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For the Controlled Tanks example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. This may take several seconds.
3. Following the compile operation, you can select interfaces for the Controlled Tanks model. Make sure to select outputs that need to be plotted and parameters that need external input. In this example, you should at least select **start->on**, **tank1->level**, and **tank2->level**, in order to generate simulation reports as demonstrated later in this tutorial..

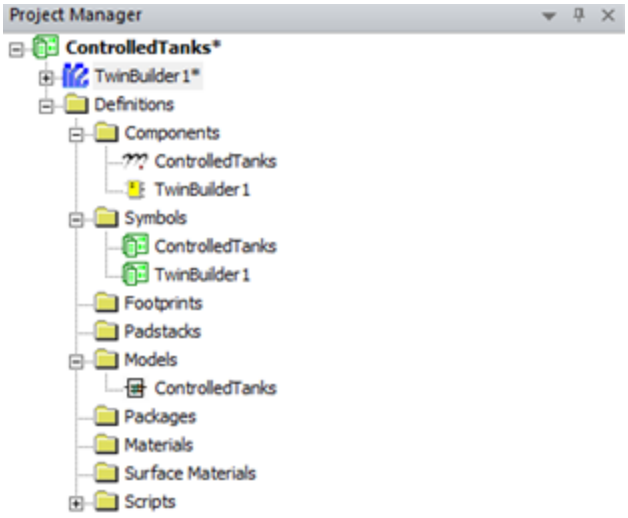


4. When finished selecting interfaces, click **Apply Changes** to close the dialog and apply the changes to the Controlled Tanks model. The **Update Components** dialog is displayed with both the Controlled Tanks model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the Controlled Tanks model and component into Twin Builder.

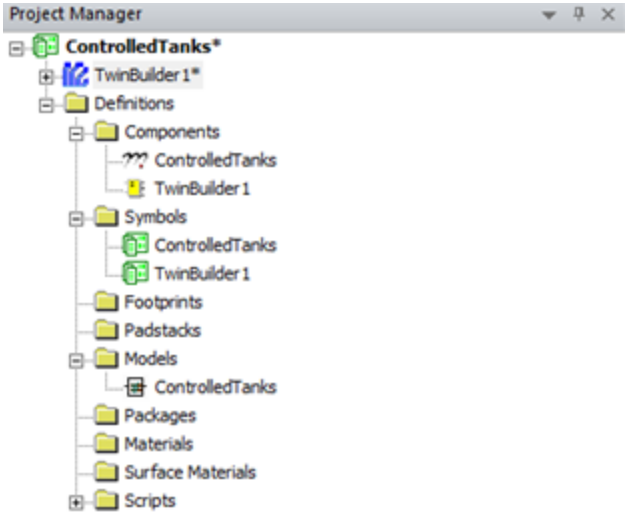
The Modelica Heat Losses component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



Next, you will add the component to a Twin Builder schematic.

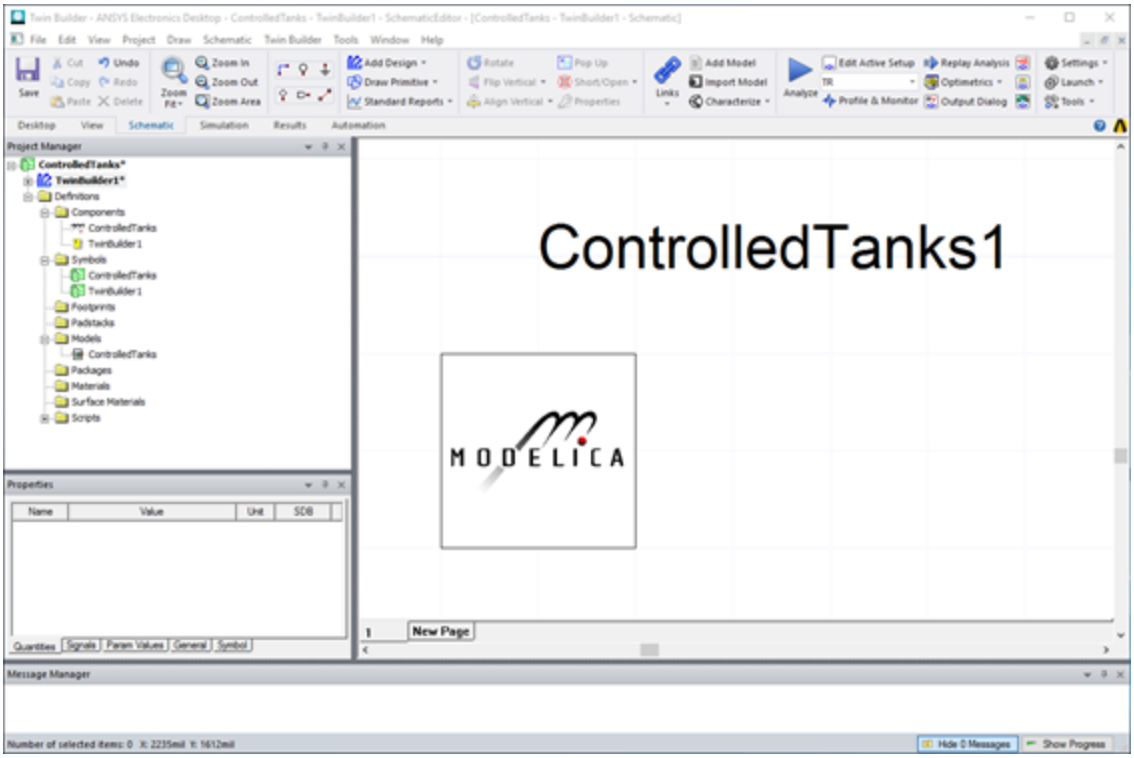
Adding the Controlled Tanks Component to a Twin Builder Schematic

When the Modelica Controlled Tanks component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.

2. Locate the Modelica **ControlledTanks** component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Because the entire Controlled Tanks system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

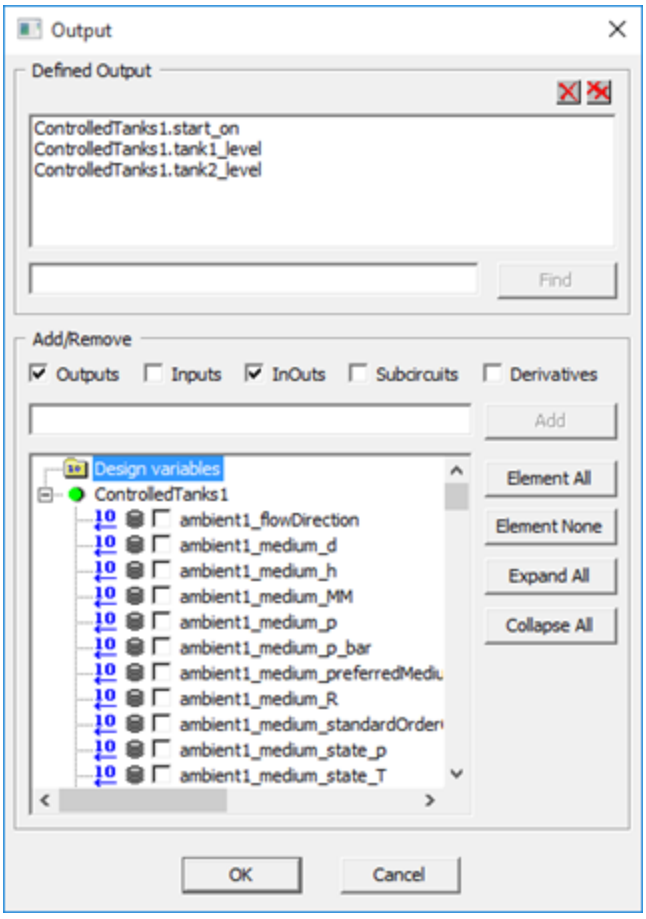
Next, you will define outputs and create reports for simulation results.

Defining Outputs and Creating Reports for the Controlled Tanks Design

When the Modelica Controlled Tanks component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder.

1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, three outputs are selected, **ControlledTanks1.start_on**, **ControlledTanks1.tank1_level**, and **ControlledTanks1.tank2_level** as shown below. Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.

   If a quantity of interest inside the Modelica model is not available to select as output, go back to ***Compiling and Updating the Controlled Tanks Project*** and check if the quantity is selected as an interface when the Modelica model is compiled.



2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

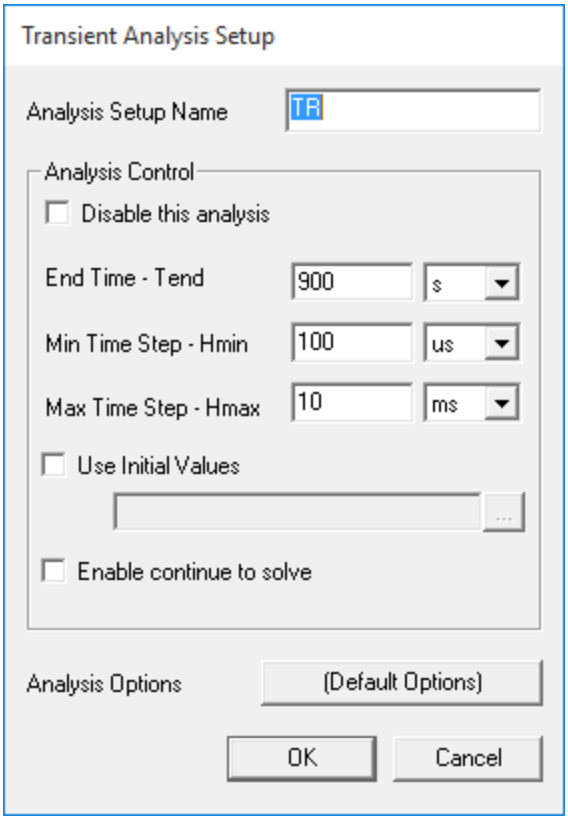
   The outputs you set in the **Output** dialog will be available in the Report dialogs for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate Controlled Tanks simulation results.

Setting Up and Running a Transient Analysis for the Controlled Tanks Design

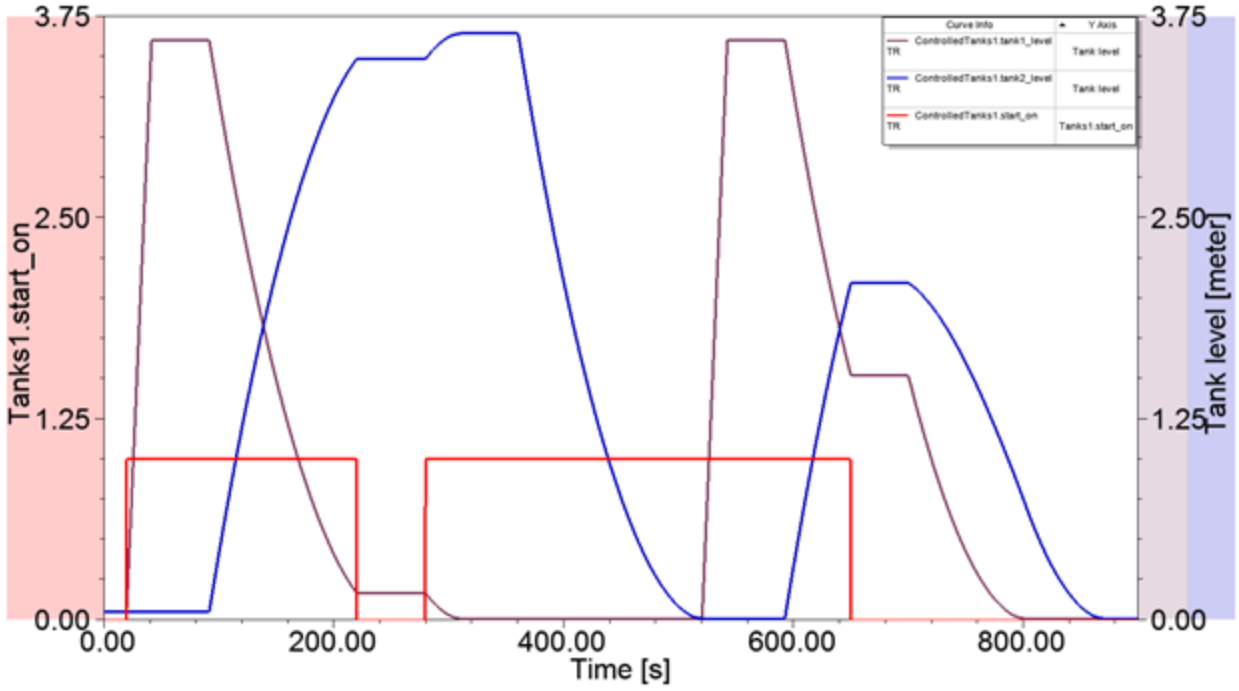
In this example, we would like to simulate the time-domain response of the Controlled Tanks system. This requires that a transient analysis needs to be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

1. Double-click on **TR** to set up end time and the desired time step for the simulation.



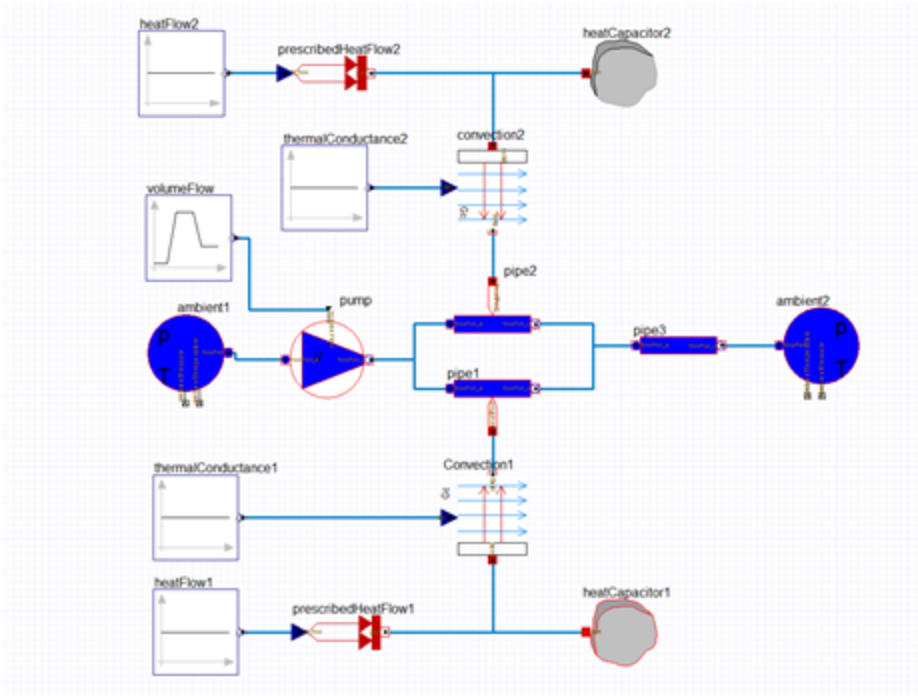
2. For this example, set **End Time** to 900s, **Min Time Step** to 100us, and **Max Time Step** to 10 ms. For more details about transient analysis, see *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
    - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
    - In the Project Manager, right click on **TR** of the design, then select **Analyze**.

Below is a sample report of the simulation results.



Thermal Example: Parallel Pump Drop Out

This example demonstrates a thermal system in which two prescribed heat sources dissipate their heat through thermal conductors via coolant flows.



The coolant flow is taken from an ambient source, and is driven by a pump with prescribed constant flow rate. The coolant flow is then split into two flows connected to the two heat sources, and afterwords merged. The pump is set up to run for 0.2s; then is shut down (using a ramp of 0.2s) for 0.2s; then started again (using a ramp of 0.2s).

This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\ParallelPumpDropOut.aedt

Parallel Pump Drop Out Example Workflow

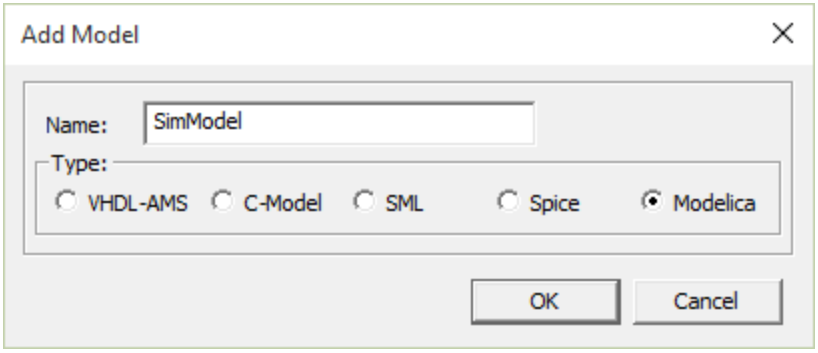
In this example, we will create a Modelica parallel pump drop out model and analyze its performance in the Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the Parallel Pump Drop Out model into the **Diagram Editor**.
- Connect the Modelica Controlled Tanks components.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.

- Define outputs and Create Reports for the Controlled Tanks simulation results.
- Set up and run a Transient analysis for the Controlled Tanks design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.
2. Create a new Modelica model by doing one of the following:
   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

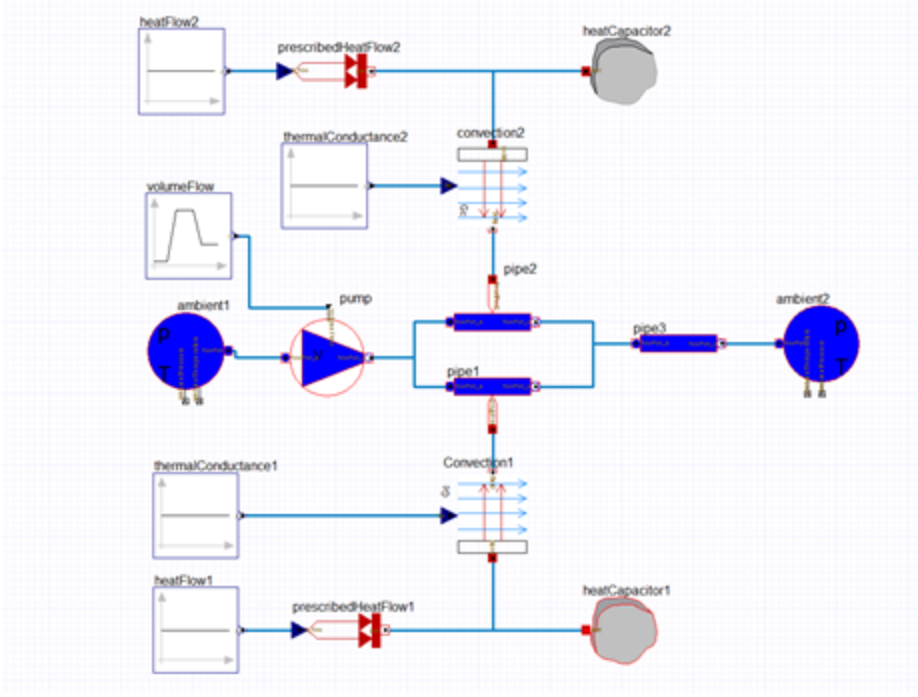
Placing Modelica Components for the Parallel Pump Drop Out Model

All of the required components for the Controlled Tanks example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
heatFlow1, heatFlow2	Modelica/Blocks/Sources/Constant
prescribedHeatFlow, prescribedHeatFlow2	Modelica/Thermal/HeatTransfer/Sources/PrescribedHeatFlow
heatCapacitor1, heatCapacitor2	Modelica/Thermal/HeatTransfer/Components/HeatCapacitor

Component Instance	Location in Library
convection1, convection2	Modelica/Thermal/HeatTransfer/Components/Convection
thermalConductance1, thermalConductance2	Modelica/Blocks/Sources/Constant
volumeFlow	Modelica/Thermal/FluidHeatFlow/Examples/Utilities/DoubleRamp
ambient1, ambient2	Modelica/Thermal/FluidHeatFlow/Sources/Ambient
pump	Modelica/Thermal/FluidHeatFlow/Sources/VolumeFlow
pipe1, pipe2	Modelica/Thermal/FluidHeatFlow/Components/HeatedPipe
pipe3	Modelica/Thermal/FluidHeatFlow/Components/IsolatedPipe

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press **Esc** after placing each component to stop the placement action. Connections will be added later.

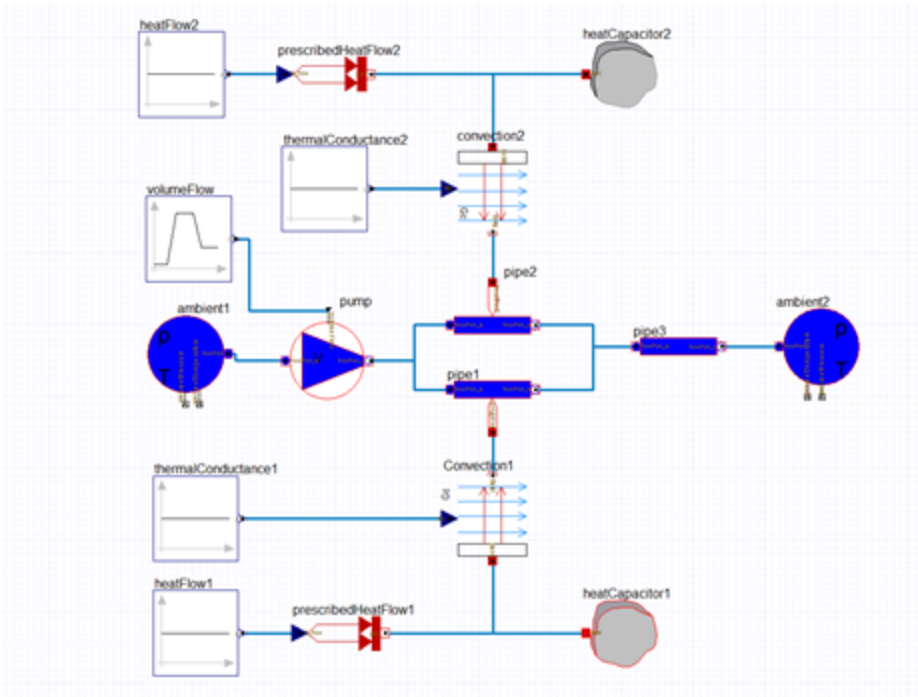


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press **Esc** to complete each action.

Next, you will add connections between the various Parallel Pump Drop Out components.

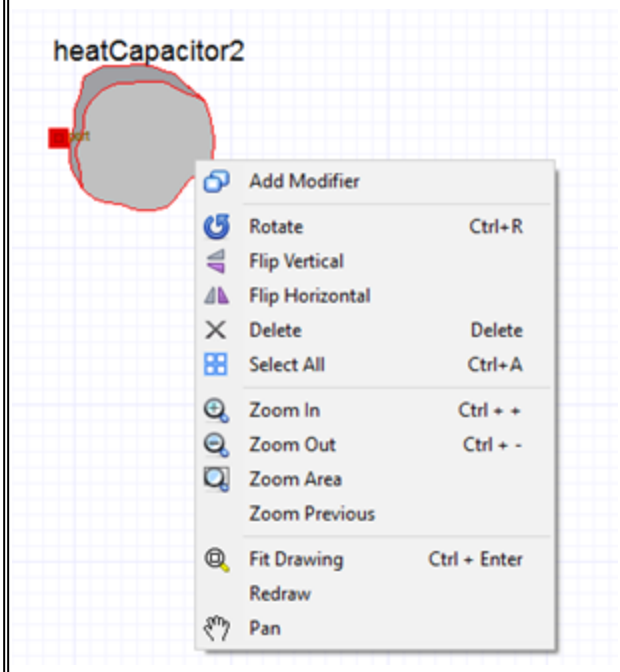
Connecting Modelica Parallel Pump Drop Out Components

To connect the components for the Parallel Pump Drop Out example as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will set up model properties for the Parallel Pump Drop Out example.

Setting Up Model Properties for the Parallel Pump Drop Out Example

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
heatFlow1	k = 5
heatFlow2	k = 10
prescribedHeatFlow1	default values
prescribedHeatFlow2	default values
heatCapacitor1	C = 0.1 T(fixed = true)
heatCapacitor2	C = 0.1 T(fixed = true)
convection1	default values
convection2	default values
thermalConductance1	k = 1
thermalConductance2	k = 1

Component Instance	Properties
volumeFlow	default values
ambient1	constantAmbientPressure = 0 constantAmbientTemperature = 293.15
ambient2	constantAmbientPressure = 0 constantAmbientTemperature = 293.15
pump	useVolumeFlowInput = true constantVolumeFlow = 1 m = 0 T0 = 293.15 T0fixed = false tapT = 1
pipe1	h_g = 0 m = 0.1 T0 = 293.15 T0fixed = true V_flowLaminar = 0.1 dpLaminar = 0.1 V_flowNominal = 1 dpNominal = 1
pipe2	h_g = 0 m = 0.1 T0 = 293.15 T0fixed = true V_flowLaminar = 0.1 dpLaminar = 0.1 V_flowNominal = 1 dpNominal = 1
pipe3	h_g = 0

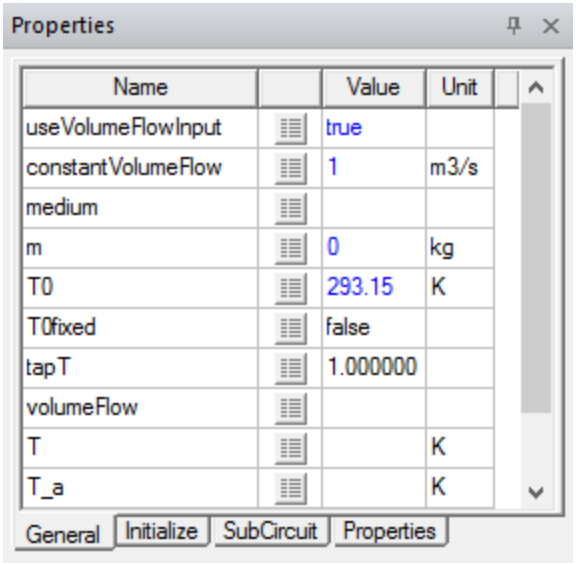
Component Instance	Properties
	m = 0.1 T0 = 293.15 T0fixed = true V_flowLaminar = 0.1 dpLaminar = 0.1 V_flowNominal = 1 dpNominal = 1

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options as needed.

For example, to set properties for the **Spring3** component in the Diagram Editor:

1. Click the **pump** component to display its **Properties** window.



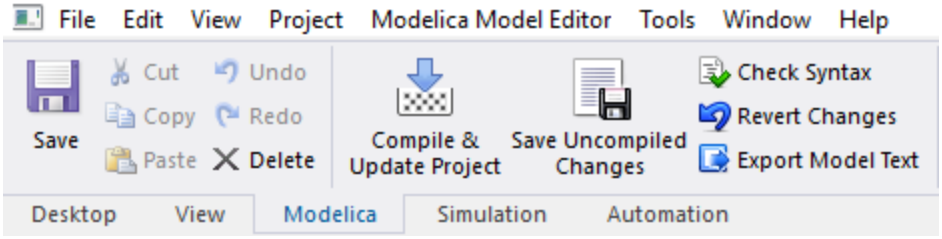
2. On the **General** tab, per the information in the above table, select **true** for the **useVolumeFlowInput** property; **1** for **constantVolumeFlow**, **0** for **m**, **293.15** for **T0**, **false** for **T0fixed**, and 1 for **tapT**. No changes are needed on the other tabs. The Properties tab allows you to set the name of the component instance.

Next, you will compile and update the project.

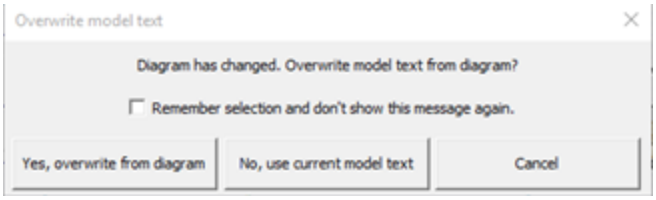
Compiling and Updating the Parallel Pump Drop Out Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.

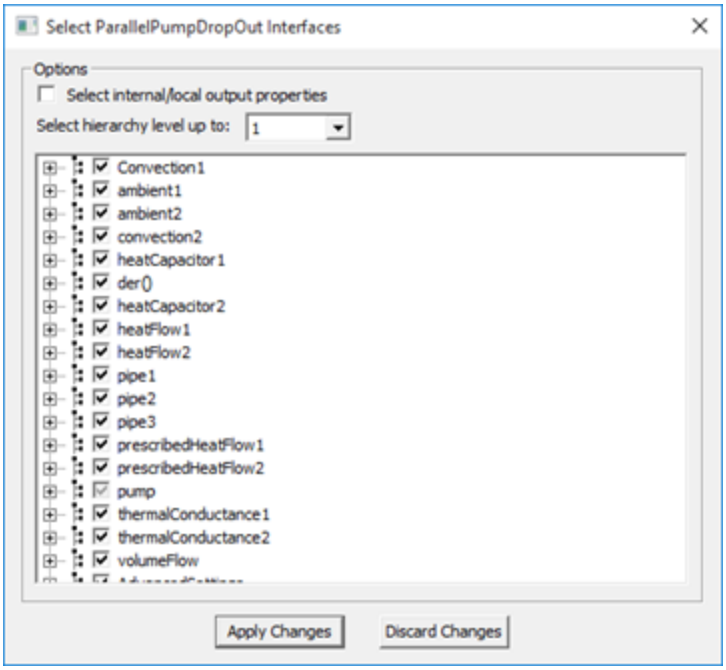


The **Overwrite model text** dialog box appears.

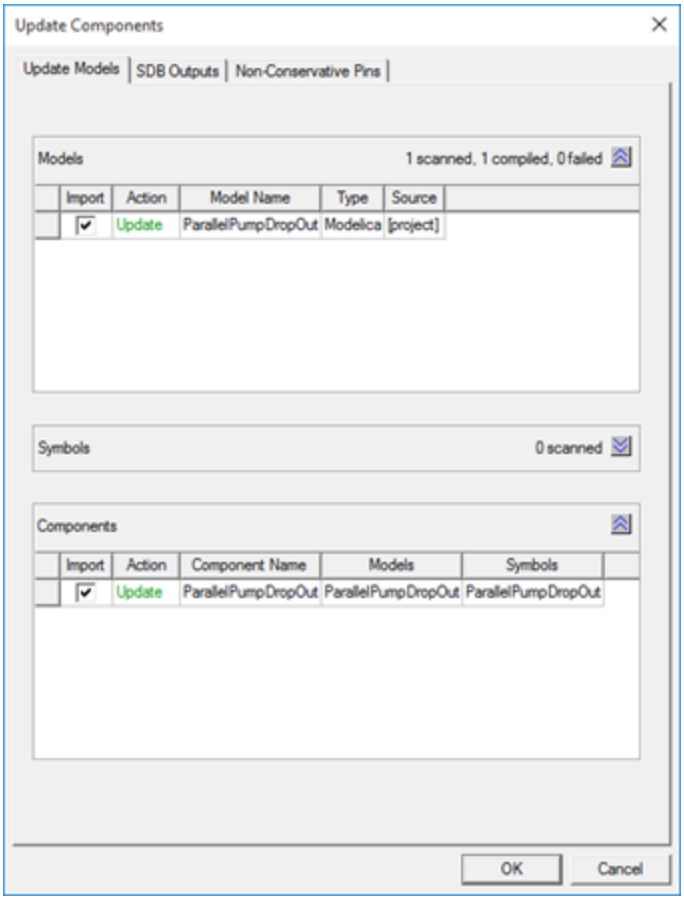


Note	This same dialog box appears when you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For Parallel Pump Drop Out example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. This may take several seconds.
3. Following the compile operation, you can select interfaces for the Parallel Pump Drop Out model, or just accept the default set of preselected interfaces.

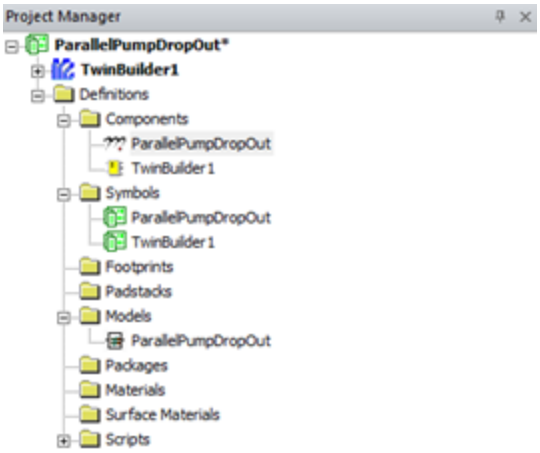


4. When finished selecting interfaces, click **Apply Changes** to close the dialog and apply the changes to the Parallel Pump Drop Out model. The **Update Components** dialog box appears with both the Parallel Pump Drop Out model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the Parallel Pump Drop Out model and component into Twin Builder.

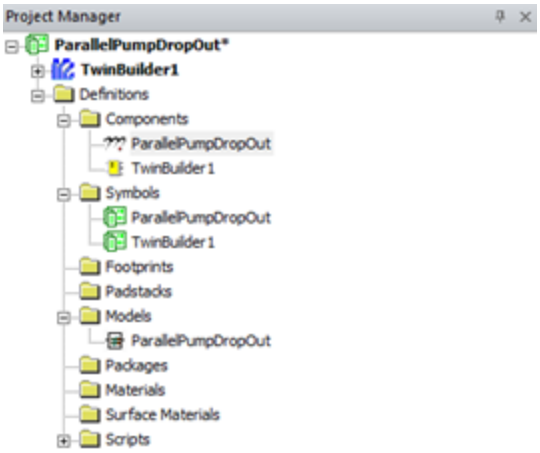
The Modelica Parallel Pump Drop Out component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



Next, you will add the component to a Twin Builder schematic.

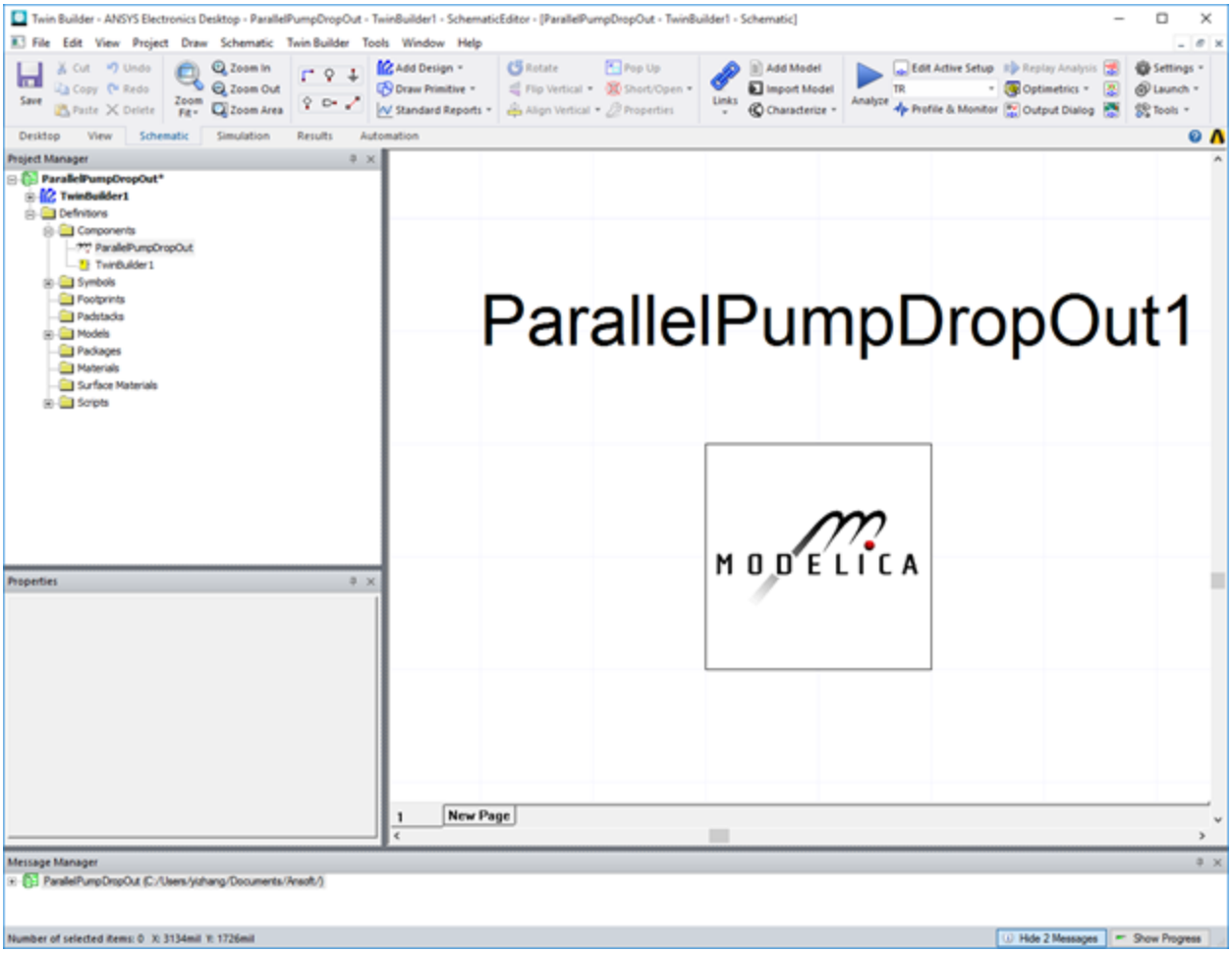
Adding the Parallel Pump Drop Out Component to a Twin Builder Schematic

When the Modelica Parallel Pump Drop Out component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.

2. Locate the Modelica ParallelPumpDropOut component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



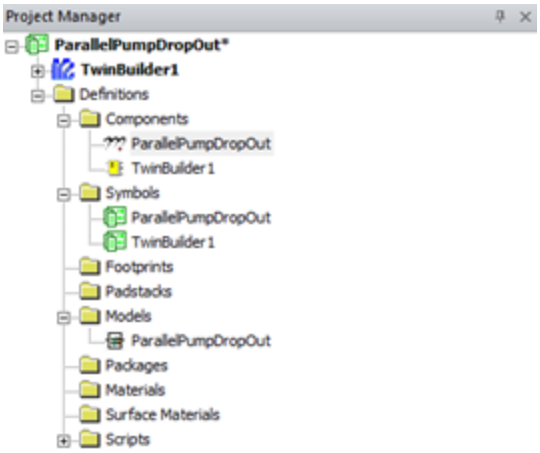
Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Because the entire PID controller system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

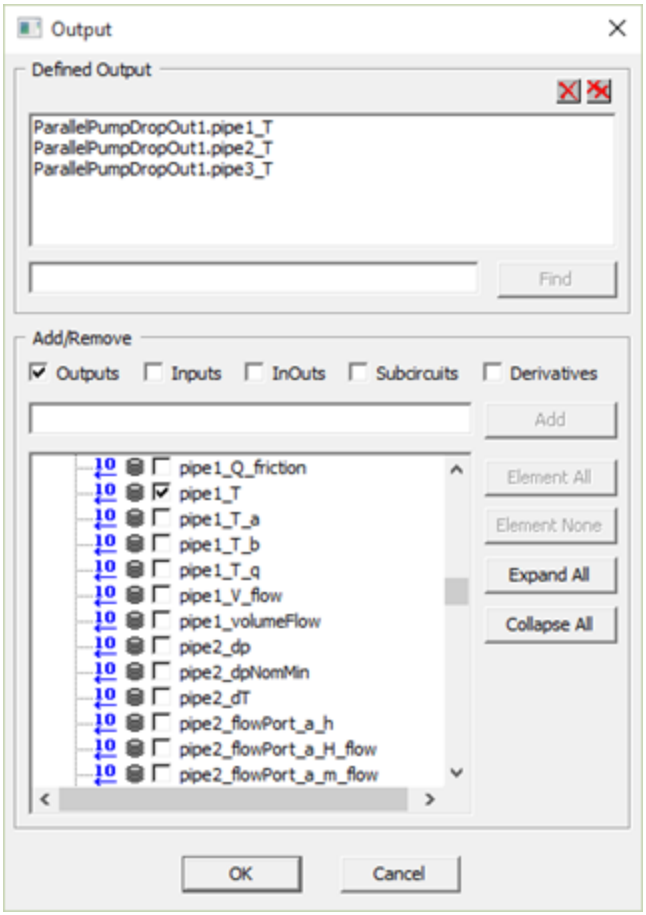
Next, you will define outputs and create reports for simulation results.

Defining Outputs and Creating Reports for the Parallel Pump Drop Out Design

When Modelica Parallel Pump drop Out component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, three outputs are selected, **ParallelPumpDropOut1.pipe1_T**, **ParallelPumpDropOut1.pipe1_T**, and **ParallelPumpDropOut1.pipe3_T**, as shown below. Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.



2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

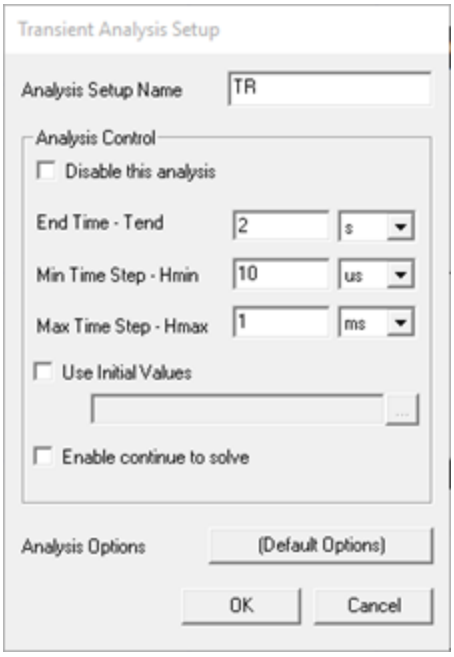
   The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate parallel pump drop out simulation results.

Setting Up and Running a Transient Analysis for the Parallel Pump Drop Out Design

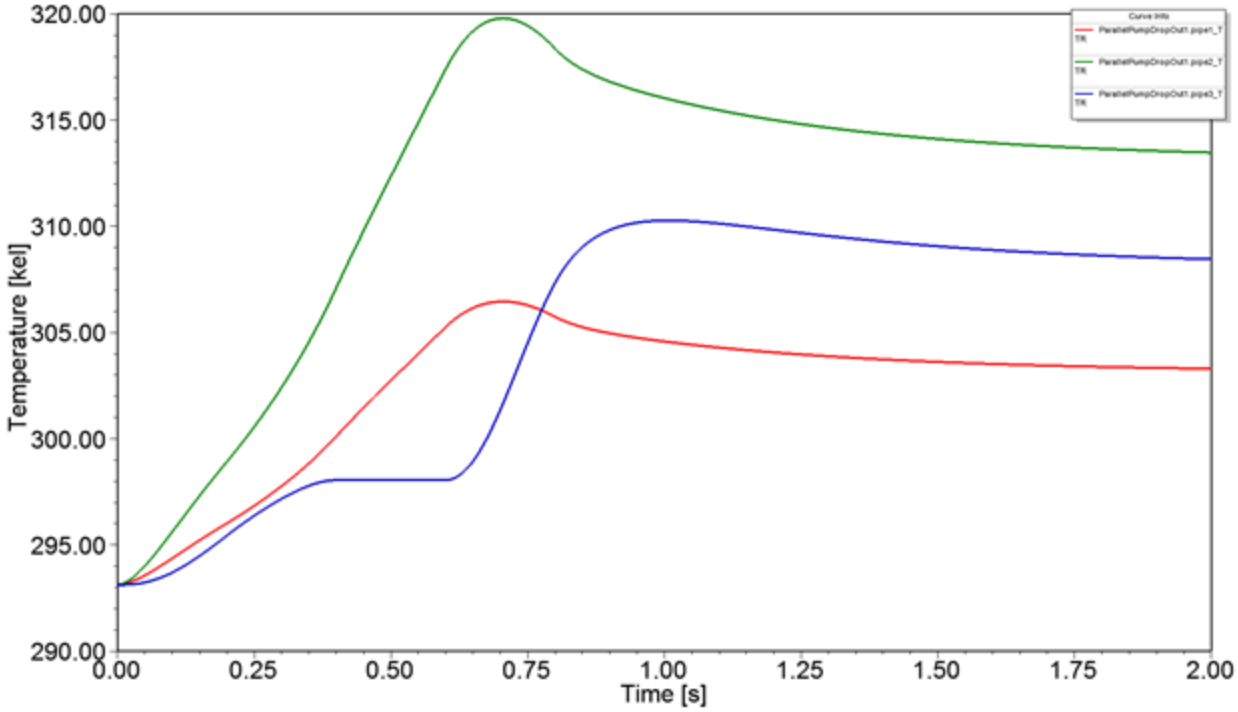
In this example, we would like to simulate the time-domain response of the Parallel Pump Drop Out system. This requires that a transient analysis needs to be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

1. Double-click on **TR** to set up end time and the desired time step for the simulation.



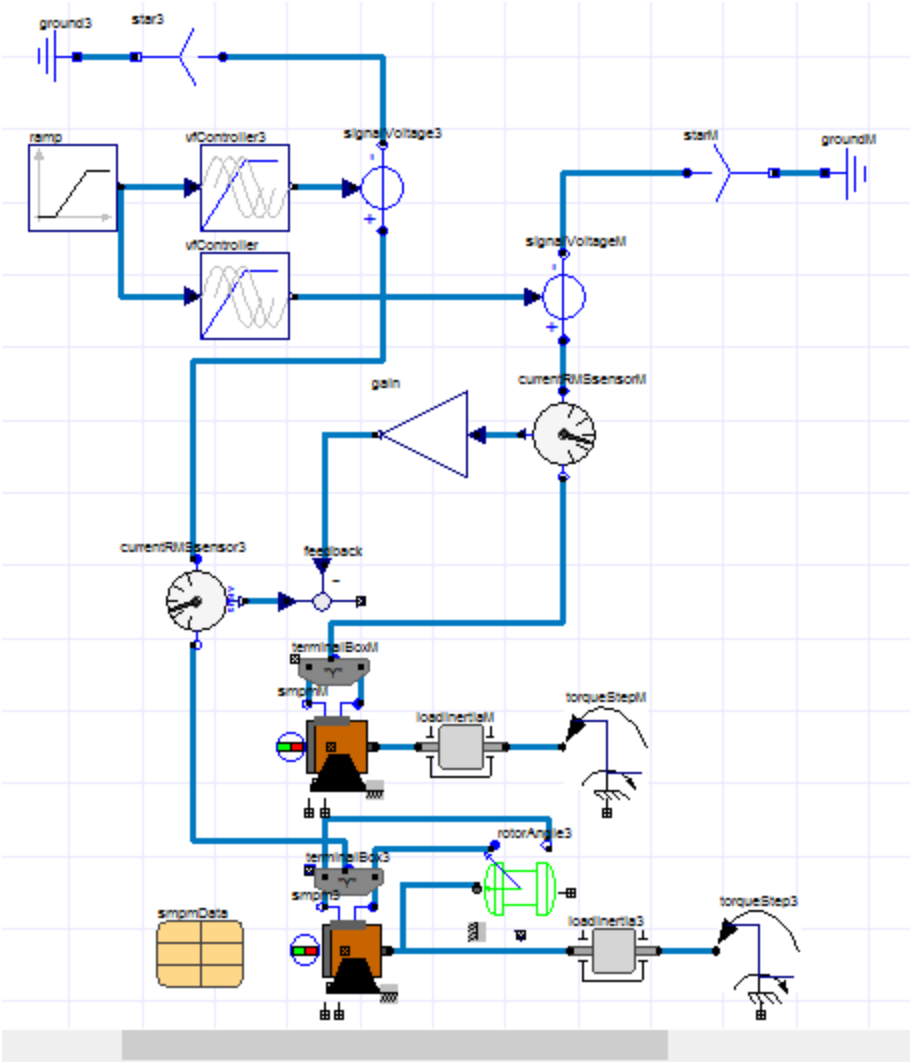
2. For this example, set **End Time** to 2s, **Min Time Step** to 10us, and keep **Max Time Step** as the default 1 ms. For more details about transient analysis, please refer to *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
    - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
    - In the Project Manager, right click on **TR** of the design, then select **Analyze**.

Below is a sample report of the simulation results.



Magnetics Example: SMPM Inverter MultiPhase

This example demonstrates a permanent magnet synchronous induction machine fed by an ideal inverter.



For this example, two equivalent permanent magnet synchronous induction motors with different numbers of phases are modeled and compared, and their equal behavior is demonstrated. An ideal frequency inverter is modeled using **VfController** and **SignalVoltage**. Frequency is then raised by a ramp function, causing the permanent magnet synchronous induction motors to start, and accelerate their respective load inertias. A load torque is applied to each motor at time **tStep**.

This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\SMPM_Inverter_MultiPhase.aedt

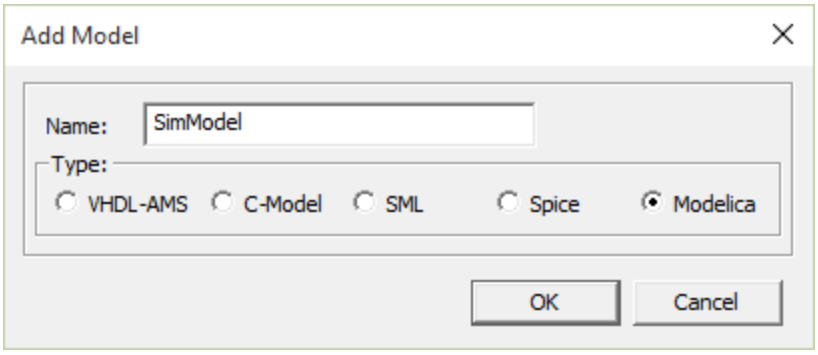
SMPM Inverter MultiPhase Example Workflow

In this example, we will create a Modelica SMPM Inverter MultiPhase model and analyze its performance in the Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the SMPM Inverter MultiPhase model into the **Diagram Editor**.
- Connect the Modelica SMPM Inverter MultiPhase components.
- Provide an initial equation and variable declarations.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Define outputs and Create Reports for the SMPM Inverter MultiPhase simulation results.
- Set up and run a Transient analysis for the SMPM Inverter MultiPhase design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.
2. Create a new Modelica model by doing one of the following:
   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

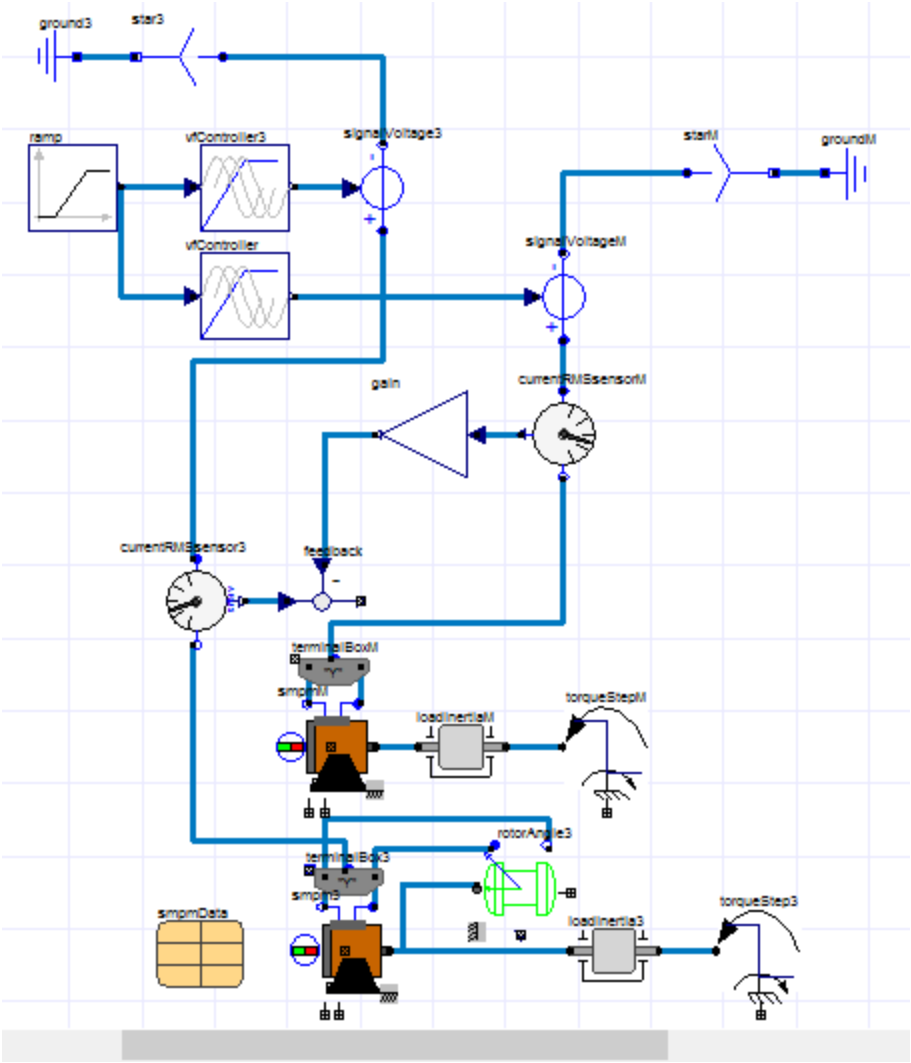
Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

Placing Modelica Components for the SMPM Inverter MultiPhase Model

All of the required components for the SMPM Inverter MultiPhase example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
smpmM and smpm3	Modelica/Magnetic/FundamentalWave/BasicMachines/SynchronousInductionMachines/SM_PermanentMagnet
terminalBoxM and terminalBox3	Modelica/Electrical/Machines/Utilities/TerminalBox
loadInertiaM and loadInertia3	Modelica/Mechanics/Rotational/Components/Inertia
torqueStepM and torqueStep3	Modelica/Mechanics/Rotational/Sources/TorqueStep
rotorAngle3	Modelica/Electrical/Machines/Sensors/RotorDisplacementAngle
smpmData	Modelica/Electrical/Machines/Utilities/ParameterRecords/SM_PermanentMagnetData
currentRMSensorM and currentRMSensor3	Modelica/Electrical/Machines/Sensors/CurrentQuasiRMSSensor
feedback	Modelica/Blocks/Math/Feedback
gain	Modelica/Blocks/Math/Gain
vfController and vfController3	Modelica/Electrical/Machines/Utilities/VfController
signalVoltageM and signalVoltage3	Modelica/Electrical/MultiPhase/Sources/SignalVoltage
ramp	Modelica/Blocks/Sources/Ramp
starM and star3	Modelica/Electrical/MultiPhase/Basic/Star
groundM and ground3	Modelica/Electrical/Analog/Basic/Ground

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press **Esc** after placing each component to stop the placement action. (Connections will be added later.)

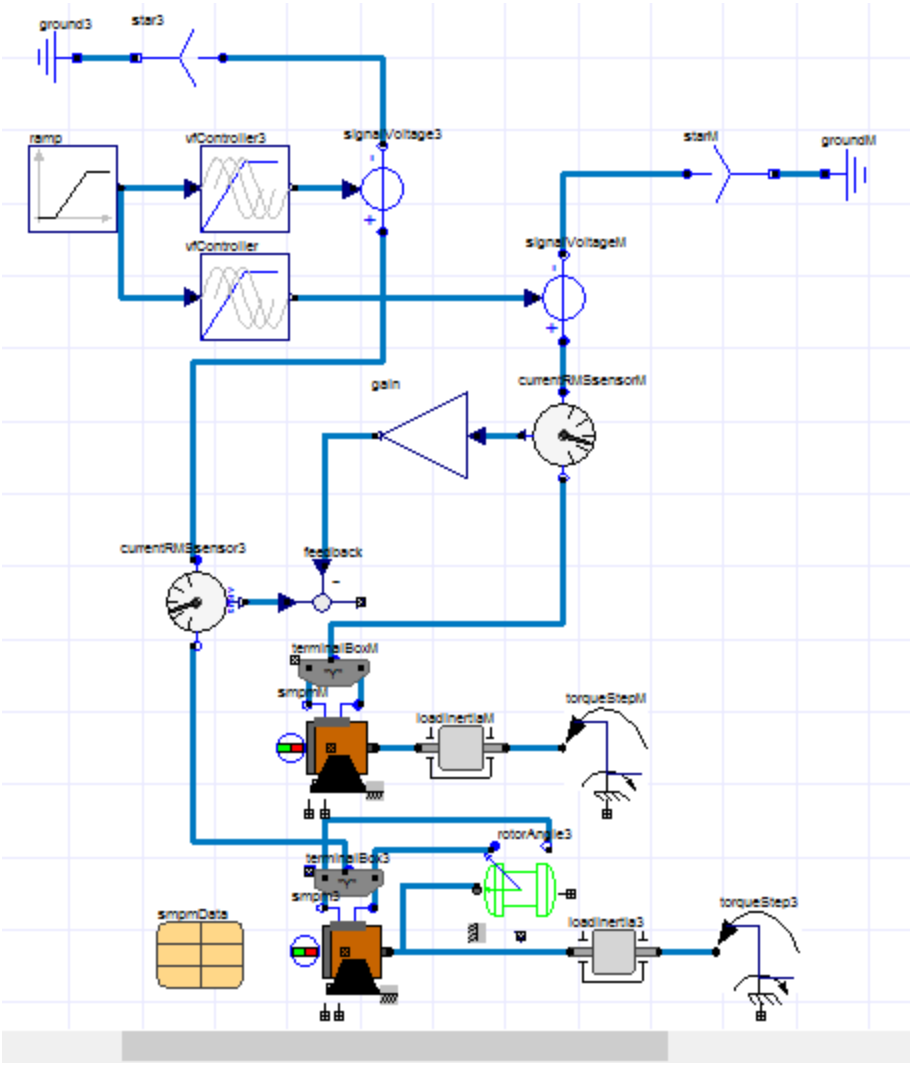


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press **Esc** to complete each action.

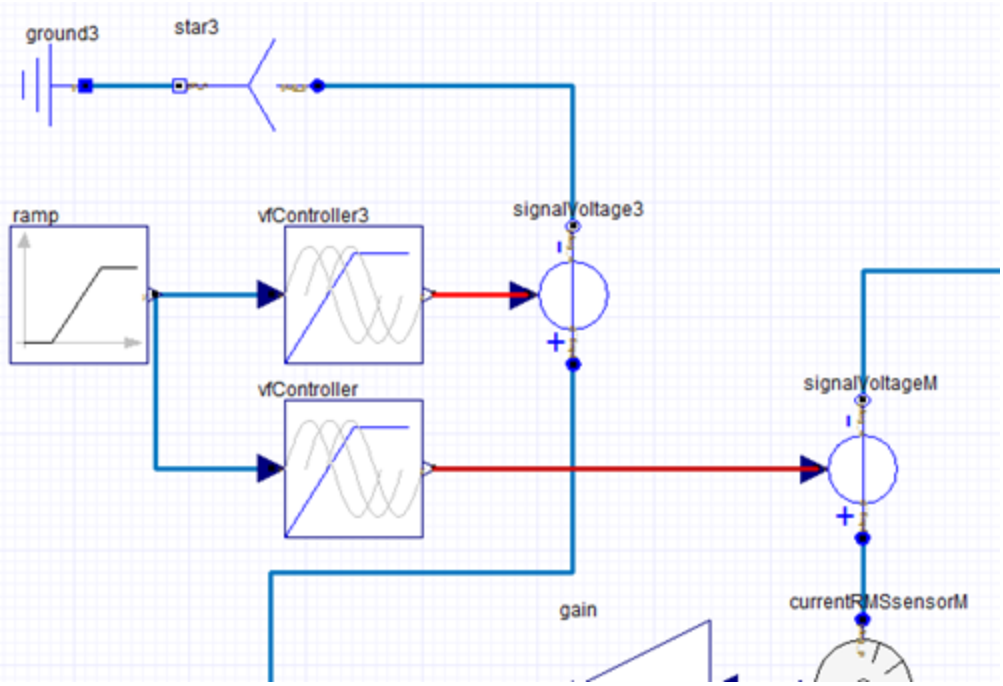
Next, you will add connections between the various SMPM Inverter MultiPhase components.

Connecting Modelica SMPM Inverter MultiPhase Components

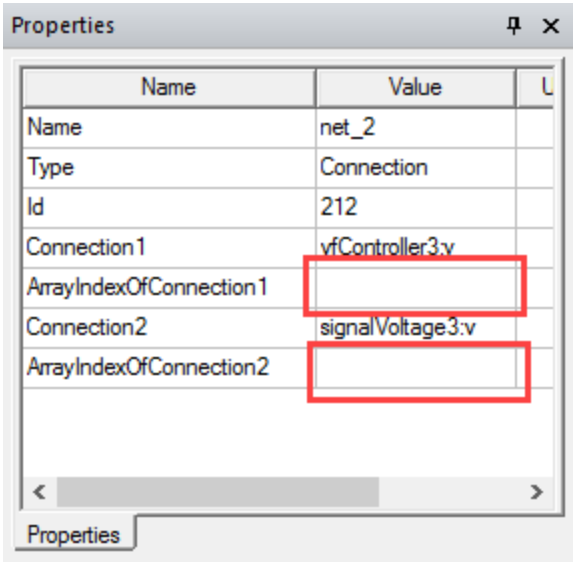
To connect the components for the SMPM Inverter MultiPhase example as shown in the diagram below:

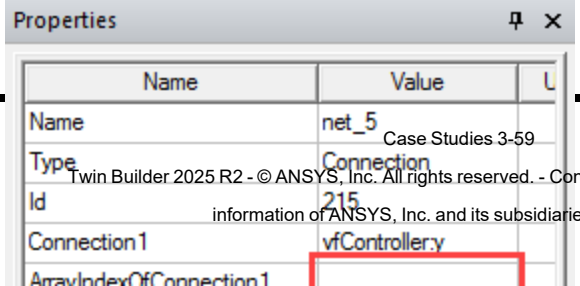


1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.
3. An additional step is required for the connections between model instances of **vfController3** and **signalVoltage3**, and between **vfController** and **signalVoltageM**. These connections are multiphase electrical vector connections that require manual adjustment to match the array size of connectors.



a. For the connection between **vfController3** and **signalVoltage3**, click the connecting wire between the components, then in the Properties window, delete the default numbers for **ArrayIndexOfConnection1** and **ArrayIndexOfConnection2** and leave them empty.



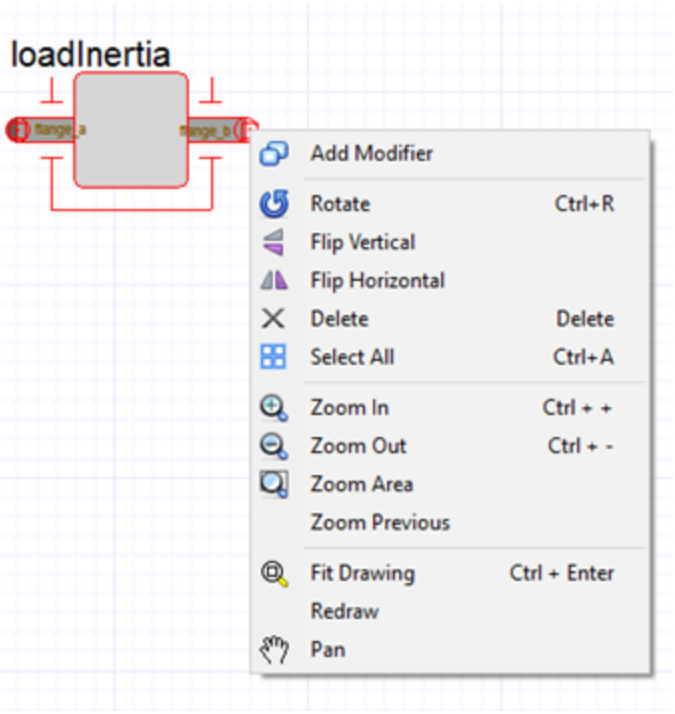


b. Similarly, \fFor the connection between **vfController** and **signalVoltageM**, click the connecting wire between the components, then in the Properties window, delete the default numbers for **ArrayIndexOfConnection1** and **ArrayIndexOfConnection2** and leave them empty.

   The above valve-to-tank settings generate the following lines in the Modelica model text code:

```
connect(vfController3.y, signalVoltage3.v);

connect(vfController.y, signalVoltageM.v);
```

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will add an initial equation and declare a variable for the SMPM Inverter MultiPhase model.

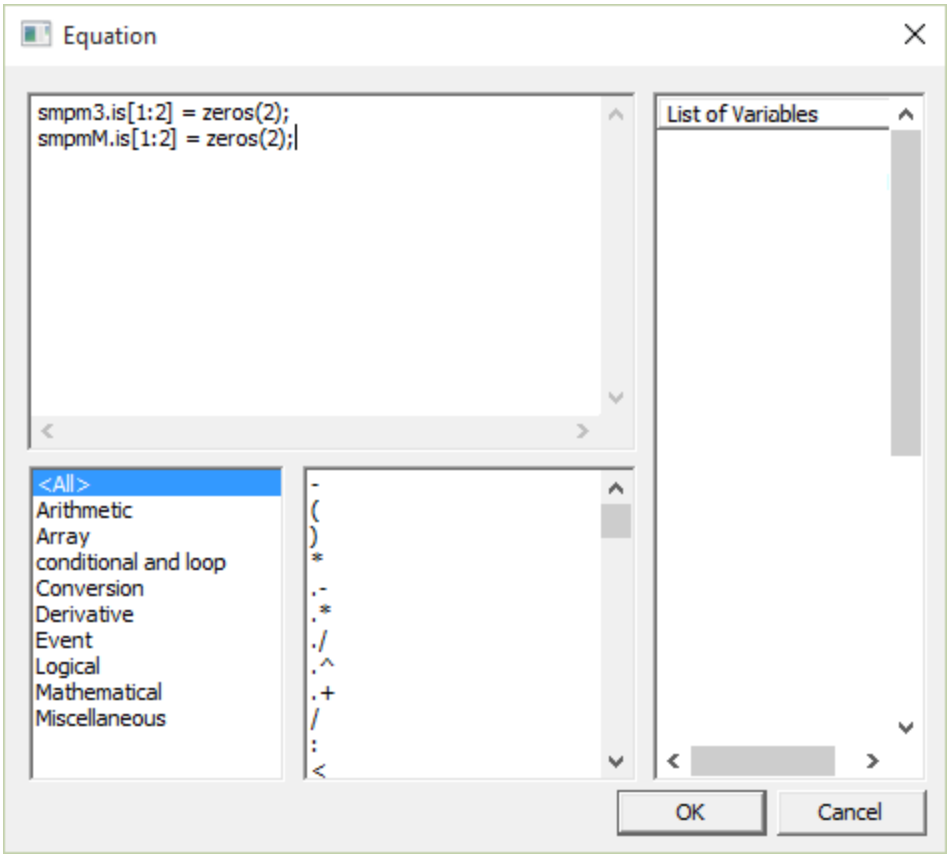
Adding an Initial Equation and Variable Declaration to the SMPM Inverter MultiPhase Design

For this example, an initial equation is needed for each of the synchronous machines, and several variables declared.

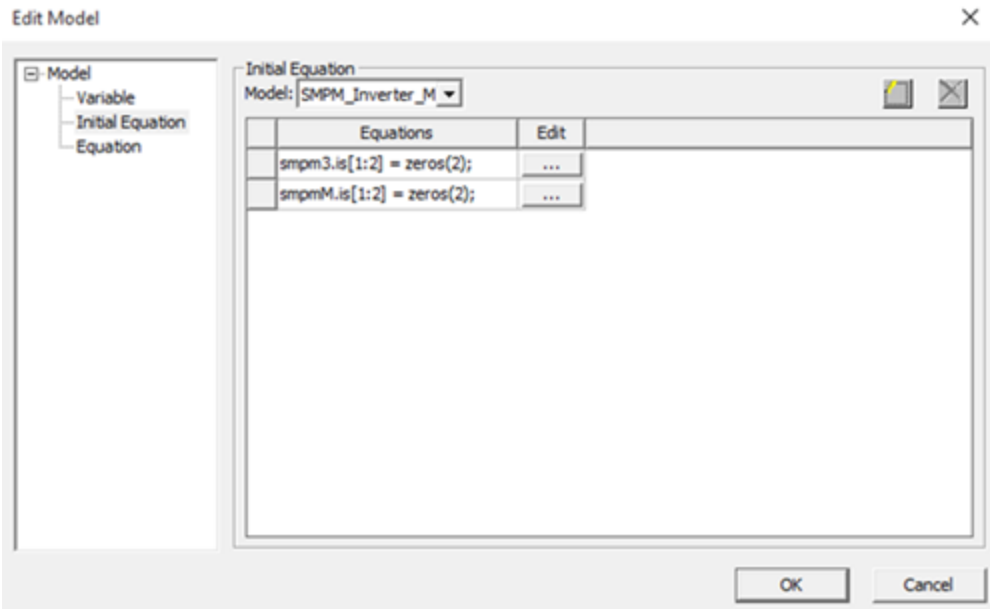
Initial Equation

1. For the initial equation, on the Twin Builder **Modelica** tab, click the Equations icon to open the **Edit Model** dialog box.
2. Select **Initial Equation**, then click the  button in the upper right of the dialog to open the **Equation** editor, and enter the equations:

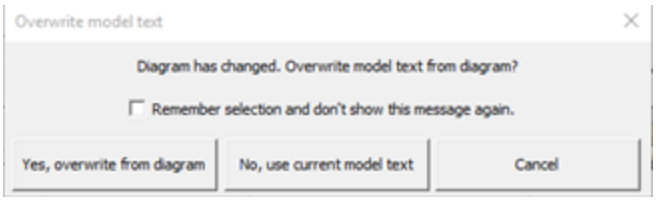
   smpm3.is[1:2] = zeros(2);
   smpmM.is[1:2] = zeros(2);



3. Click **OK** to close the dialog box, adding the equation to the **Edit Model** dialog box.



4. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog box appears.



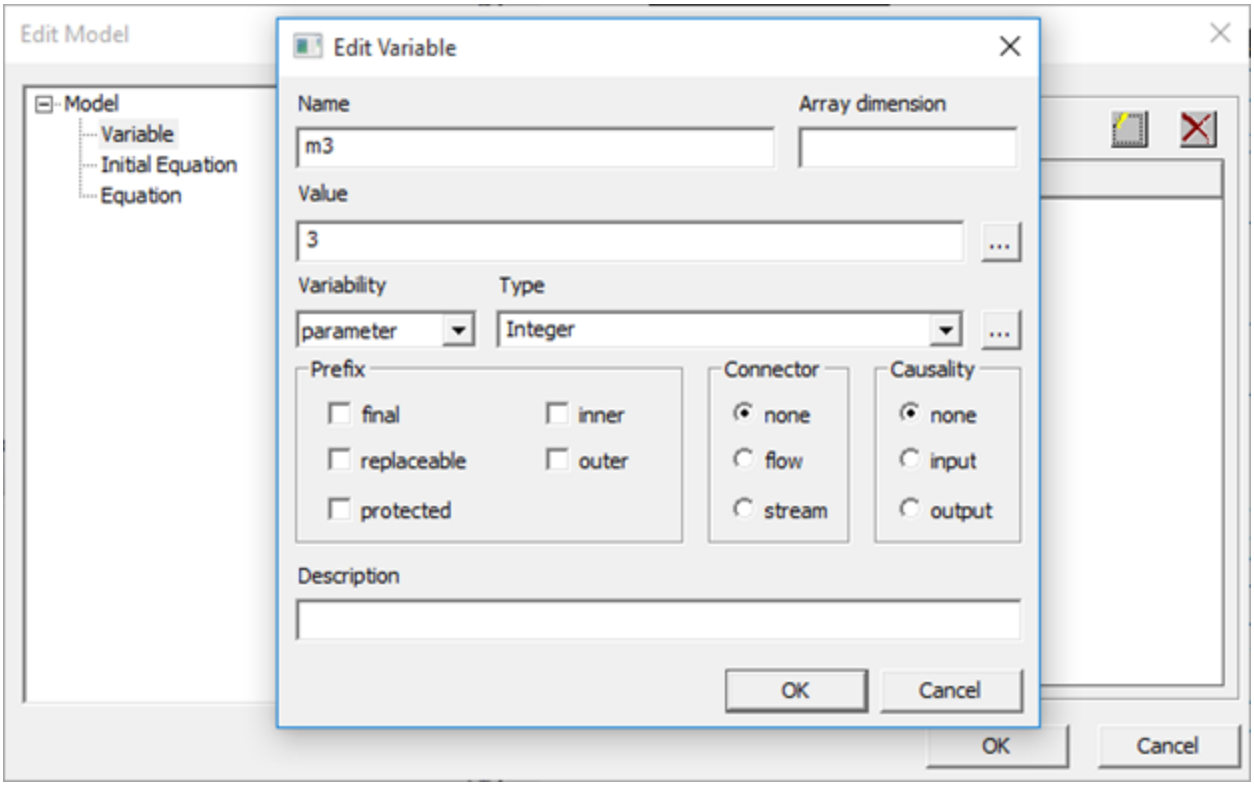
5. Click **Yes, overwrite from diagram** to generate the equation and add it to the model text.

```
initial equation
    smpm3.is[1:2] = zeros(2);
    smpmM.is[1:2] = zeros(2);
```

Variable Declarations

1. To create the needed variables for this design, on the Twin Builder **Modelica** tab, click the **Variables** icon to open the **Edit Model** dialog box.
2. Click the  button in the upper right of the dialog box to open the **Edit Variable** dialog box, and enter the settings such as for the integer parameter **m3=3** shown below. (The

Prefix, Connector, and Causality settings remain the defaults as shown.)



3. Click **OK** to close the dialog box, adding the variable to the **Edit Model** dialog box.
4. Repeat, adding the variables listed in the following table:

Variable Name	Value	Variability Type	Type
m3	3	parameter	Integer
m	5	parameter	Integer
VsNominal	100	parameter	Modelica.SIunits.Voltage
fsNominal	50	parameter	Modelica.SIunits.Frequency
fKnee	50	parameter	Modelica.SIunits.Frequency
tRamp	1	parameter	Modelica.SIunits.Time
T_Load	181.4	parameter	Modelica.SIunits.Torque
tStep	1.2	parameter	Modelica.SIunits.Time
J_Load	0.29	parameter	Modelica.SIunits.Inertia

5. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog box appears.
6. Click **Yes, overwrite from diagram** to add the variable declarations to the model text.

```
//Declaration(s)
parameter Integer m3 = 3;
parameter Integer m = 5;
parameter Modelica.SIunits.Voltage VsNominal = 100;
parameter Modelica.SIunits.Frequency fsNominal = 50;
parameter Modelica.SIunits.Frequency fKnee = 50;
parameter Modelica.SIunits.Time tRamp = 1;
parameter Modelica.SIunits.Torque T_Load = 181.4;
parameter Modelica.SIunits.Time tStep = 1.2;
parameter Modelica.SIunits.Inertia J_Load = 0.29;
```

Next, you will set up properties for the SMPM Inverter MultiPhase model.

Setting Up Model Properties for the SMPM Inverter MultiPhase Example

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
smpmM	m = m Jr = smpmData.Jr p = smpmData.p fsNominal = smpmData.fsNominal VsOpenCircuit = smpmData.VsOpenCircuit TsOperational = 20 TrOperational = 20 Rs = smpmData.Rs*m/3 TsRef = smpmData.TsRef alpha20s = smpmData.alpha20s Lssigma = smpmData.Lssigma*m/3 Lszero = smpmData.Lszero*m/3 Lmd = smpmData.Lmd*m/3

Component Instance	Properties
	Lmq = smpmData.Lmq*m/3 useDamperCage = smpmData.useDamperCage Lrsigmad = smpmData.Lrsigmad Lrsigmaq = smpmData.Lrsigmaq Rrd = smpmData.Rrd Rrq = smpmData.Rrq TrRef = smpmData.TrRef alpha20r = smpmData.alpha20r frictionParameters = smpmData.frictionParameters statorCoreParameters = smpmData.statorCoreParameters strayLoadParameters = smpmData.strayLoadParameters permanentMagnetLossParameters = smpmData.permanentMagnetLossParameters **NOTE:** For the **phiMechanical**, **wMechanical**, and **ir** properties, refer to the "Modification of variables ir, phiMechanical and wMechanical " on page 3-68 step below.
Smpm3	m = m3 Jr = smpmData.Jr p = smpmData.p fsNominal = smpmData.fsNominal VsOpenCircuit = smpmData.VsOpenCircuit TsOperational = 20 TrOperational = 20 Rs = smpmData.Rs TsRef = smpmData.TsRef alpha20s = smpmData.alpha20s Lssigma = smpmData.Lssigma Lszero = smpmData.Lszero

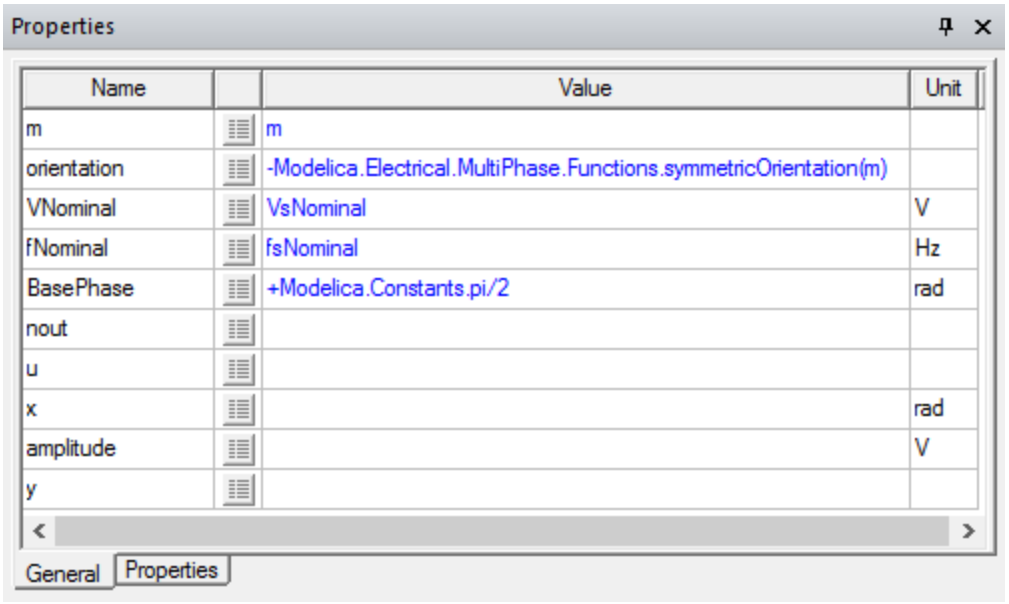
Component Instance	Properties
	Lmd = smpmData.Lmd Lmq = smpmData.Lmq useDamperCage = smpmData.useDamperCage Lrsigmad = smpmData.Lrsigmad Lrsigmaq = smpmData.Lrsigmaq Rrd = smpmData.Rrd Rrq = smpmData.Rrq TrRef = smpmData.TrRef alpha20r = smpmData.alpha20r frictionParameters = smpmData.frictionParameters statorCoreParameters = smpmData.statorCoreParameters strayLoadParameters = smpmData.strayLoadParameters permanentMagnetLossParameters = smpmData.permanentMagnetLossParameters **NOTE:** For the **phiMechanical**, **wMechanical**, and **ir** properties, refer to the "Modification of variables ir, phiMechanical and wMechanical " on page 3-68 step below.
terminalBoxM	m = m terminalConnection = “Y”
terminalBox3	m = m3 terminalConnection = “Y”
loadInertiaM	J = J_Load
loadInertia3	J = J_Load
torqueStepM	stepTorque = -T_Load offsetTorque = 0 startTime = tStep
torqueStepM	stepTorque = -T_Load offsetTorque = 0 startTime = tStep

Component Instance	Properties
rotorAngle3	p = smpm3.p
smpmData	Use defaults
currentRMSensorM	m = m
currentRMSensor3	m = m3
feedback	Use defaults
gain	k = (m/m3)
vfController	m = m orientation = -Modelica.Electrical.MultiPhase.Functions.symmetricOrientation(m) VNominal = VsNominal fNominal = fsNominalBasePhase = +Modelica.Constants.pi/2
vfController3	m = m3 orientation = -Modelica.Electrical.MultiPhase.Functions.symmetricOrientation(m3) VNominal = VsNominal fNominal = fsNominalBasePhase = +Modelica.Constants.pi/2
signalVoltageM	m = m
signalVoltage3	m = m3
ramp	height = fKnee duration = tRamp
starM	m = m
star3	m = m3
groundM	Use defaults
ground3	Use defaults

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options from the table above as needed. For example, to set properties for the **vfController** component in the Diagram Editor:

a. Click on the **vfController** component to display its **Properties** window.



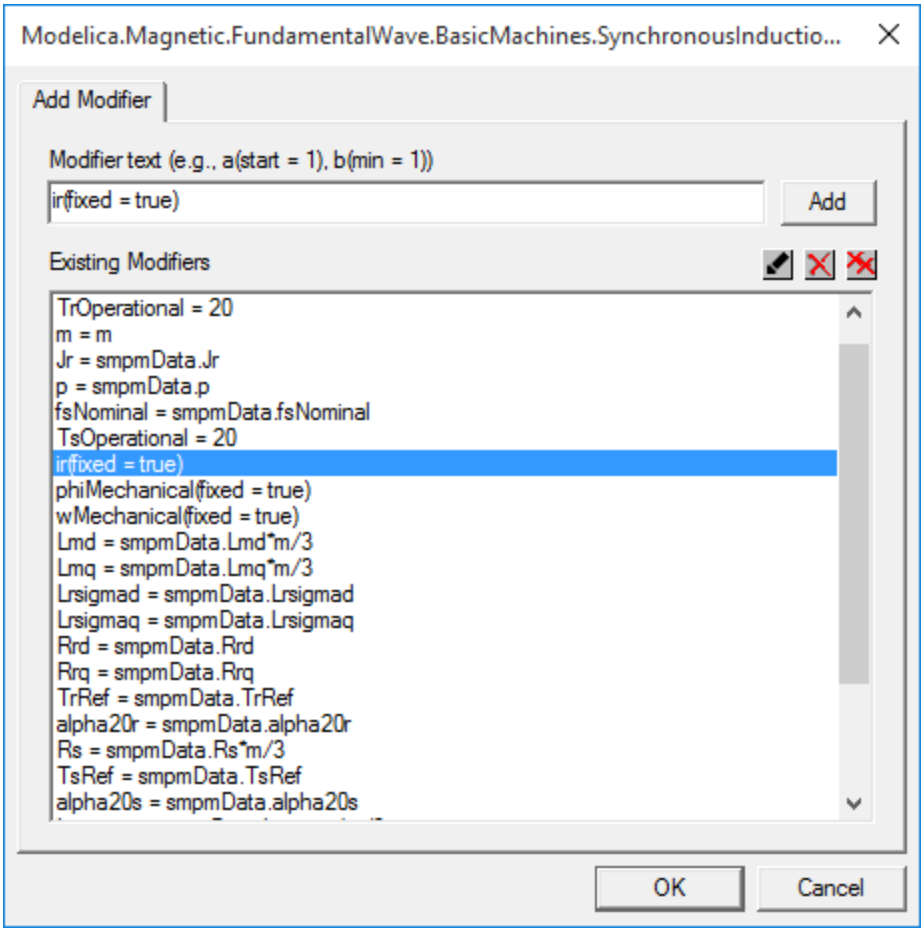
b. On the **General** tab, per the information in the above table, enter **m** for the **m** property; **VsNominal** for **VNominal**, **fsNominal** for **fNominal**, **Modelica.Electrical.MultiPhase.Functions.symmetricOrientation(m)** for **orientation**, and **+Modelica.Constants.pi/2** for **BasePhase**. No other changes are needed. The **Properties** tab allows you to set the name of the component instance.

Modification of variables ir, phiMechanical and wMechanical

This example requires modification of variables **ir**, **phiMechanical** and **wMechanical** for the machines **smpmM** and **smpm3**. This cannot be done directly in the Properties window, but instead requires that modifiers be added.

1. To add modifiers to **smpmM**, select the **smpmM** instance in diagram editor and on the ribbon Modelica tab click **Add Modifier** (or right click on the **smpmM** instance and select **Add Modifier** from the context menu). In the **Add Modifier** dialog box, add the following 3 lines:

```
ir(fixed=true)

wMechanical(fixed = true)

phiMechanical(fixed = true)
```



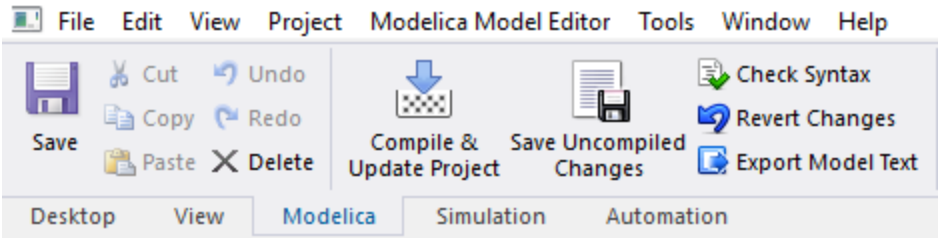
2. Repeat the above to add similar modifiers to **smpm3**.

Next, you will compile and update the project.

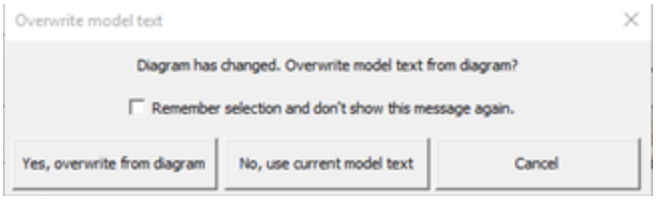
Compiling and Updating the SMPM Inverter MultiPhase Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.

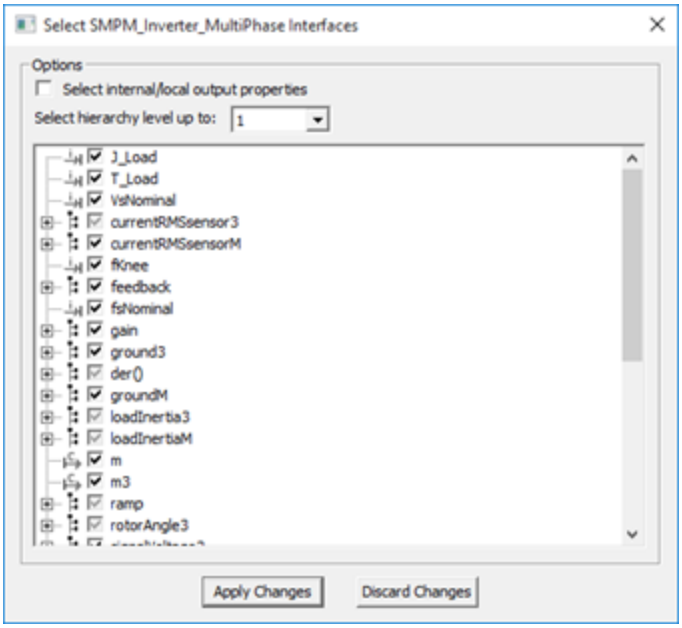


The **Overwrite model text** dialog box appears.

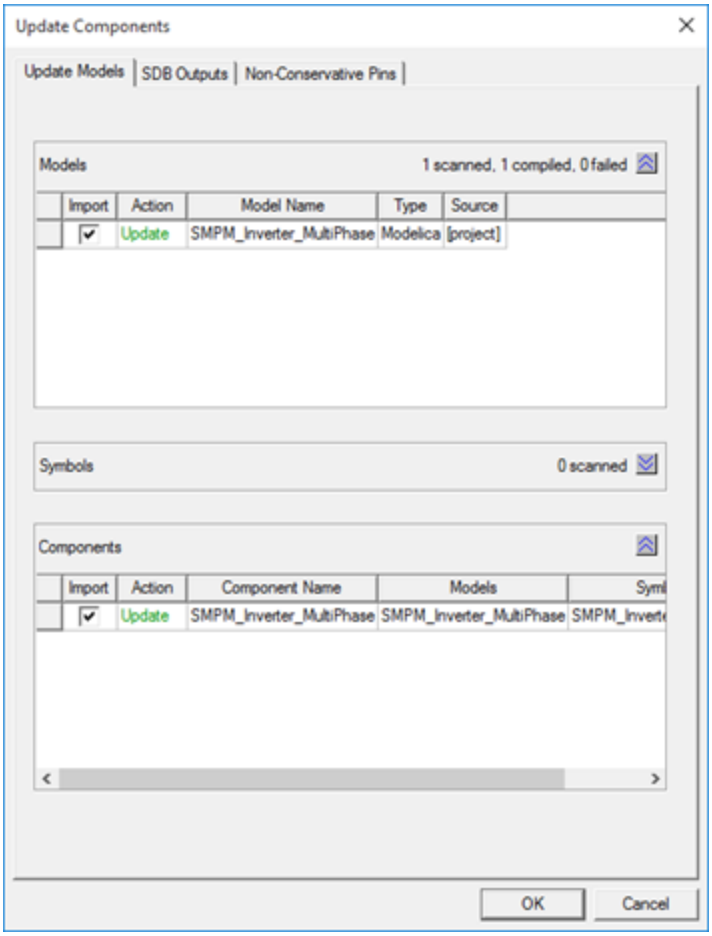


Note	This same dialog box appears when you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For SMPM Inverter MultiPhase example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. This may take several seconds.
3. Following the compile operation, you can select interfaces for the SMPM Inverter MultiPhase model, or just accept the default set of preselected interfaces.

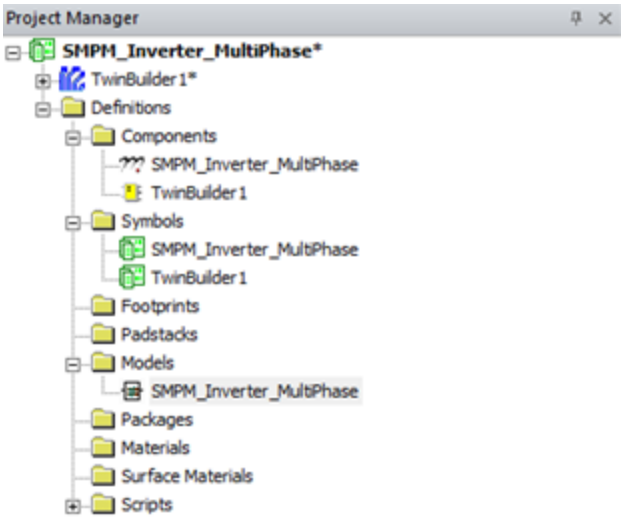


4. When finished selecting interfaces, click **Apply Changes** to close the dialog box and apply the changes to the SMPM Inverter MultiPhase model. The **Update Components** dialog box is displayed with both the SMPM Inverter MultiPhase model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the SMPM Inverter MultiPhase model and component into Twin Builder.

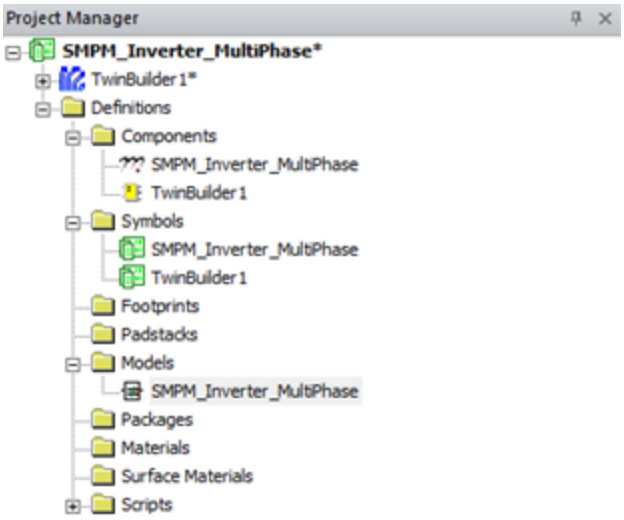
The Modelica SMPM Inverter MultiPhase component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



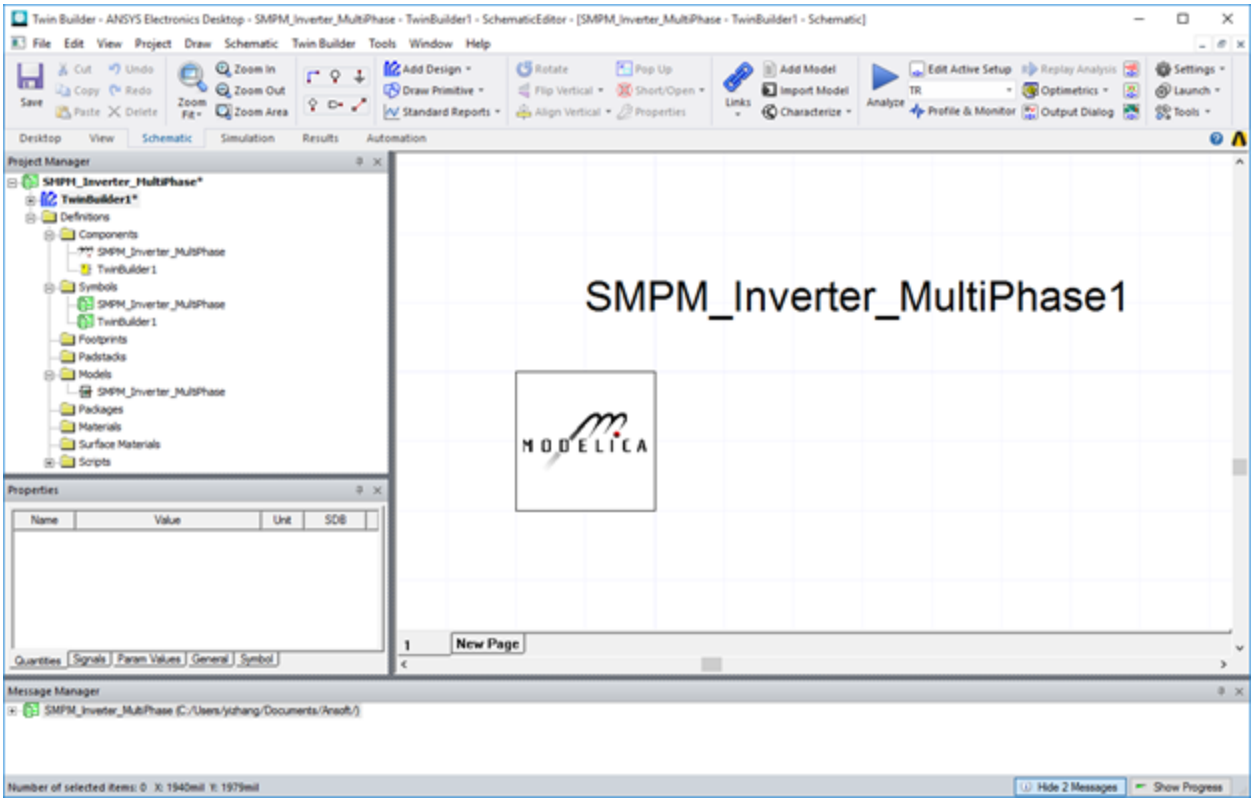
Next, you will add the component to a Twin Builder schematic.

Adding the SMPM Inverter MultiPhase Component to a Twin Builder Schematic

When the Modelica SMPM Inverter MultiPhase component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.
2. Locate the Modelica SMPM_Inverter_MultiPhase component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



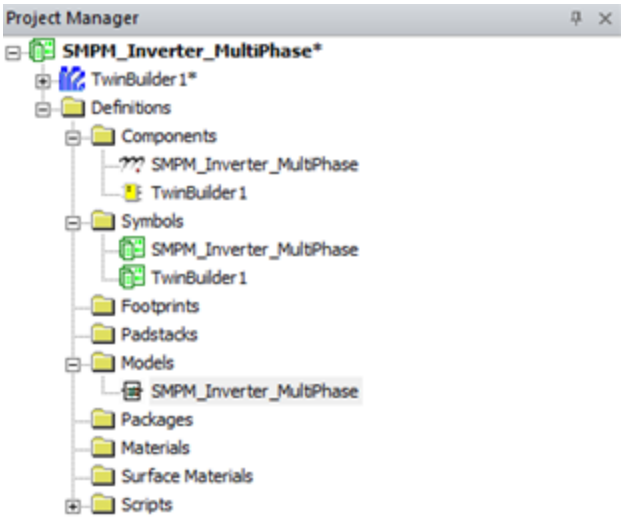
Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Because the entireSMPM Inverter MultiPhase system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

Next, you will define outputs and create reports for simulation results.

Defining Outputs and Creating Reports for the SMPM Inverter MultiPhase Design

When Modelica SMPM Inverter MultiPhase component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.

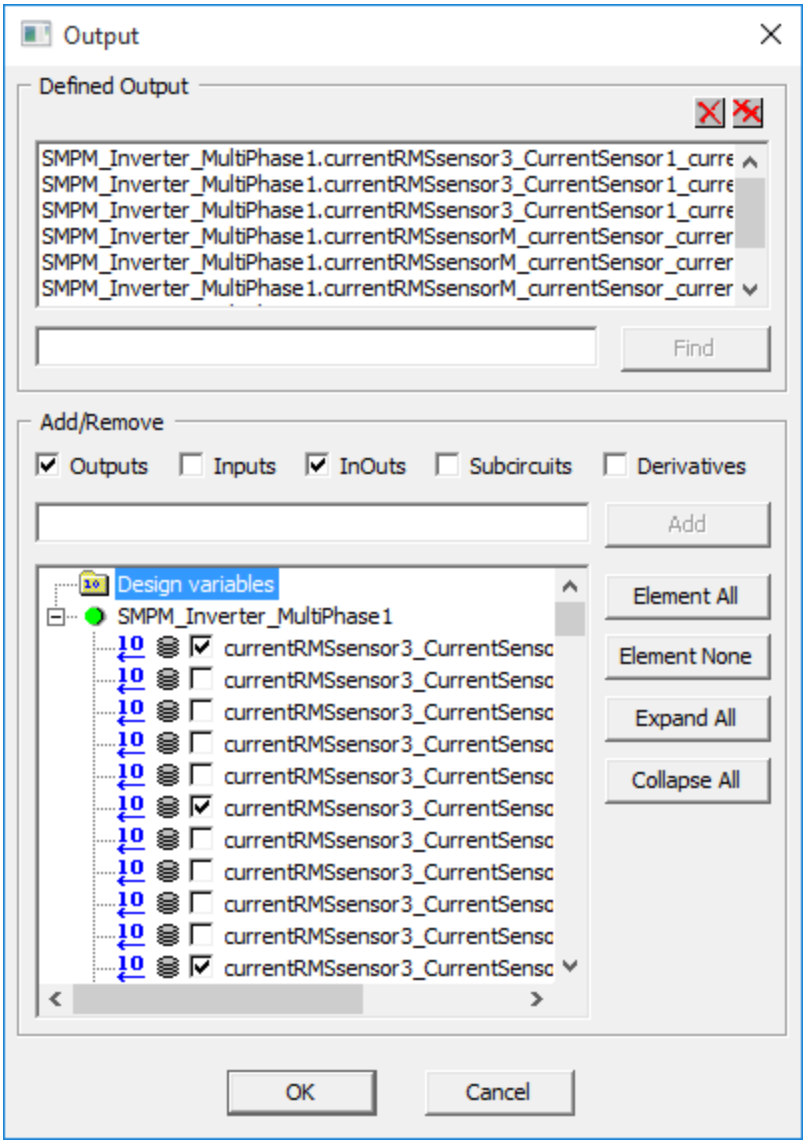


1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog box to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, the following outputs are selected as shown below:

    - SMPM_Inverter_MultiPhase1.currentRMSensor3_CurrentSensor1_currentSensor_1_i
    - SMPM_Inverter_MultiPhase1.currentRMSensor3_CurrentSensor1_currentSensor_2_i
    - SMPM_Inverter_MultiPhase1.currentRMSensor3_CurrentSensor1_currentSensor_3_i
    - SMPM_Inverter_MultiPhase1.currentRMSensorM_CurrentSensor1_currentSensor_1_i
    - SMPM_Inverter_MultiPhase1.currentRMSensorM_CurrentSensor1_currentSensor_2_i
    - SMPM_Inverter_MultiPhase1.currentRMSensorM_CurrentSensor1_currentSensor_3_i

- SMPM_Inverter_MultiPhase1.currentRMSensorM_CurrentSensor1_currentSensor_4_i
- SMPM_Inverter_MultiPhase1.currentRMSensorM_CurrentSensor1_currentSensor_5_i

See *Setting the Outputs for Simulation* in the Twin Builder help for more information.



2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

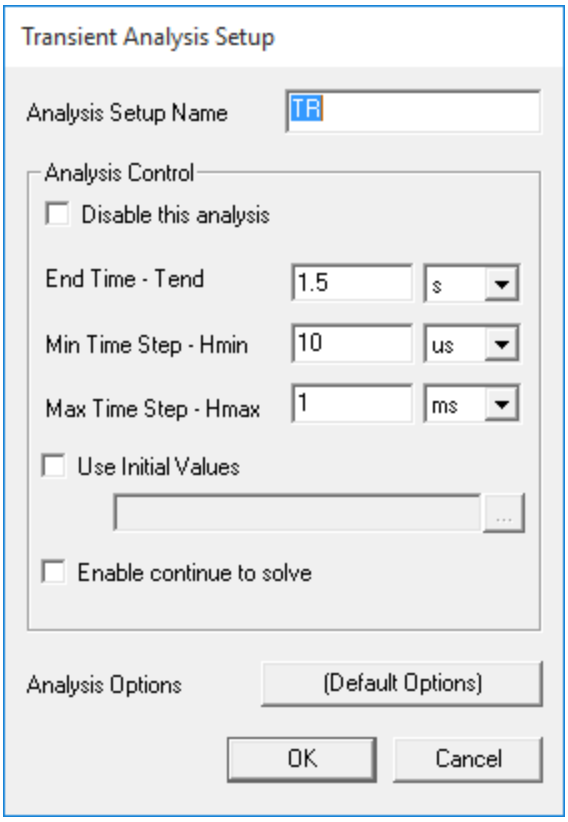
The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate SMPM Inverter MultiPhase simulation results.

Setting Up and Running a Transient Analysis for the SMPM Inverter MultiPhase Design

In this example, we would like to simulate the time-domain response of the SMPM Inverter MultiPhase system. This requires that a transient analysis needs to be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

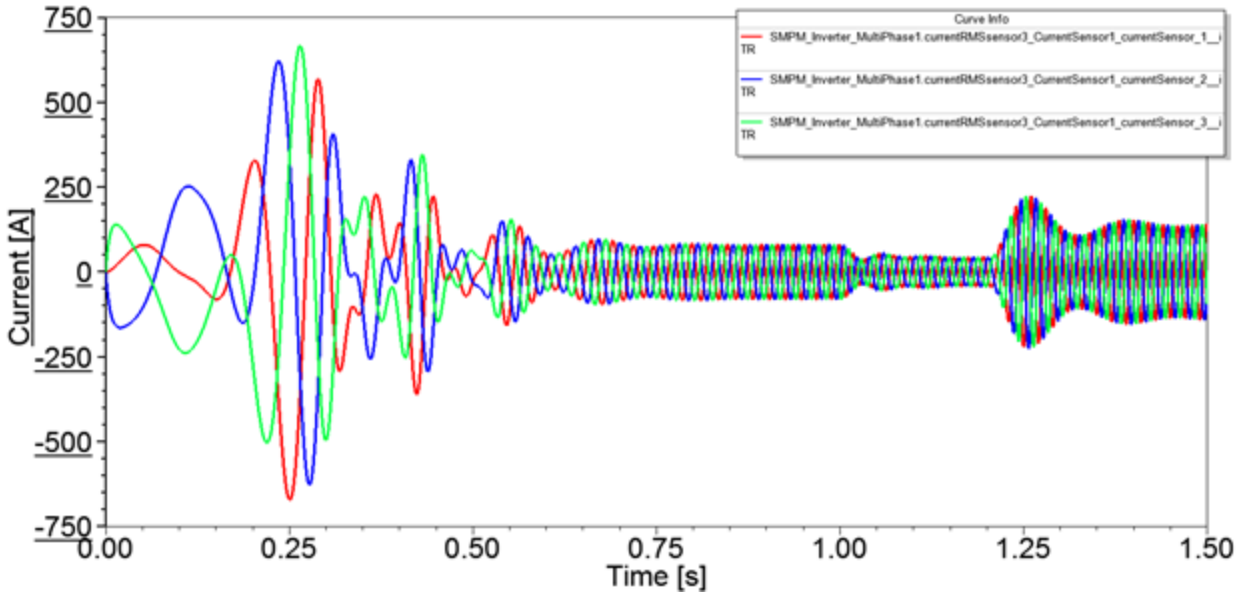
1. Double-click on **TR** to set up end time and the desired time step for the simulation.

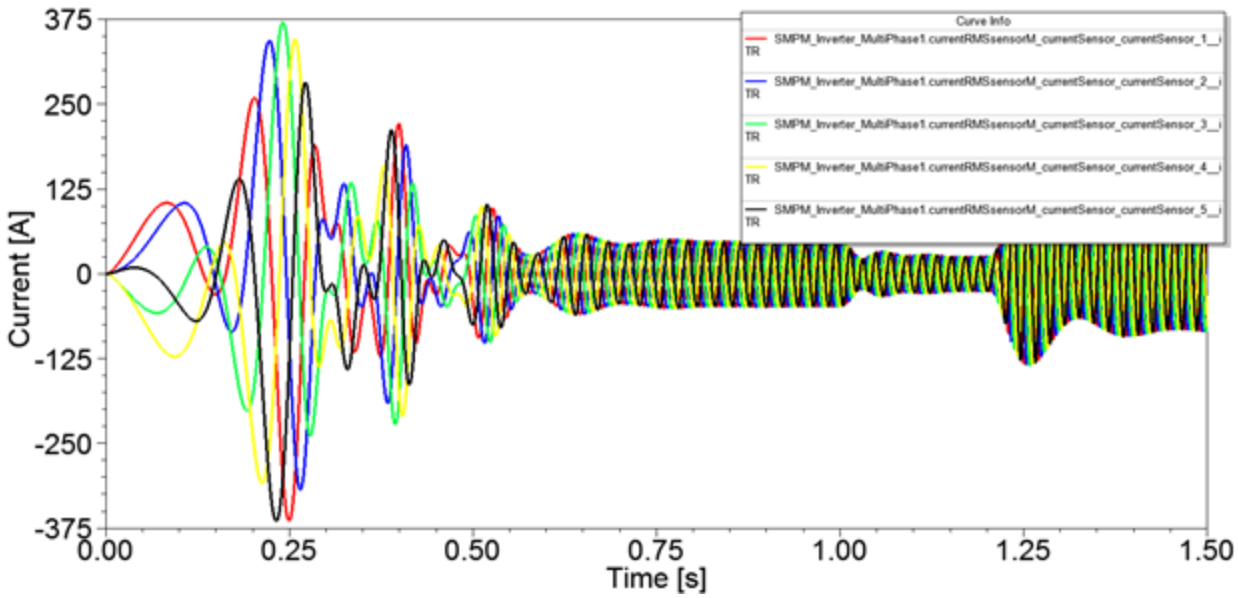


2. For this example, set **End Time** to 1.5 s, **Min Time Step** to 10 us, and keep **Max Time Step** as the default 1 ms. For more details about transient analysis, see *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:

- On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
- In the Project Manager, right click on **TR** of the design, then select **Analyze**.

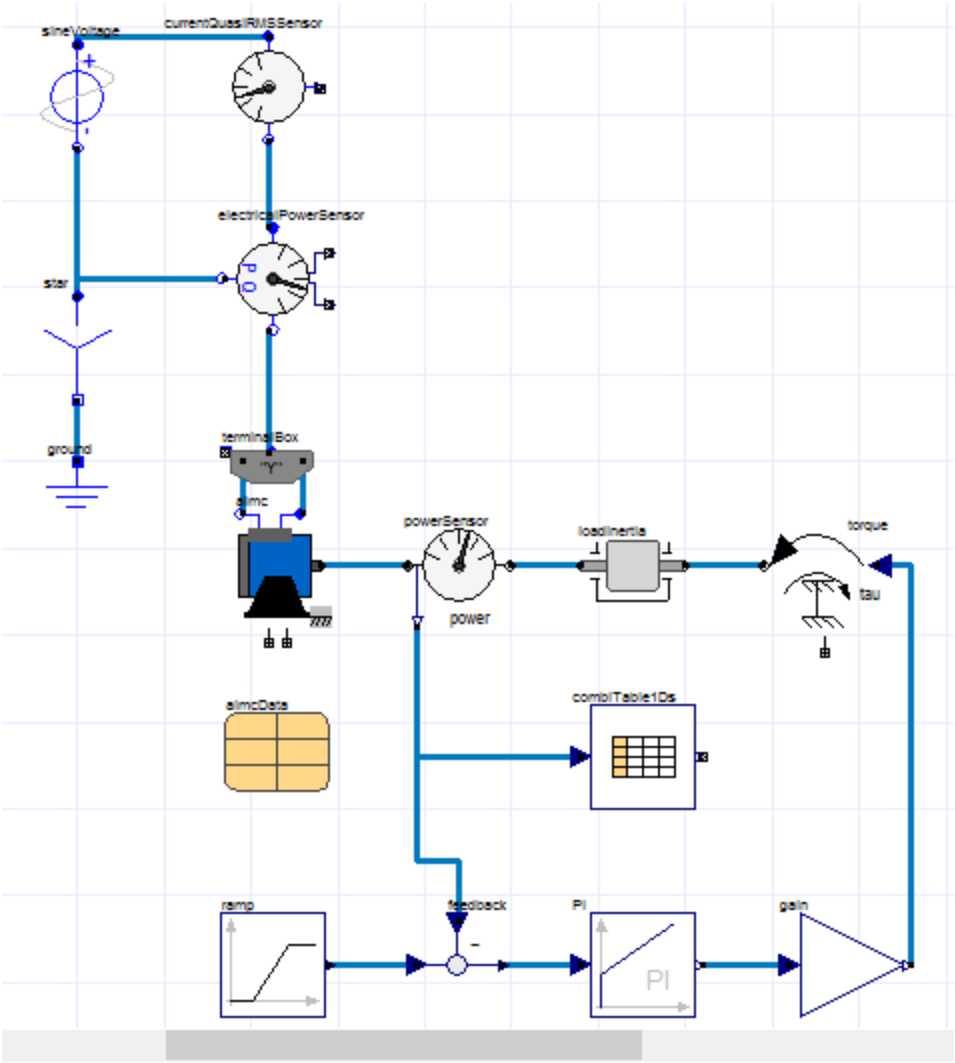
Below are sample reports of the simulation results for both 3-phase and 5-phase currents.





Electrical Example: AIMC with Losses

This example demonstrates an asynchronous squirrel cage induction machine and its characteristics, including losses.



Machine parameters are those for a standard 18.5 kW, 400 V, 50 Hz motor. For the first 5 seconds of simulation time, the machine is started at a nominal speed and flux is built up in the machine. Then, for the next 5 seconds, a load ramp is applied.

This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\AIMC.aedt

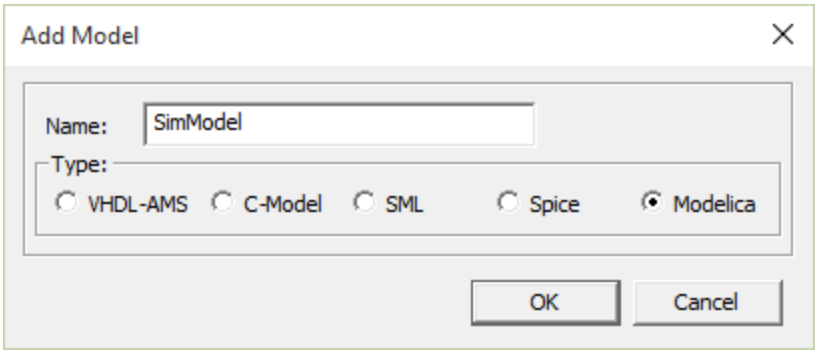
AIMC Example Workflow

In this example, we will create a Modelica AIMC (asynchronous squirrel cage induction machine with losses) model and analyze its performance in the Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the AIMC model into the **Diagram Editor**.
- Connect the Modelica AIMC components.
- Provide initial equations and variable declarations.
- Assign the various component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Define outputs and Create Reports for the AIMC simulation results.
- Set up and run a Transient analysis for the AIMC design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > **New** to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename** from the context menu.
2. Create a new Modelica model by doing one of the following:
   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click on the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

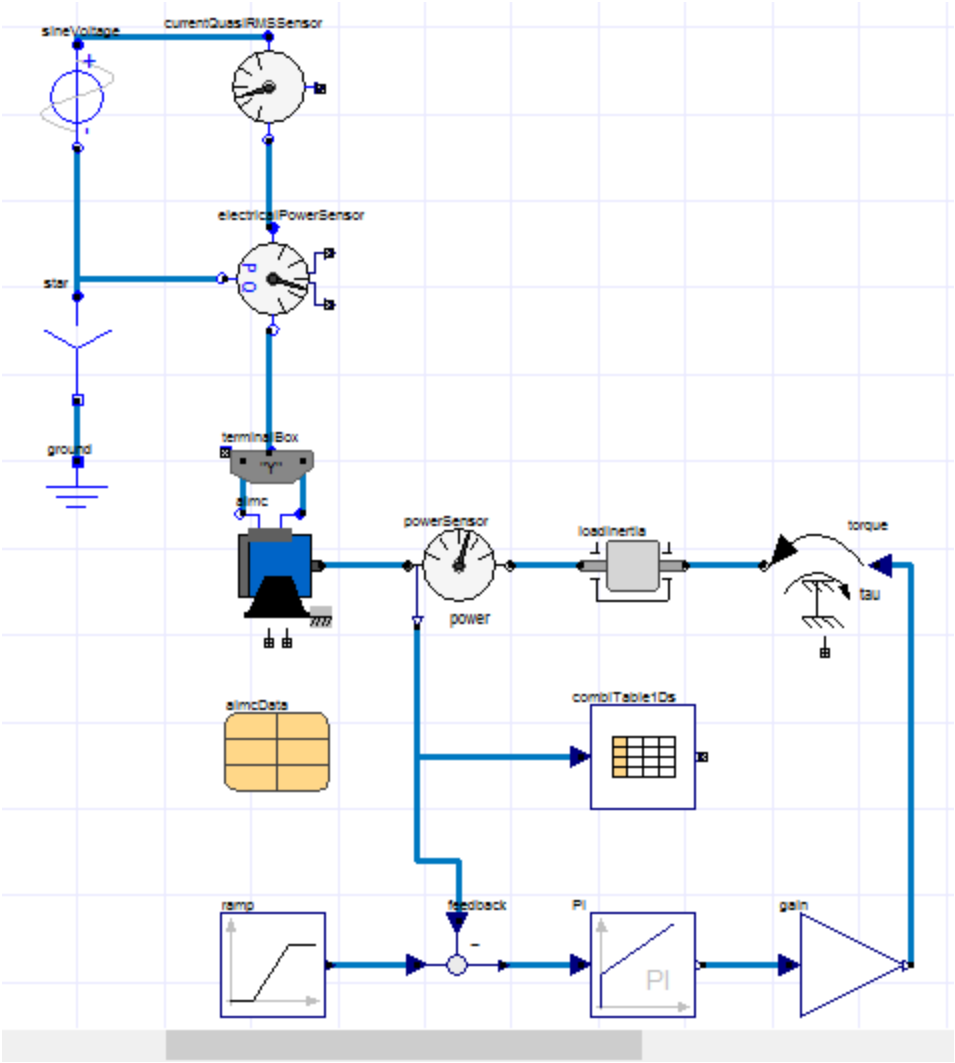
Next, you will place the Modelica components that are needed for the new model into the **Diagram Editor**.

Placing Modelica Components for the AIMC Model

All of the required components for the AIMC example can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
terminalBox	Modelica/Electrical/Machines/Utilities/TerminalBox
aimc	Modelica/Electrical/Machines/BasicMachines/AsynchronousInduction Machines/AIM_SquirrelCage
powerSensor	Modelica/Mechanics/Rotational/Sensors/PowerSensor
loadInertia	Modelica/Mechanics/Rotational/Components/Inertia
torque	Modelica/Mechanics/Rotational/Sources/Torque
aimcData	Modelica/Electrical/Machines/Utilities/ParameterRecords/AIM_ SquirrelCageData
combiTable1Ds	Modelica/Blocks/Tables/CombiTable1Ds
ramp	Modelica/Blocks/Sources/Ramp
feedback	Modelica/Blocks/Math/Feedback
PI	Modelica/Blocks/Continuous/PI
gain	Modelica/Blocks/Math/Gain
star	Modelica.Electrical/MultiPhase/Basic/Star
ground	Modelica/Electrical/Analog/Basic/Ground
sineVoltage	Modelica/Electrical/MultiPhase/Sources/SineVoltage
currentQuasiRMSSensor	Modelica/Electrical/Machines/Sensors/CurrentQuasiRMSSensor
electricalPowerSensor	Modelica/Electrical/Machines/Sensors/ElectricalPowerSensor

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press **Esc** after placing each component to stop the placement action. Connections will be added later.

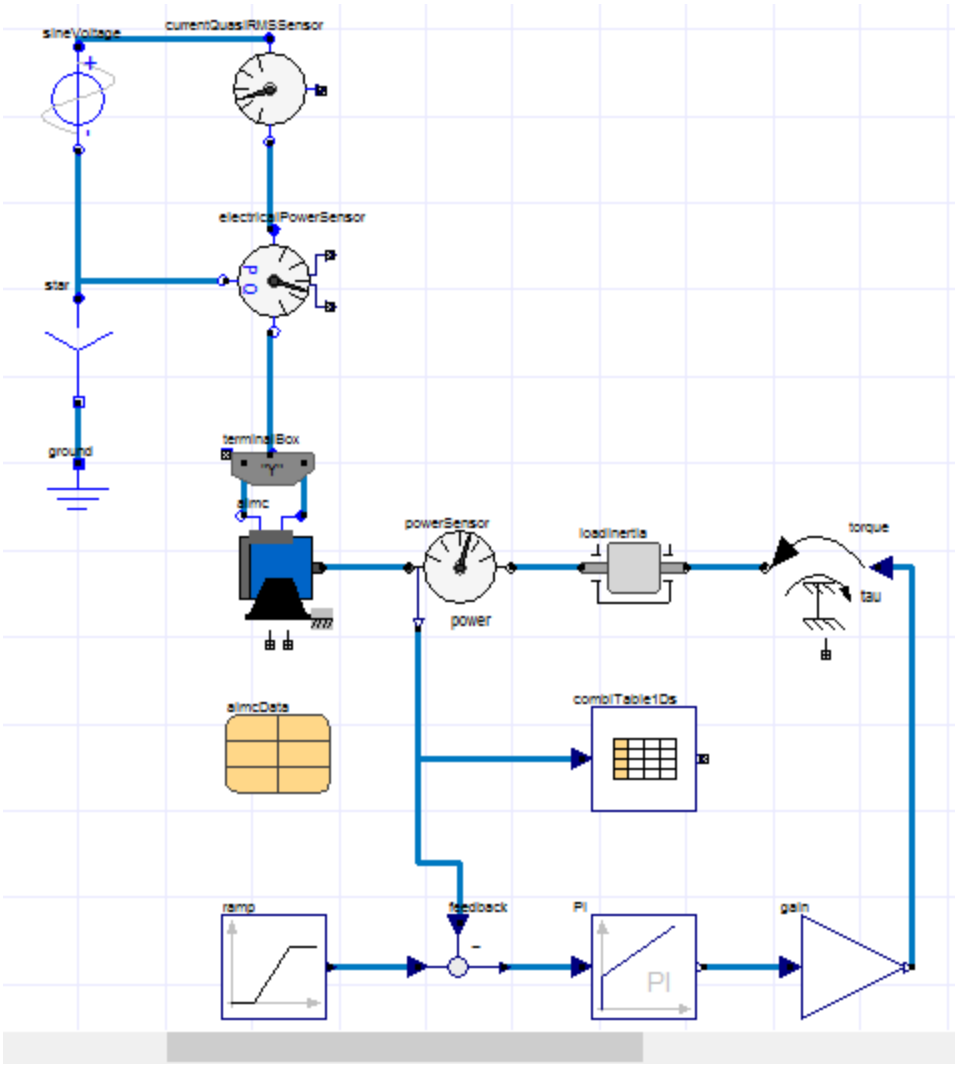


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press **Esc** to complete each action.

Next, you will add connections between the various AIMC components.

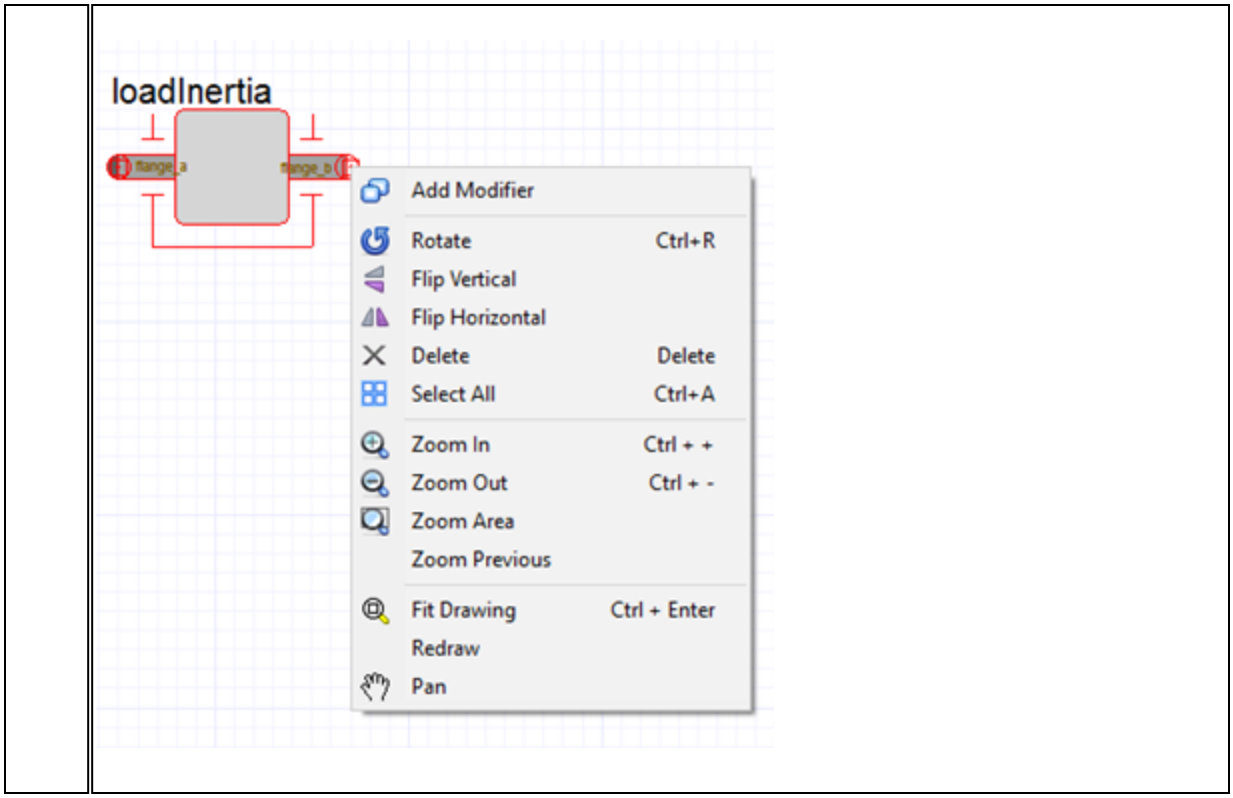
Connecting Modelica AIMC Components

To connect the components for the AIMC example as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will add an initial equation and declare a variable for the AIMC design.

Adding an Initial Equation and Variable Declaration to the AIMC Design

For this example, three initial equations are needed for the AIMC machine, and several variables declared.

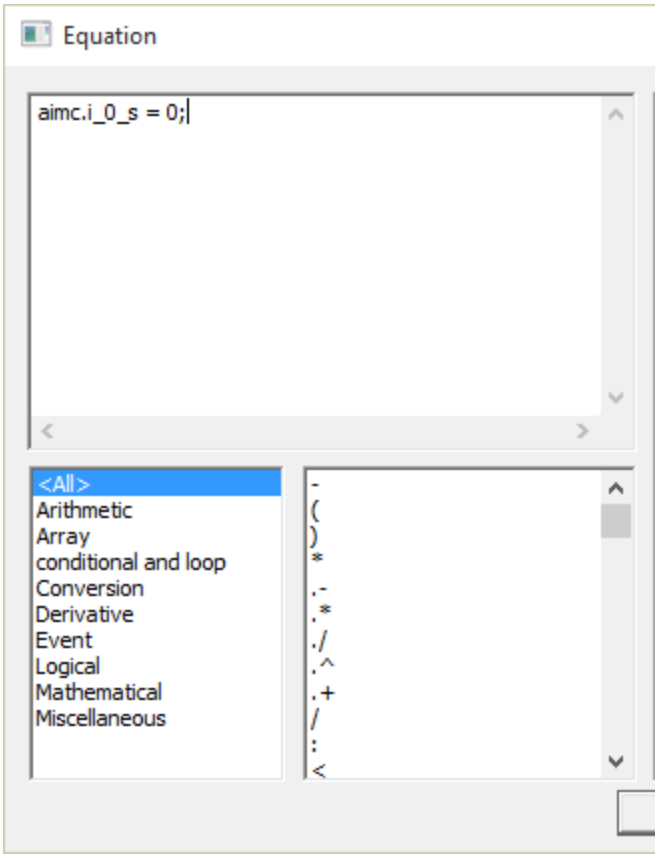
Initial Equation

1. For the initial equations, on the Twin Builder **Modelica** tab, click the Equations icon to open the **Edit Model** dialog box.
2. Select **Initial Equation**, then click the button in the upper right of the dialog box to open the **Equation** editor, and enter the following equations:

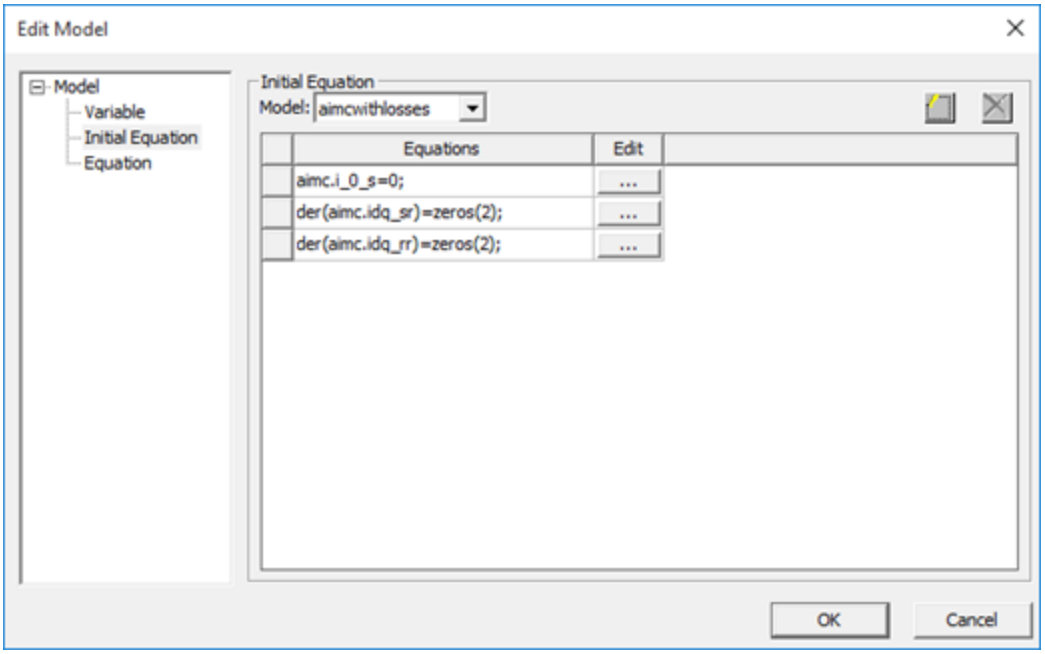
   aimc.i_0_s = 0;

   der(aimc.idq_sr) = zeros(2);

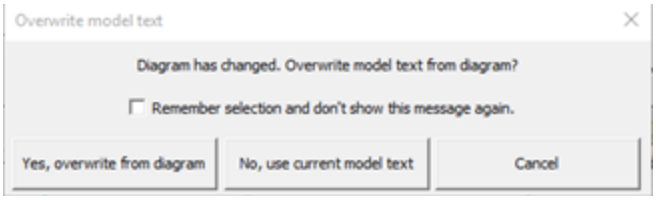
   der(aimc.idq_rr) = zeros(2);



3. Click **OK** to close the dialog box, adding the equation to the **Edit Model** dialog box.
4. Repeat for each of the equations.



5. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog box appears.



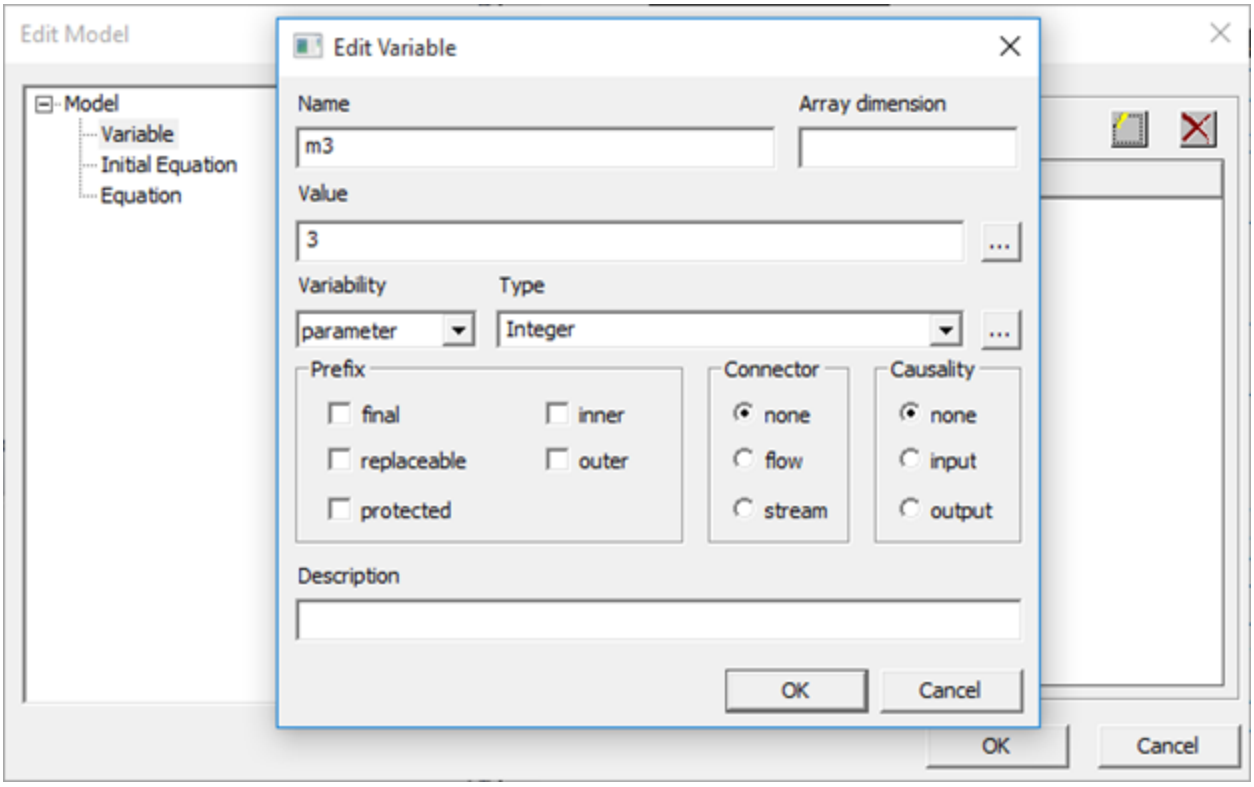
6. Click **Yes, overwrite from diagram** to generate the equation and add it to the model text.

```
initial equation
    aimc.i_0_s = 0;
    der(aimc.idq_sr) = zeros(2);
    der(aimc.idq_rr) = zeros(2);
```

Variable Declarations

1. To create the needed variables for this design, on the Twin Builder **Modelica** tab, click the **Variables** icon to open the **Edit Model** dialog box.

2. Click the button in the upper right of the dialog to open the **Edit Variable** dialog box, and enter the settings such as for the integer parameter **m3=3** shown below. The Prefix, Connector, and Causality settings remain the defaults as shown.



3. Click **OK** to close the dialog box, adding the variable to the **Edit Model** dialog box.
4. Repeat, adding the variables listed in the following table:

Variable Name	Value	Variability	Type	Array dimension
m	3	parameter	real	
PNominal	18500	parameter	real	
VNominal	400	parameter	real	
INominal	32.85	parameter	real	

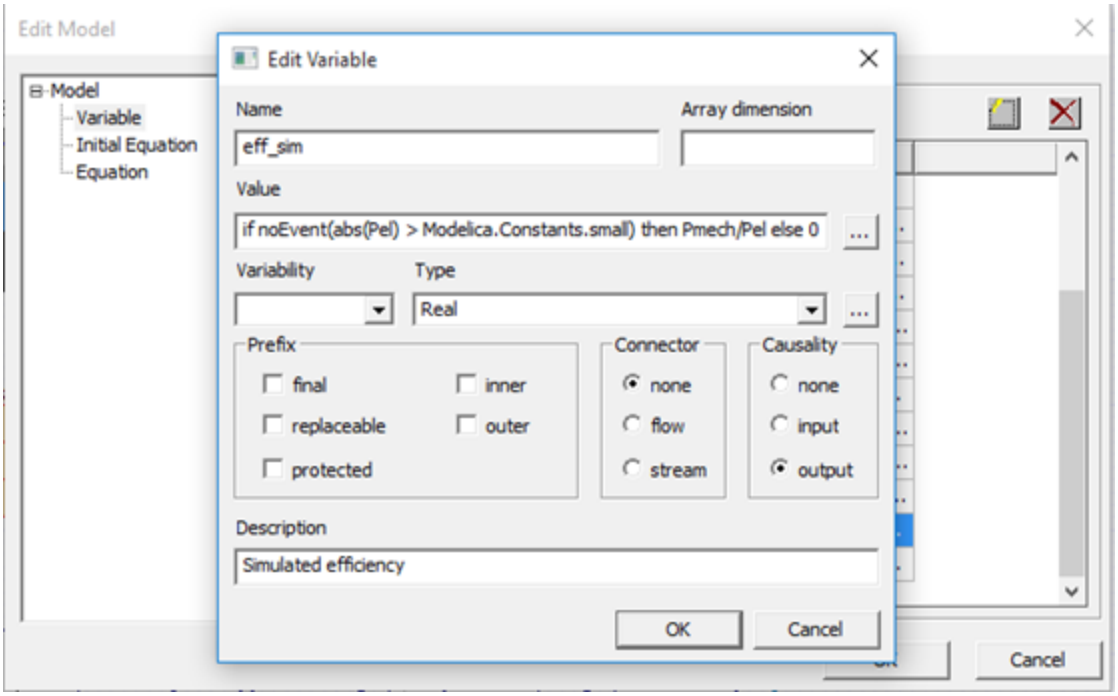
Variable Name	Value	Variability	Type	Array dimension
pfNominal	0.898	parameter	real	
fNominal	50	parameter	real	
wNominal	Modelica.SIunits.Conversions.from_rpm (1462.5)	parameter	real	
TNominal	PNominal/wNominal	parameter	real	
TempNominal	Modelica.SIunits.Conversions.from_degC(90)	parameter	real	
Ptable	{1E-6,1845,3549,5325,7521,9372,11010,12930, 14950,16360,18500,18560,20180,22170}	parameter	real	:
Itable	{11.0,11.20,12.27,13.87,16.41,18.78,21.07,23.92, 27.05,29.40,32.85,32.95,35.92,39.35}	parameter	real	:
ntable	{1500,1496,1493,1490,1486,1482,1479,1475, 1471, 1467,1462,1462,1458,1453}	parameter	real	:
ctable	{0.085,0.327,0.506,0.636,0.741,0.797,0.831,0.857, 0.875,0.887,0.896,0.896,0.902,0.906}	parameter	real	:
etable	{0,0.7250,0.8268,0.8698,0.8929,0.9028,0.9064,0.9088, 0.9089,0.9070,0.9044,0.9043,0.9008,0.8972}	parameter	real	:

Note that for array parameters such as **Ptable**, you must specify **:** in the **Array dimension** field so that the size can adapt to user input.

5. Outputs can also be declared as a special type of variable by setting **Causality** to **output**. For this example, declare the following special variables:

Variable Name	Value	Variability	Type	Prefix/Causality
Pel	electricalPowerSensor.P		Modelica.SIunits.Power	none
Pmech	powerSensor.power		Modelica.SIunits.Power	output
Eff_sim	if noEvent(abs(Pel) > Modelica.Constants.smal		real	output

Variable Name	Value	Variability	Type	Prefix/Causality
	l) then Pmech/Pel else 0			



6. Click **OK** to close the **Edit Model** dialog box. The **Overwrite Model Text** dialog box appears.
7. Click **Yes, overwrite from diagram** to add the variable declarations to the model text.

```
//Declaration(s)
parameter Real m = 3;
parameter Real PNominal = 18500;
parameter Real VNominal = 400;
parameter Real INominal = 32.85;
parameter Real pfNominal = 0.898;
parameter Real fNominal = 50;
parameter Real wNominal = Modelica.SIunits.Conversions.from_rpm(1462.5);
parameter Real TNominal = PNominal/wNominal;
parameter Real TempNominal = Modelica.SIunits.Conversions.from_degC(90);
parameter Real Ptable[:]  = {1E-6,1845,3549,5325,7521,9372,11010,12930,
  14950,16360,18500,18560,20180,22170};
parameter Real Itable[:]  = {11.0,11.20,12.27,13.87,16.41,18.78,21.07,
  23.92,27.05,29.40,32.85,32.95,35.92,39.35};
parameter Real ntable[:]  = {1500,1496,1493,1490,1486,1482,1479,1475,1471,
  1467,1462,1462,1458,1453};
parameter Real ctable[:]  = {0.085,0.327,0.506,0.636,0.741,0.797,0.831,
  0.857,0.875,0.887,0.896,0.896,0.902,0.906};
parameter Real etable[:]  = {0,0.7250,0.8268,0.8698,0.8929,0.9028,0.9064,
  0.9088,0.9089,0.9070,0.9044,0.9043,0.9008,0.8972};
output Real eff_sim = if noEvent(abs(Pel) > Modelica.Constants.small) then Pmech/Pel else 0;
Modelica.SIunits.Power Pel = electricalPowerSensor.P;
output Modelica.SIunits.Power Pmech = powerSensor.power;
```

Next, you will set up properties for the AIMC model.

Setting Up Model Properties for the AIMC Example

The model properties for the components used in this example are summarized in the following table.

Component Instance	Properties
terminalBox	terminalConnection = “D”
aimc	Lm = aimcData.Lm Lrsigma = aimcData.Lrsigma Rr = aimcData.Rr TrRef = aimcData.TrRef alpha20r = aimcData.alpha20r TrOperational = TempNominal p = aimcData.p fsNominal = aimcData.fsNominal TsOperational = TempNominal

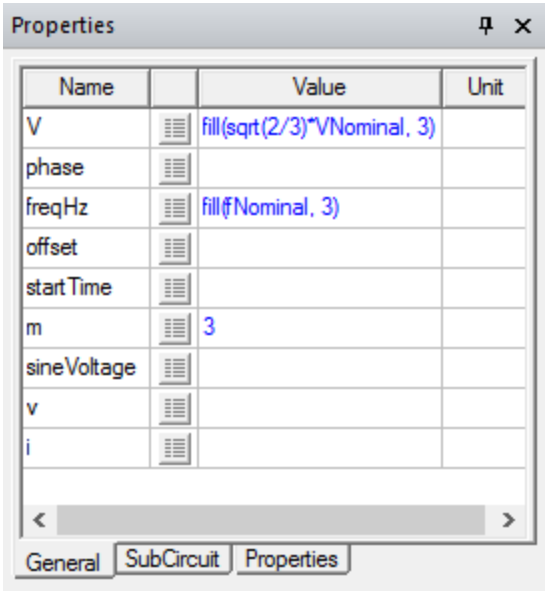
Component Instance	Properties
	Rs = aimcData.Rs TsRef = aimcData.TsRef alpha20s = aimcData.alpha20s Lszero = aimcData.Lszero Lssigma = aimcData.Lssigma Jr = aimcData.Jr useSupport = false Js = aimcData.Js useThermalPort = false statorCoreParameters = aimcData.statorCoreParameters strayLoadParameters = aimcData.strayLoadParameters frictionParameters = aimcData.frictionParameters **phiMechanical**(fixed = true) **wMechanical**(fixed = true, start = 2*Modelica.Constants.pi*aimcData.fsNominal/aimcData.p) **NOTE:** For the **phiMechanical**and **wMechanical** properties, refer to the "Modification of variables ir, phiMechanical and wMechanical " on page 3-93 step below.
powerSensor	Use defaults
loadInertia	J = aimcData.Jr stateSelect = stateSelect.defalut
torque	Use defaults
aimcData	Jr = 0.12 Lm = 66.4/(2*Modelica.Constants.pi*fNominal) Lrsigma = 2.31/(2*Modelica.Constants.pi*fNominal) Rr = 0.42 Rs = 0.56 alpha20r = Modelica.Electrical.Machines.Thermal.Constants.alpha20Aluminium

Component Instance	Properties
	alpha20r = Modelica.Electrical.Machines.Thermal.Constants.alpha20Copper Lssigma = 1.52/(2*Modelica.Constants.pi*fNominal) frictionParameters = frictionParameters(PRef=180, wRef=wNominal) statorCoreParameters = statorCoreParameters(PRef=410, VRef=387.9) strayLoadParameters = strayLoadParameters(PRef=0.005*sqrt(3)*VNominal*INominal*pfNominal, IRef=INominal/sqrt(3), wRef=wNominal)
combiTable1Ds	verboseRead = false columns = 2:size(combiTable1Ds.table, 2) table = {{Ptable[j],Itable[j],ntable[j],ctable[j],etable[j]} for j in 1:size(Ptable, 1)} smoothness = Modelica.Blocks.Types.Smoothness.ContinuousDerivative
ramp	height = 1.2*PNominal duration = 5.5 startTime = 4.5
feedback	Use defaults
PI	k = 0.01 T = 0.01 initType = Modelica.Blocks.Types.Init.InitialState
gain	k = -1
star	Use defaults
ground	Use defaults
sineVoltage	V = fill(sqrt(2/3)*VNominal, 3) freqHz = fill(fNominal, 3) m =3
currentQuasiRMSSensor	Use defaults
electricalPowerSensor	m =3

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options from the table above as needed. For example, to set properties for the **sineVoltage** component in the Diagram Editor:

    a. Click on the **sineVoltage** component to display its **Properties** window.



    b. On the **General** tab, per the information in the above table, enter **3** for the **m** property; **fill(fNominal, 3)** for **freqHz**, and **fill(sqrt(2/3)*VNominal, 3)** for **V**. No other changes are needed. The **Properties** tab allows you to set the name of the component instance.

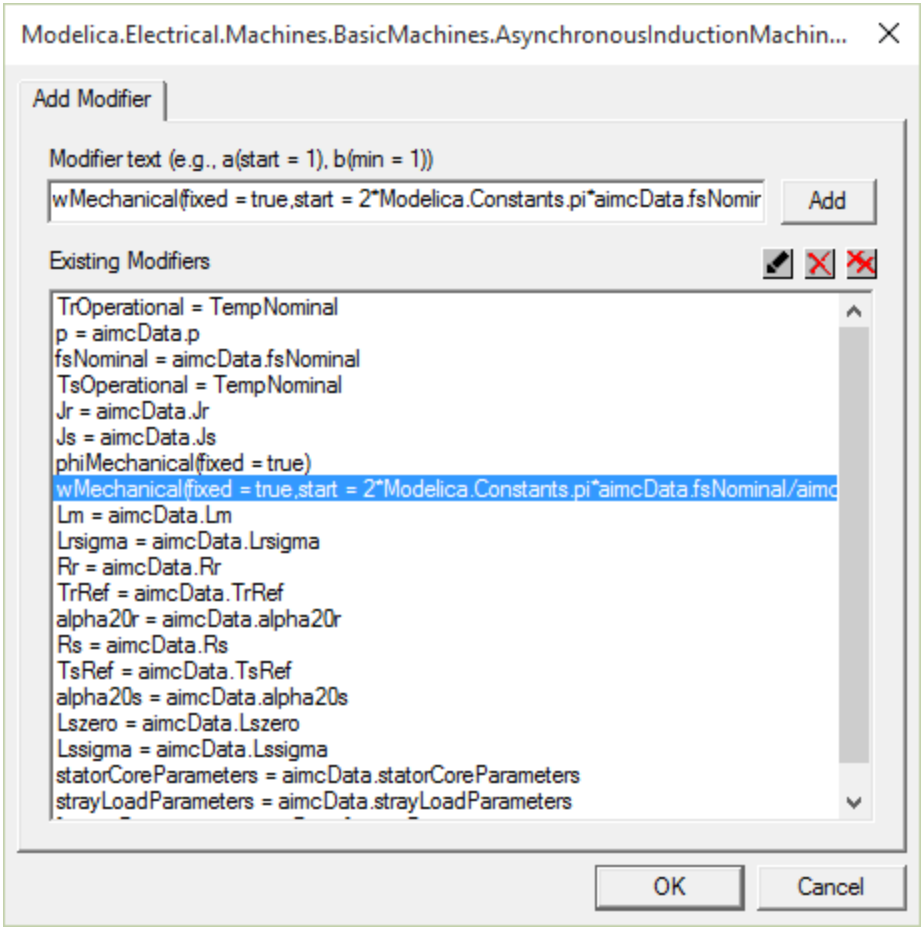
Modification of variables ir, phiMechanical and wMechanical

This example requires modification of variables **phiMechanical** and **wMechanical** for the **aimc** machine. This cannot be done directly in the **Properties** window, but instead requires that modifiers be added.

1. To add modifiers to **aimc**, select the **aimc** instance in diagram editor and on the ribbon Modelica tab click **Add Modifier** (or right click on the **aimc** instance and select **Add Modifier** from the context menu). In the **Add Modifier** dialog box, add the following three lines:

```
wMechanical(fixed=true,
```

```
start=2*Modelica.Constants.pi*aimcData.fsNominal/aimcDat
a.p)

phiMechanical(fixed = true)
```

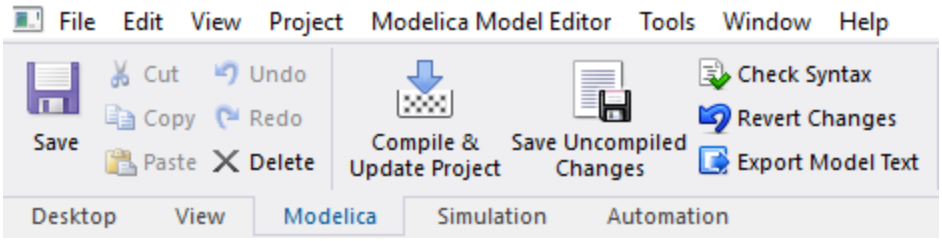


Next, you will compile and update the project.

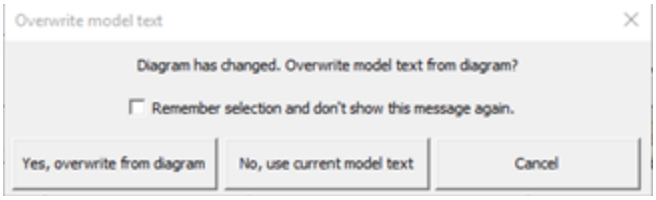
Compiling and Updating the AIMC Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.

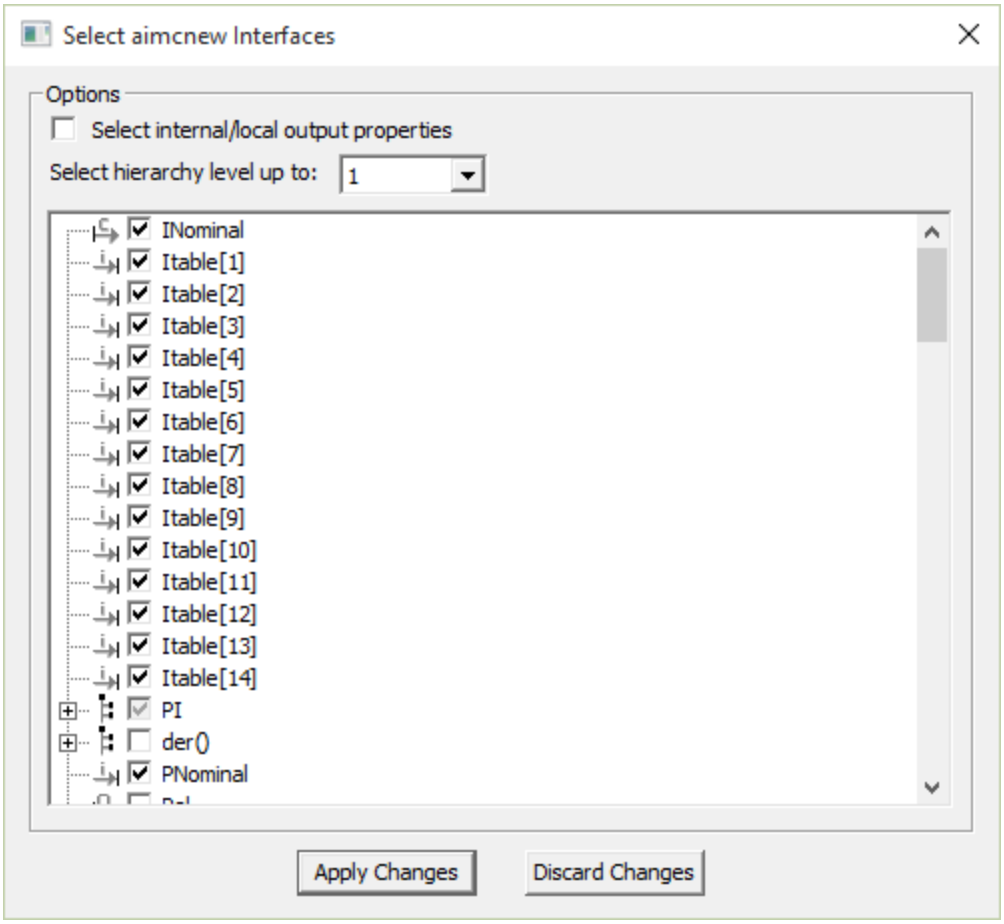


The **Overwrite model text** dialog box appears.

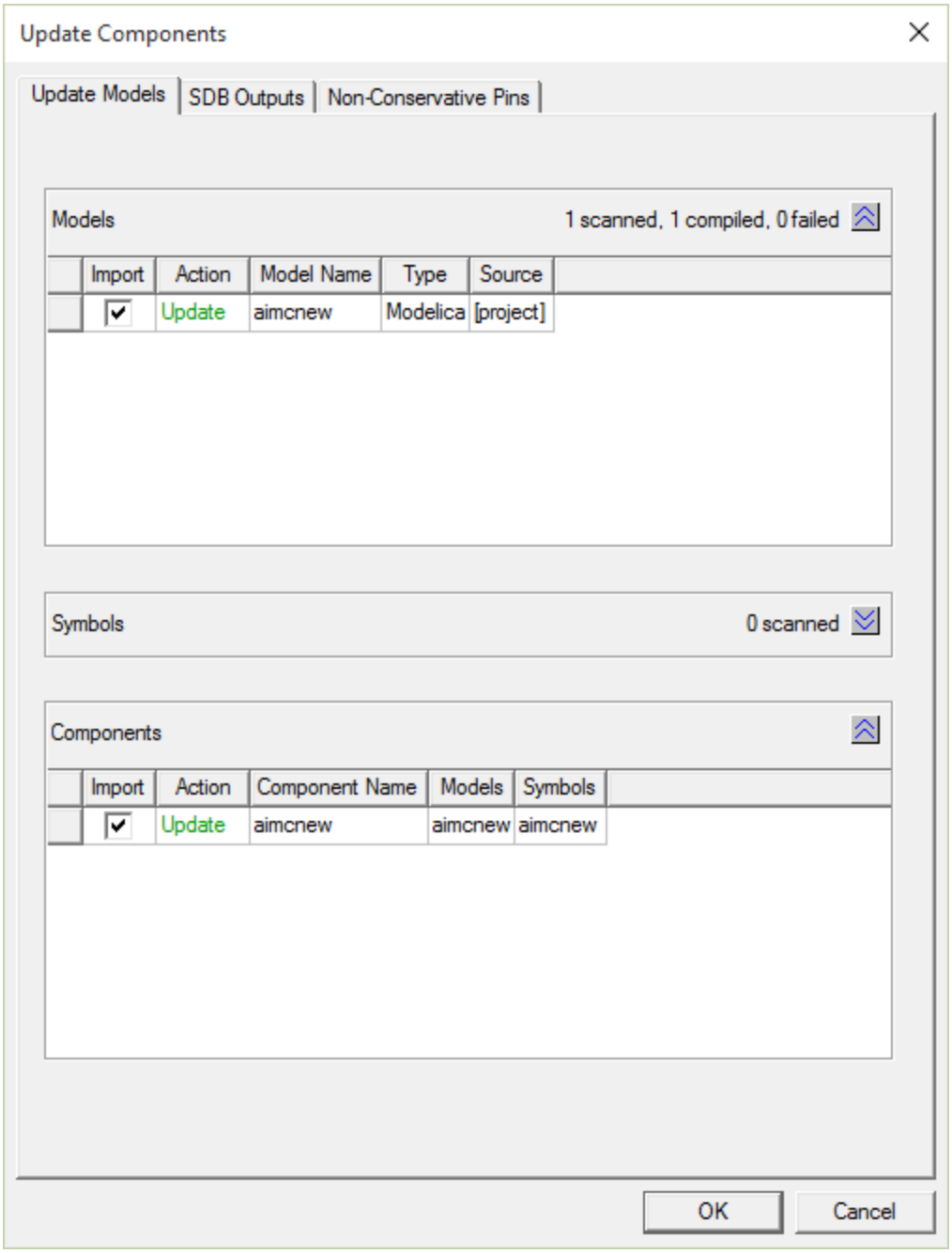


Note	This same dialog is also displayed whenever you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For the AIMC example, all the editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. This may take several seconds.
3. Following the compile operation, you can select interfaces for the AIMC model, or just accept the default set of preselected interfaces.

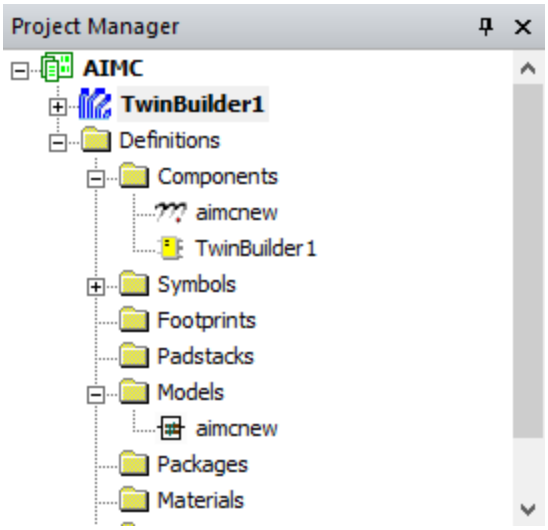


4. When finished selecting interfaces, click **Apply Changes** to close the dialog box and apply the changes to the AIMC model. The **Update Components** dialog box appears with both the AIMC model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the AIMC model and component into Twin Builder.

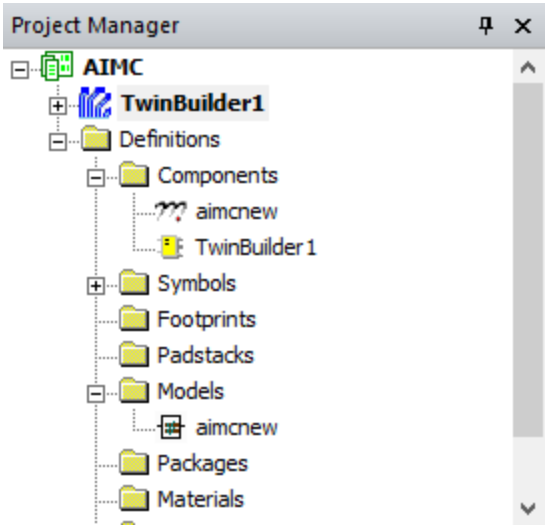
The Modelica AIMC component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



Next, you will add the component to a Twin Builder schematic.

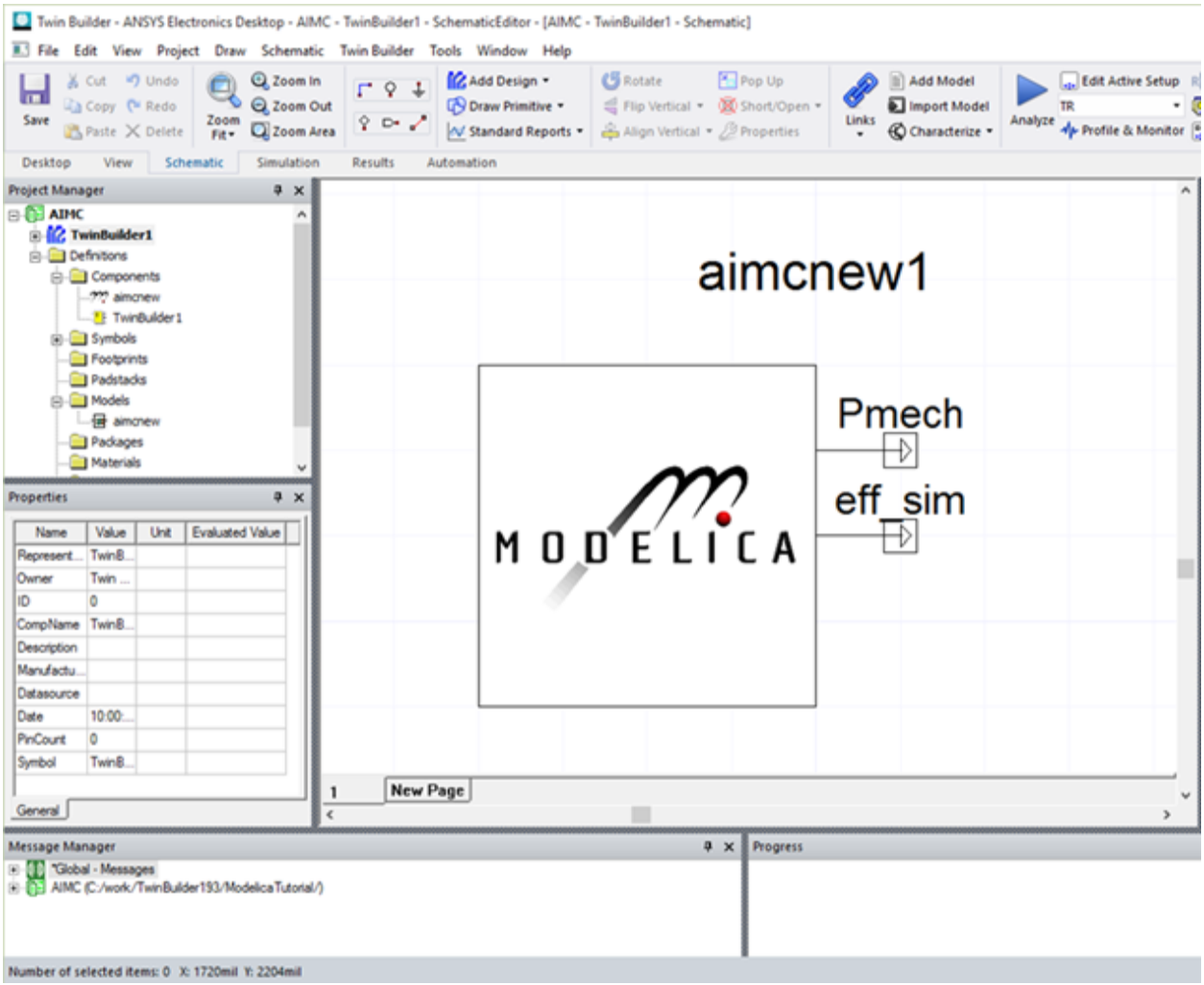
Adding the AIMC Component to a Twin Builder Schematic

When the Modelica AIMC component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.

2. Locate the Modelica AIMC component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design schematic, as shown below.



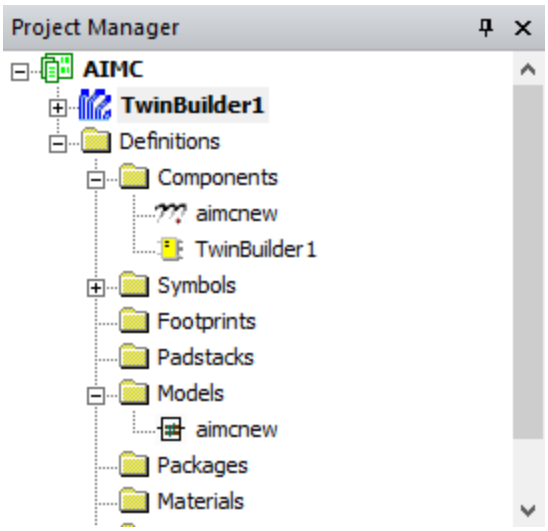
Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Because the entire AIMC system is built inside the Modelica component, the Twin Builder design is now complete and ready to analyze.

Next, you will define outputs and create reports for simulation results.

Defining Outputs and Creating Reports for the AIMC Design

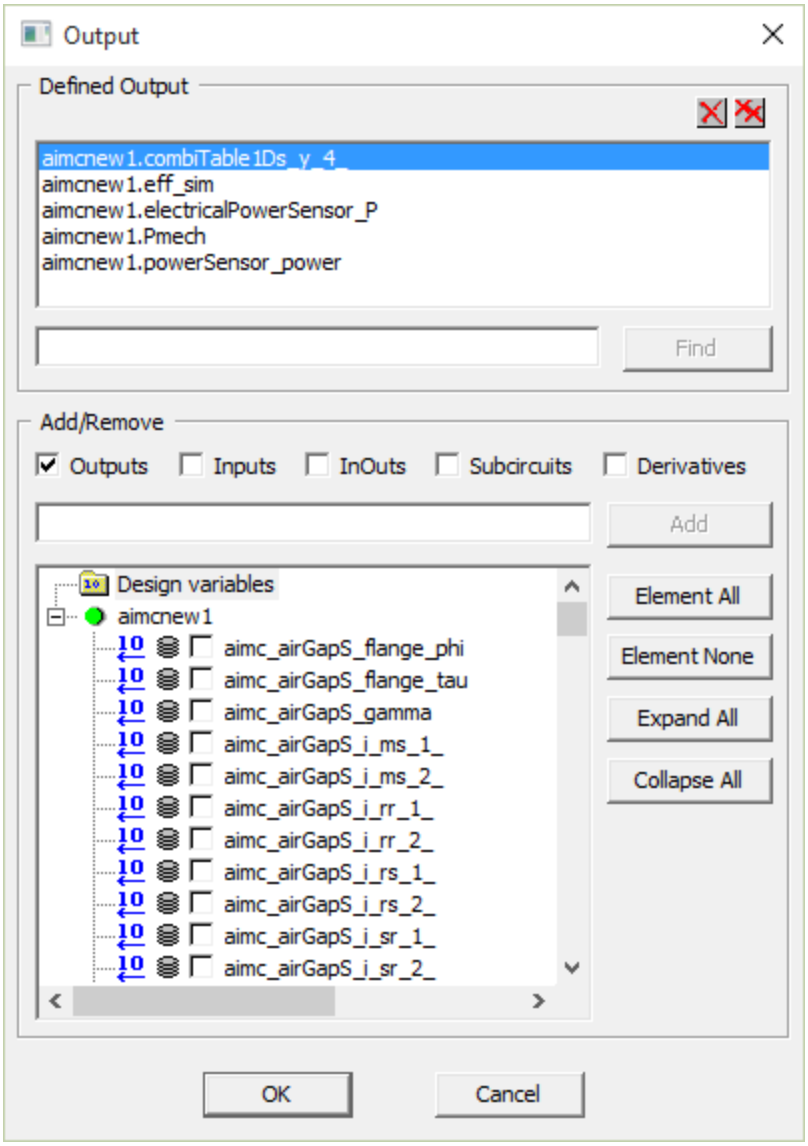
When Modelica AIMC component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog box to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, the following outputs are selected as shown below:

   - aimcnew1.combiTable1Ds_y_4_
   - aimcnew1.eff_sim
   - aimcnew1.electricalPowerSensor_P
   - aimcnew1.powerSensor_power
   - aimcnew1.Pmech

   Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.



2. In the Project Manager, you can then right-click **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

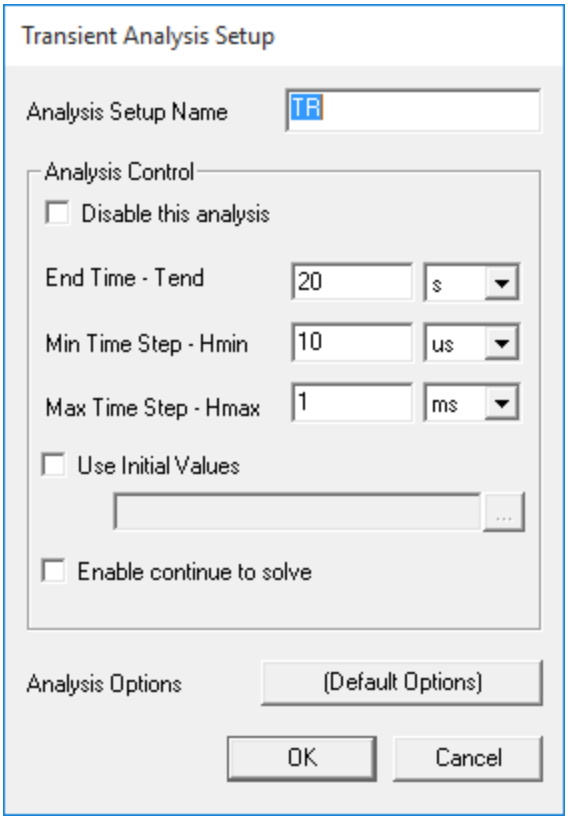
   The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate AIMC simulation results.

Setting Up and Running a Transient Analysis for the AIMC Design

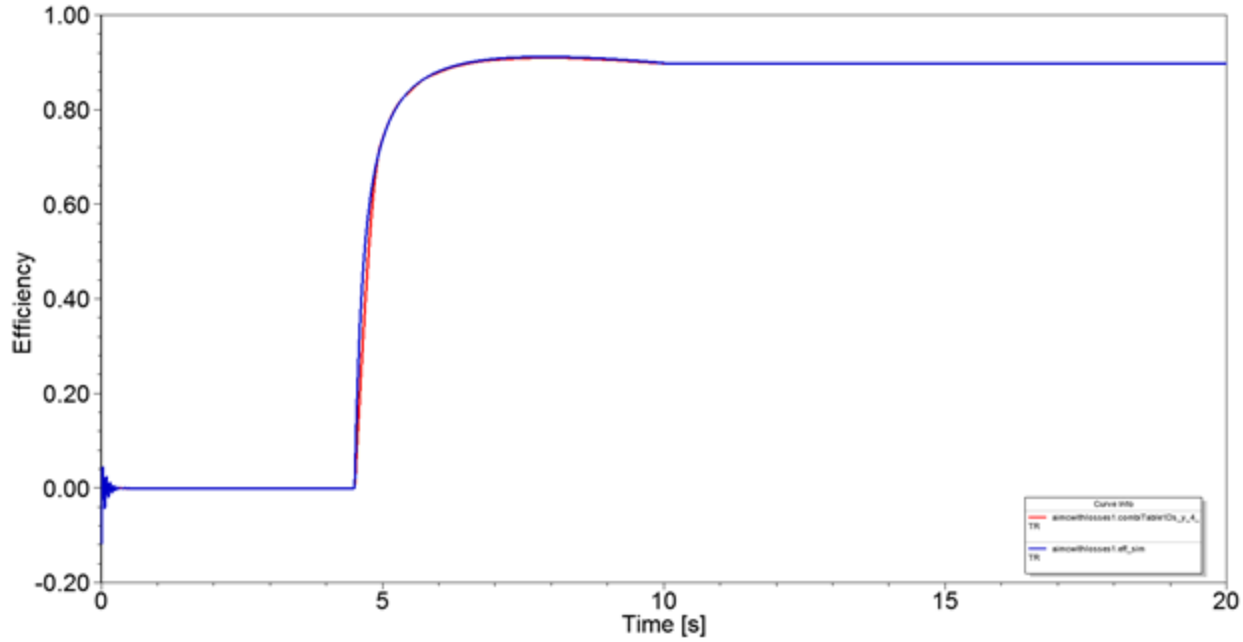
In this example, we would like to simulate the time-domain response of the AIMC system. This requires that a transient analysis needs to be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

1. Double-click on **TR** to set up end time and the desired time step for the simulation.



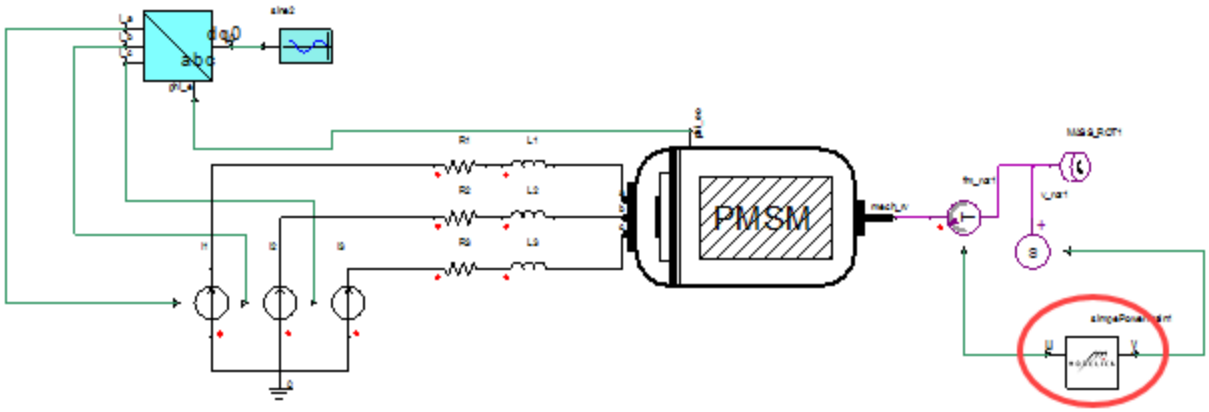
2. For this example, set **End Time** to 20 s, **Min Time Step** to 10 us, and keep **Max Time Step** as the default 1 ms. For more details about transient analysis, see *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
   - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
   - In the Project Manager, right-click on **TR** of the design, then select **Analyze**.

Below is a sample report of the simulation results for the design.



Example: Simple Power Train

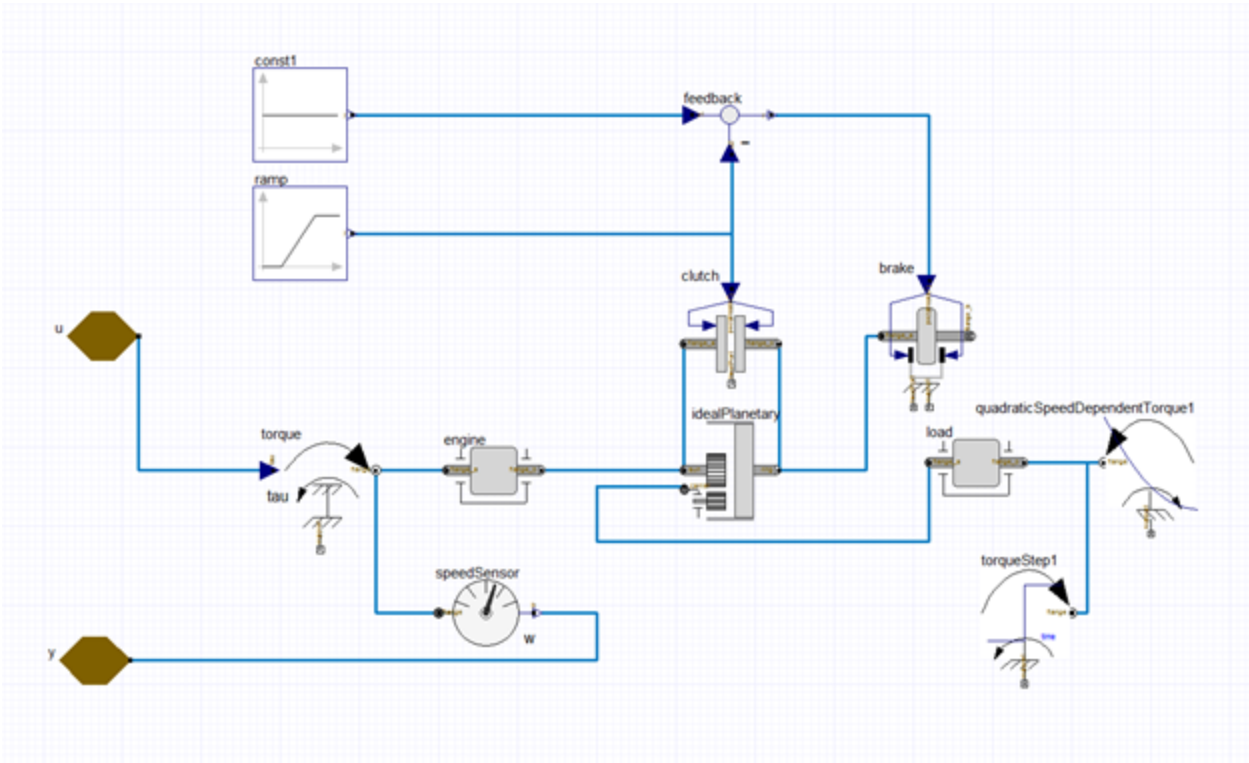
This example illustrates a Twin Builder design for a simple motor power train consisting of a PMSM (Permanent Magnet Synchronous Motor) and other components taken from the Twin Builder component libraries that are used to drive a load, which is a Modelica gear shift component that has been modeled in the Modelica environment.



In Twin Builder schematic shown above, the current sources, PMSM component, and all components (except the circled Modelica gear shift component) are from Twin Builder component libraries. The Modelica gear shift load is modeled in the Modelica environment.

The figure below shows the internal diagram of the gear shift Modelica model. It is based on the **Modelica.Mechanics.Rotational.Examples.SimpleGearShift** in the Modelica Standard Library. It shows how an automatic gear shift is built up from a planetary gear, a brake, a clutch, and related Modelica components. The basic operation of the model is as follows:

- In the beginning, the clutch is free (disengaged) and the brake prevents the ring of the planetary gear from moving. Thus the gearRatio = 1/(1 + planetary.ratio).
- At time = 2 s, the brake releases the ring of the planetary gear, while the clutch blocks the ring and the sun gear. Thus we obtain a gearRatio = 1.



This example is based on the following example project located in the Modelica Tutorial Examples folder:

<Installation location>\Examples\Twin Builder\Modelica Tutorial\Tutorial Examples\SimplePowerTrain.aedt

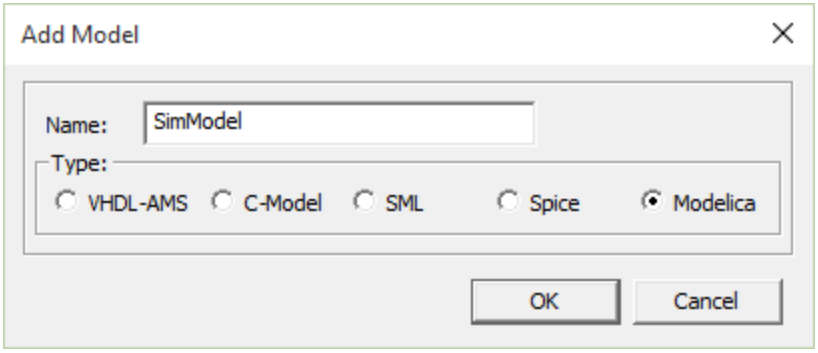
Simple Power Train Example Workflow

In this example, we will create a Modelica gear shift model, then add it as one component in a larger Twin Builder Power Train design that includes a permanent magnet synchronous motor and supporting components taken from Twin Builder's component libraries. We will then analyze its performance in the Twin Builder environment. The following steps outline the process for accomplishing these goals.

- Add a new project and Modelica Model in Twin Builder.
- Place the Modelica components for the simple power train model into the **Diagram Editor**.
- Connect the Modelica simple power train components.
- Assign the various simple power train component properties.
- Compile and update the Twin Builder project.
- Add the Modelica component to the Twin Builder Schematic.
- Add Twin Builder Components to the design and add connections.
- Assign properties to the Twin Builder components.
- Define outputs and Create Reports for the simple power train design.
- Set up and run a Transient analysis for the simple power train design.

Adding a New Project and Modelica Model in Twin Builder

1. Select **File** > to create a new project in Twin Builder. You can rename the project by right-clicking the project icon in the **Program Manager**, and selecting **Rename**.
2. Create a new Modelica model by doing one of the following:
   - Select **Twin Builder** > **Add Model**.
   - Press Ctrl+Shift+M.
   - In the Program Manager **Definitions** folder, right-click the **Models** folder and select **Add Definition**.



3. Select **Modelica** as the Type and enter a **Name** for the new model.

   The Modelica environment is displayed, with the Modelica libraries appearing in the **Component Libraries** window on the right.

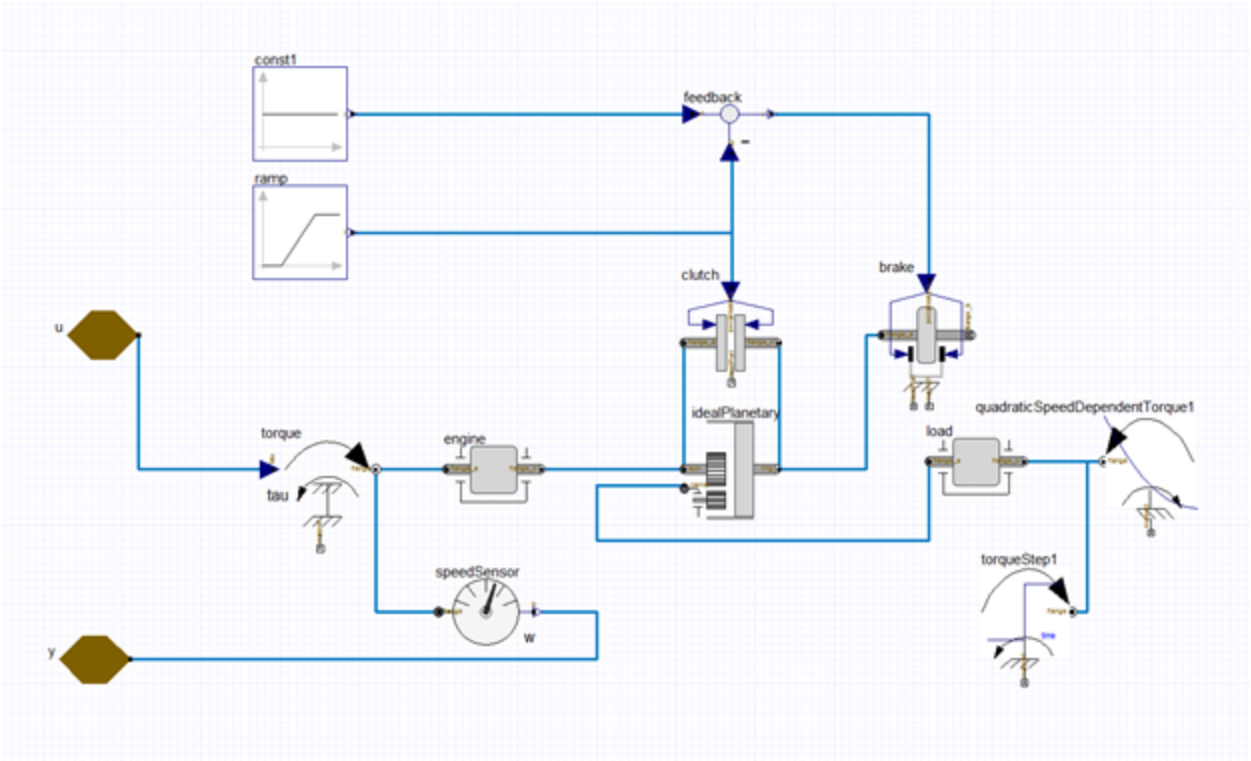
Next, you will place the Modelica components that are needed for the simple power train model into the **Diagram Editor**.

Placing Modelica Components for the Simple Power Train Model

All of the required components for the Modelica simple power train gear shift model can be found in the **Modelica** library in the locations shown in the following table.

Component Instance	Location in Library
u	Modelica/Blocks/Interfaces/RealInput
y	Modelica/Blocks/Interfaces/RealOutput
torque	Modelica/Mechanics/Rotational/ Sources/Torque
engine	Modelica/Mechanics/Rotational/Components/Inertia
idealPlanetary	Modelica/Mechanics/Rotational/Sources/IdealPlanetary
clutch	Modelica/Mechanics/Rotational/Components/Clutch
brake	Modelica/Mechanics/Rotational/Components/Brake
load	Modelica/Mechanics/Rotational/Components/Inertia
quadraticSpeedDependentTorque1	Modelica/Mechanics/Rotational/ Sources/QuadraticSpeedDependentTorque
torqueStep1	Modelica/Mechanics/Rotational/ Sources/TorqueStep
speedSensor	Modelica/Mechanics/Rotational/Sensors/SpeedSensor
feedback	Modelica/Blocks/Math/Feedback
const1	Modelica/Blocks/Sources/Constant
ramp	Modelica/Blocks/Sources/Ramp

1. Locate each of the components in the library.
2. Drag and drop the components into the Diagram Editor in the Modelica Environment, arranging them as shown in the figure below. Press **Esc** after placing each component to stop the placement action. (Connections will be added later.)

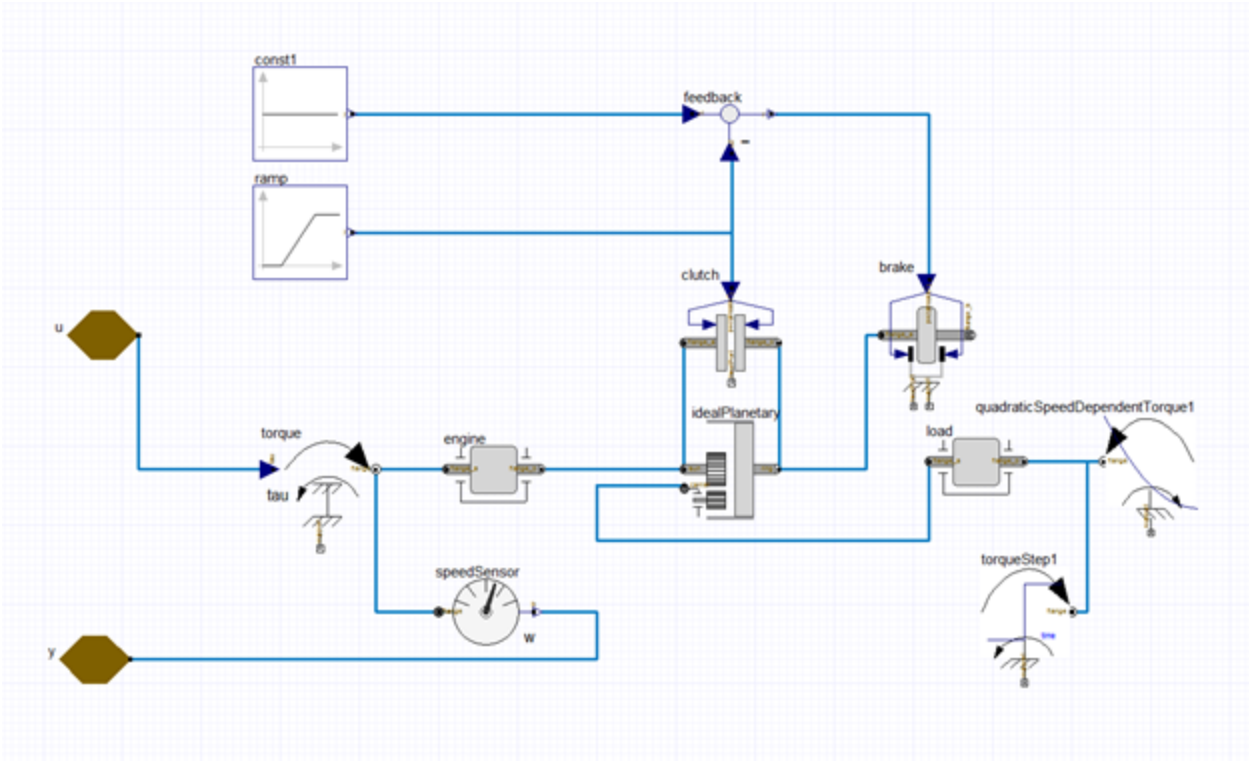


3. You can change the instance names for a components by clicking on the component to display its Properties window. You can enter the desired **instanceName** in the associated **Value** field on the **Properties** tab
4. Press **Esc** to complete each action.

Next, you will add connections between the various power train components.

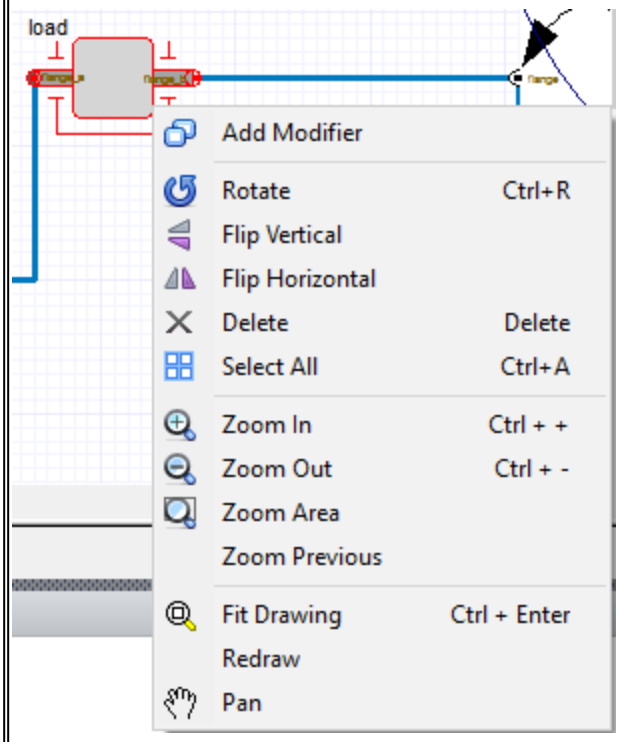
Connecting Modelica Power Train Components

To connect the components for the power train system as shown in the diagram below:



1. Click on an input or output pin of one of the components and connect it to the desired pin of another component.
2. Repeat for each connection.

Note	More graphic options are available in a context menu by right-clicking on a component, as shown below. For more information on these options refer to the *Diagram Editor* section in the Twin Builder help.



Next, you will set up properties for the power train model.

Setting Up Model Properties for the Modelica Power Train Example

The model properties for the components used in this example are summarized in the following table.

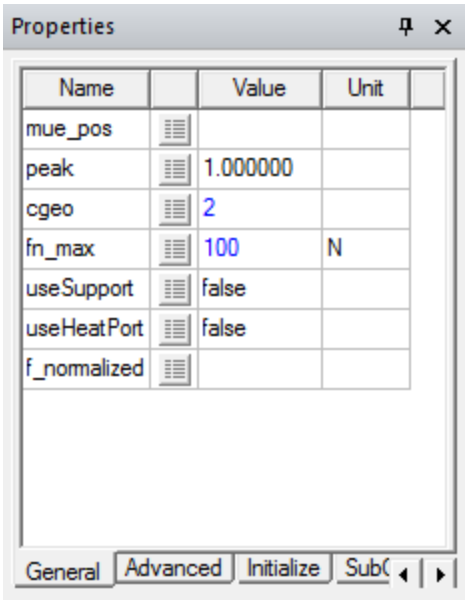
Component Instance	Properties
u	Use defaults
y	Use defaults
torque	Use defaults
engine	J = 1
idealPlanetary	ratio = 75/50
clutch	cgeo = 2 fn_max = 100 phi_rel(fixed = true)* w_rel(fixed = true)* ***NOTE:** To set up the **phi_rel**and **w_rel** properties, refer to the "Modification of variables phi, w, phi_rel

Component Instance	Properties
	and w_rel" on page 3-111 step below.
brake	cgeo = 2 fn_max = 100
load	J = 10 phi(start = 0)* w(start = 0)* ***NOTE:** To set up the **phi**and **w** properties, refer to the "Modification of variables phi, w, phi_rel and w_rel" on the facing page step below.
quadraticSpeedDependentTorque1	tau_nominal = -20 w_nominal = 100
torqueStep1	stepTorque = 10 startTime = 1
speedSensor	Use defaults
feedback	Use defaults
const1	k = 1
ramp	duration = 0.1 startTime = 2

To display the editable parameters and properties of a component:

1. Click the model to open the model's **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options from the table above as needed. For example, to set properties for the **brake** component in the Diagram Editor:

a. Click on the **brake** component to display its **Properties** window.



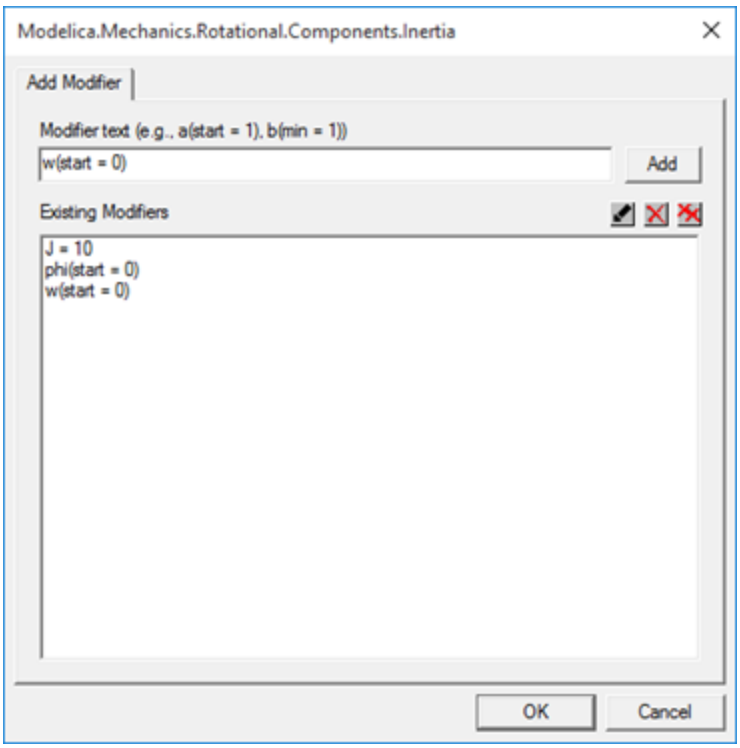
b. On the **General** tab, per the information in the above table, enter **2** for the **cgeo** property; and **100** for **fn_max**. No other changes are needed. (The **Properties** tab allows you to set the name of the component instance.)

Modification of variables phi, w, phi_rel and w_rel

This example requires modification of variables **phi** and **w**for the **load** component; and variables **phi_rel**and **w_rel**for the **clutch** component. This cannot be done directly in the Properties window, but instead requires that modifiers be added.

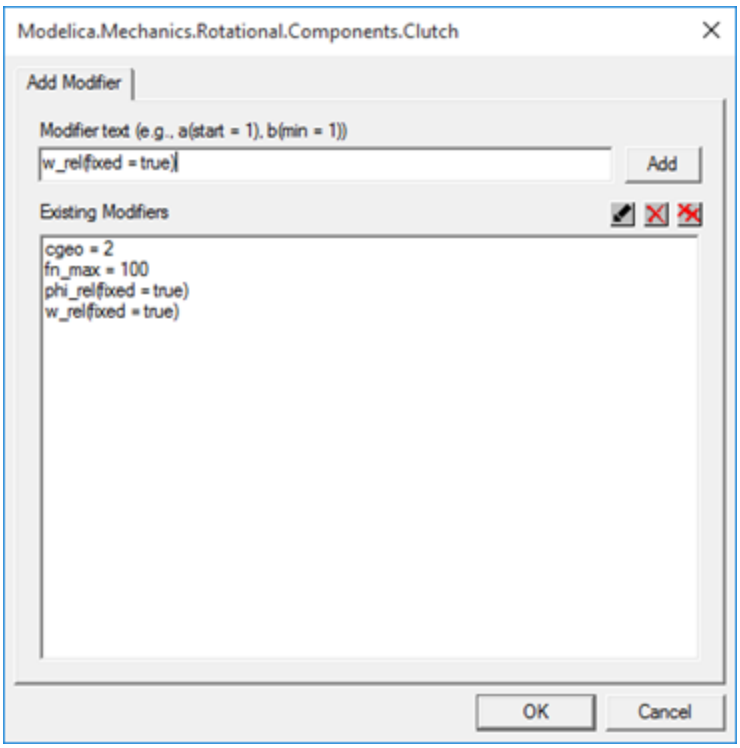
1. To add modifiers to **load**, select the **load** instance in diagram editor and on the ribbon Modelica tab click **Add Modifier** (or right click on the **load** instance and select **Add Modifier** from the context menu). In the **Add Modifier** dialog box, add the following three lines:

```
phi(start = 0)

w(start = 0)
```



2. To add modifiers to **clutch**, select the **clutch** instance in diagram editor and on the ribbon Modelica tab click **Add Modifier** (or right click on the **load** instance and select **Add Modifier**). In the **Add Modifier** dialog box, add the following three lines:

```
phi_rel(fixed = true)

w_rel(fixed = true)
```

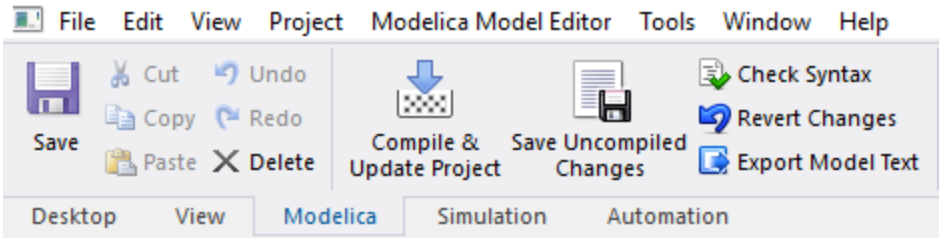


Next, you will compile and update the project.

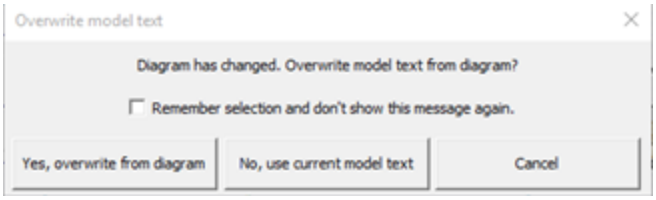
Compiling and Updating the Simple Power Train Project

A model created or edited using the Diagram Editor can be saved and used in the project only after the model text has been updated. When you have completed the Modelica model, it is ready to be compiled and added to the Twin Builder project as a component. To do this:

1. Click **Compile & Update Project** on the ribbon Modelica tab as shown below.



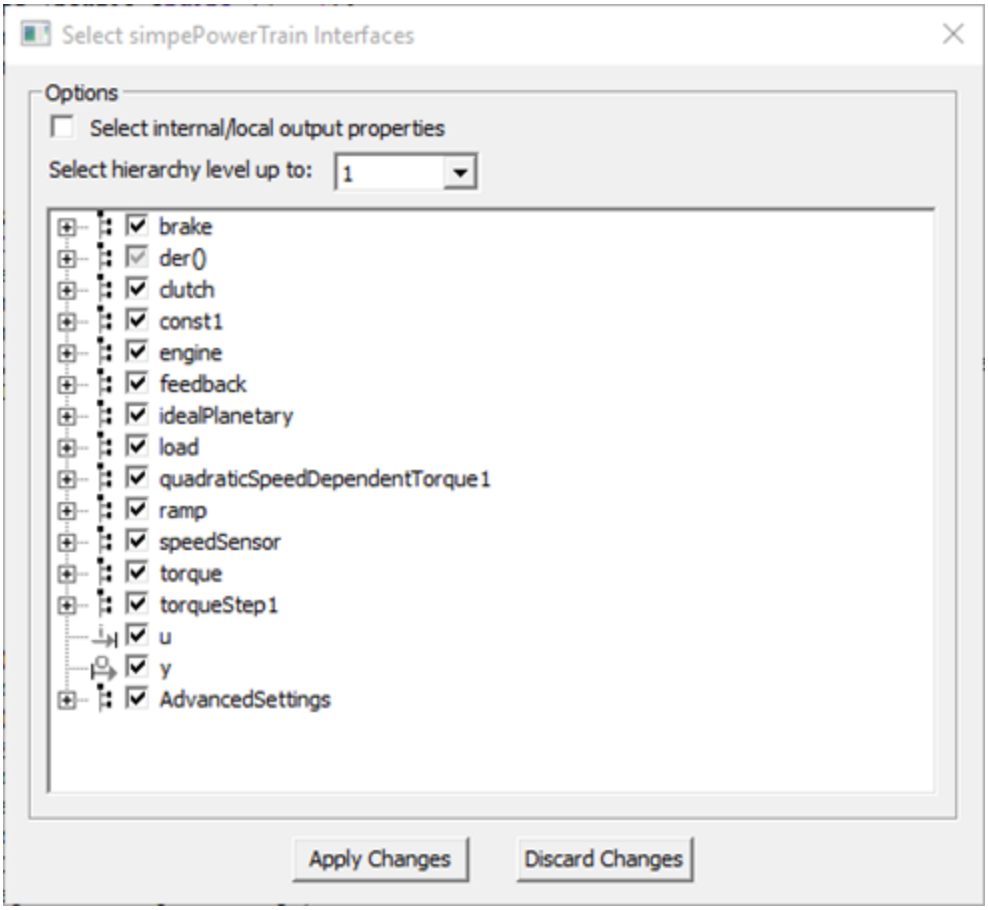
The **Overwrite model text** dialog box appears.



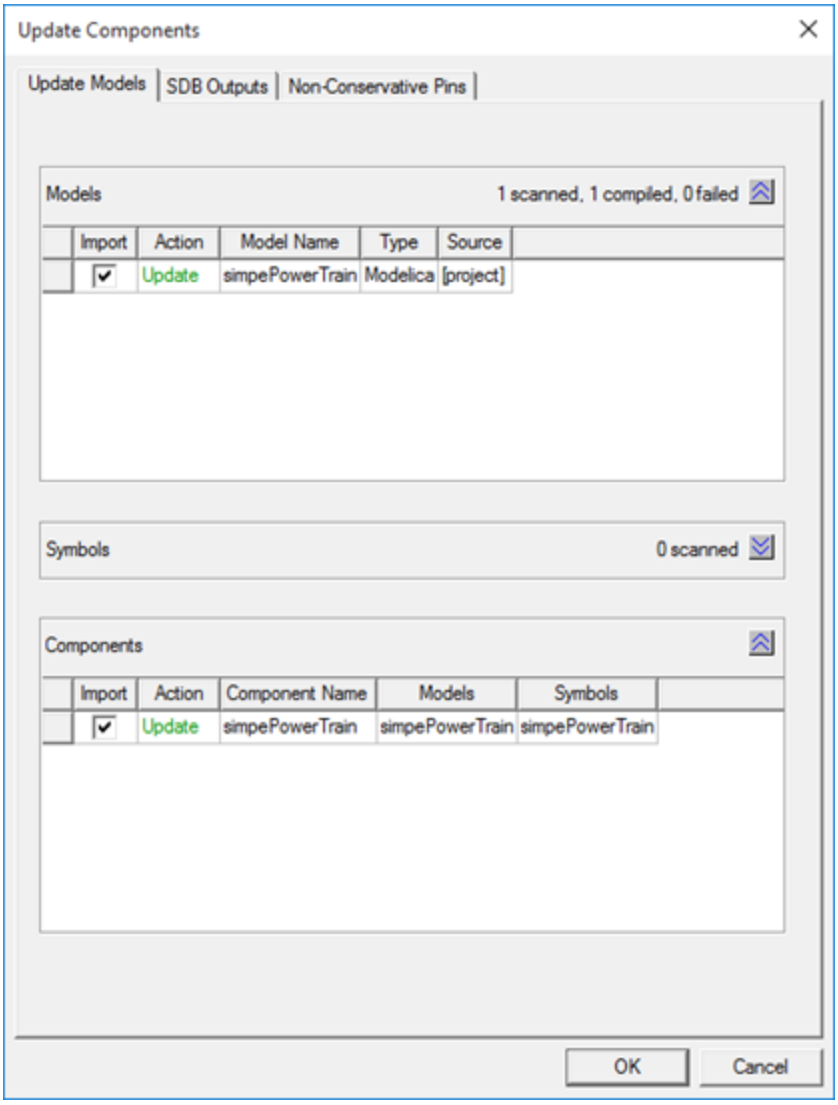
Note	This same dialog box appears when you switch to the **SimModel** tab to view the Modelica model text. You can choose either to overwrite from the diagram, or to use the current model text. The latter choice won't apply any changes made in the diagram since the most recent save or compile.

2. For the Simple Power Train example, all the Modelica model editing has been completed in the diagram, so select **Yes, overwrite from diagram**. The model is compiled. This may take several seconds.

3. Following the compile operation, you can select interfaces for the simple power train model, or just accept the default set of preselected interfaces.

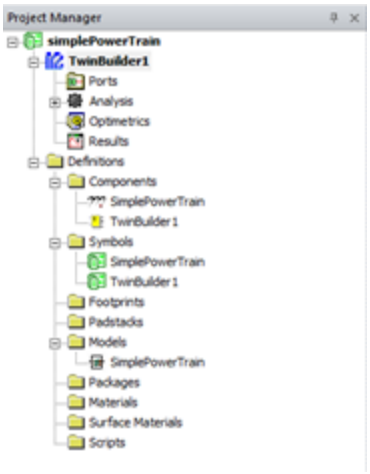


4. When finished selecting interfaces, click **Apply Changes** to close the dialog and apply the changes to the simple power train model. The **Update Components** dialog box appears with both the simplePowerTrain model and component selected for import into Twin Builder.



5. Click **OK** to complete the update and import the simplePowerTrain model and component into Twin Builder.

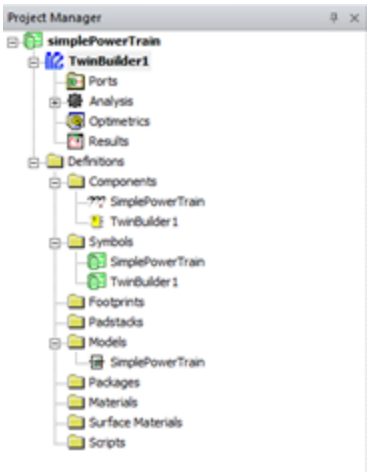
The Modelica simplePowerTrain component is now ready to use in a Twin Builder design. You can find the Modelica component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



Next, you will add the component to a Twin Builder schematic.

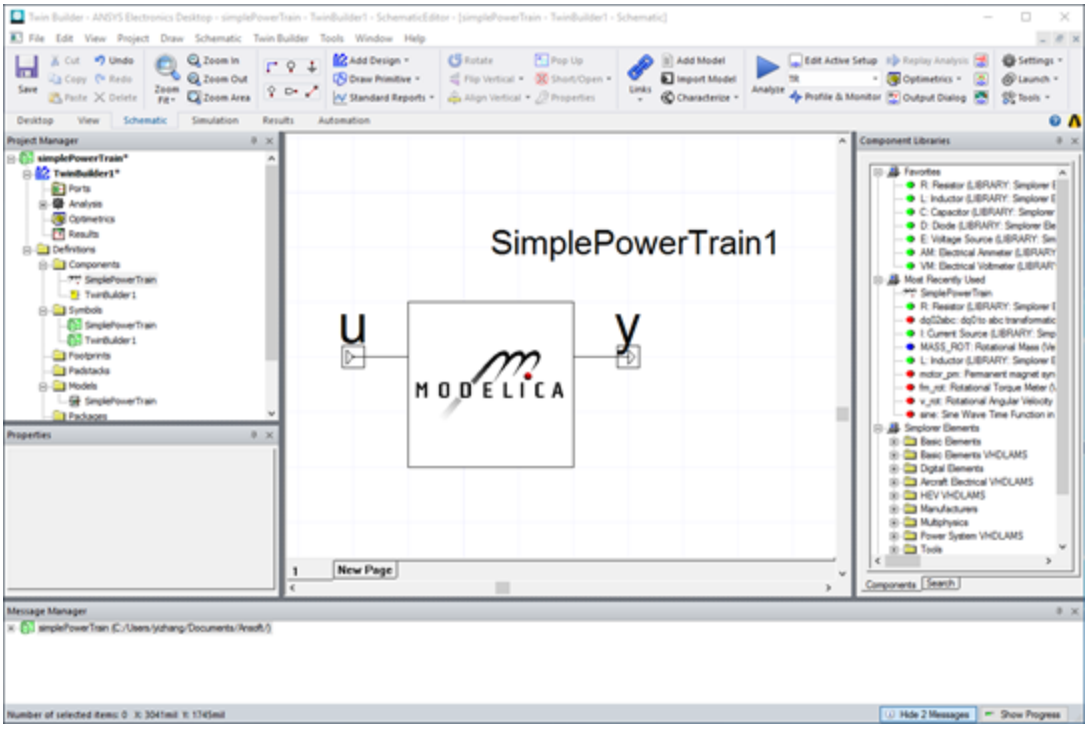
Adding the Simple Power Train Component to a Twin Builder Schematic

When the Modelica Simple Power Train component is ready to use in a Twin Builder design, you can find the component in the **Project Manager** > **Definitions** > **Components** folder, as shown below.



1. To open the schematic editor for a design, in the Project Manager window, find the project in which the Modelica component definition is located, then double-click the icon for an existing design to open the schematic editor.
2. Locate the Modelica SimplePowerTrain component in the **Project Manager** > **Definitions** > **Components** folder, then drag and place it in the design

schematic, as shown below.



Note	If you need to look inside or modify the Modelica model, do one of the following to open the model again in the Modelica environment: • In the schematic editor, right-click the Modelica component and select **Edit Model**. • In the Project Manager, find the model in **Definitions** > **Models** folder, then right-click on the model icon and select **Edit Model**. Any change made in the Modelica environment will need to be compiled and updated to the project again.

Next, you will add and connect Twin Builder components to the design.

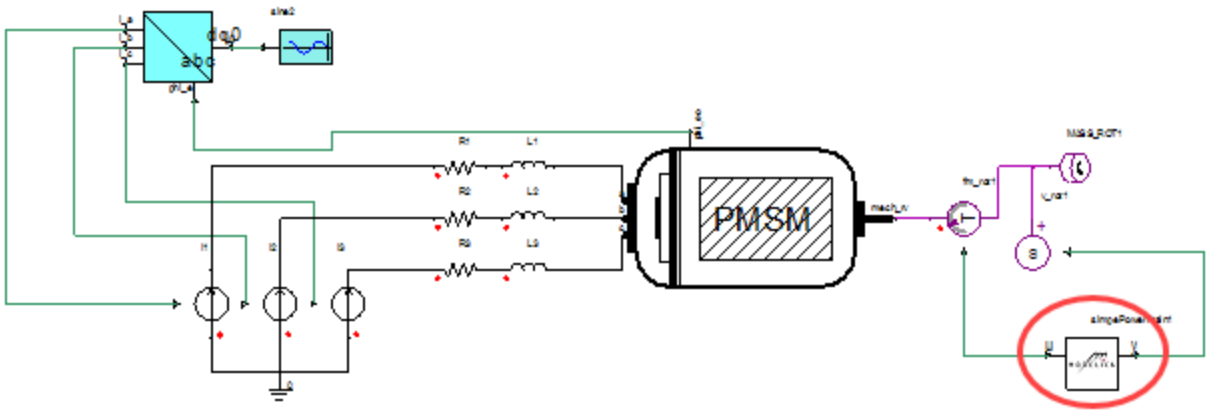
Placing and Connecting Twin Builder Components for the Simple Power Train Design

All of the required Twin Builder components for the simple power train design can be found in the Twin Builder libraries locations shown in the following table.

Component Instance	Location in Library
sine1	Basic Elements VHDLAMS\Tools\Time Functions:sine
dq02abc1	Aircraft Electrical VHDLAMS\Basic:dq02abc

Component Instance	Location in Library
Current sources I1 – I3	Basic Elements\Circuit\Sources:I
ground	VDALibs VHDLAMS\hybrid_emc_vda\elements_general\gnd
Resistors R1 – R3	Basic Elements\Circuit\Passive Elements:R
Inductors L1 – L3	Basic Elements\Circuit\Passive Elements:L
PMSM	Aircraft Electrical VHDLAMS\Load:motor_pm
Rotational torque meter	Basic Elements\Measurement\Mechanical\Velocity-Force-Representation\Rotational_V:FM_ROT
Rotational angular velocity source	Basic Elements\Physical Domains\Mechanical\Velocity-Force-Representation\Rotational_V:V_ROT
Rotational mass	Basic Elements\Physical Domains\Mechanical\Velocity-Force-Representation\Rotational_V:MASS_ROT

1. Locate each of the components in the library.
2. Drag and drop the components into the Schematic Editor in the Twin Builder Environment, arranging them as shown in the figure below. (The circled Modelica component was already placed on the schematic.)

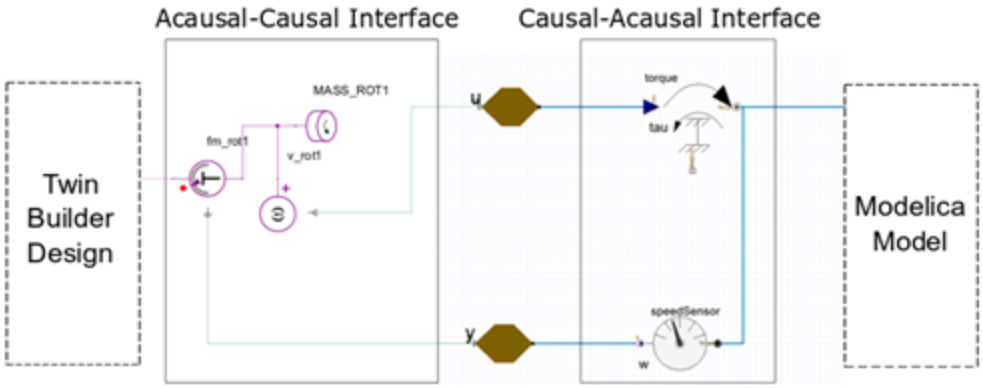


3. Press **Esc** after placing each component to stop the placement action. Connections will be added below.
4. To change a component instance name, click on one to display its properties. Enter the desired **instanceName** in the associated **Value** field on the **Properties** tab.
5. To connect the components, click on an input or output pin of one of the components and connect it to the desired pin on the other component. Note that:

- You can right-click a component and choose **Pin Visibility** to toggle pin display.
- In the figure above, input pin **i_q** for the **dq02abc** component is fed by a sinusoidal signal. The other two inputs (**i_d** and **i_0**) can be hidden.
- In the figure above, for the PMSM component, input signal pin **free** is not used, and output pin **phi_eo** must be turned on and connected to input **phi_e** of the **dq02abc** component.

6. Repeat this procedure for each connection.

The Modelica model is based on a Functional Mock-up Interface (FMI) which only supports causal input/output pins. However, physical connections in Twin Builder is usually acausal, as a result, acausal-causal and causal-acausal connectors are necessary for this example to connect Twin Builder PMSM element and created Modelica power train model, as shown below.



Next, you will assign properties for the Twin Builder components.

Setting Up Properties for the Simple Power Train Twin Builder Components

The properties for the Twin Builder components used in this example are summarized in the following table.

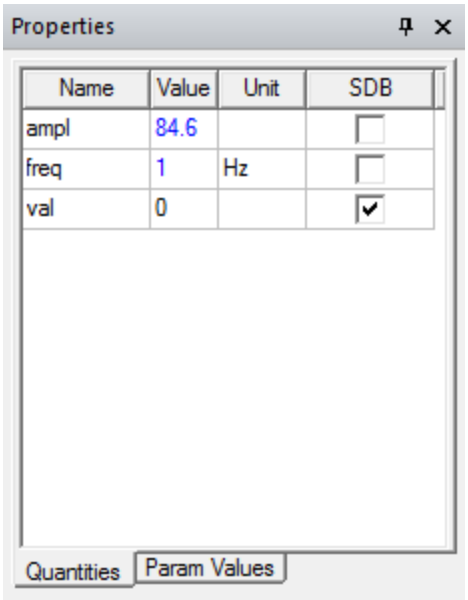
Component Instance	Properties
sine1	Ampl = 84.6 Freq = 1 Hz
dq02abc1	I_d = 0 I_0 = 0
Current sources I1 – I3	Use defaults
ground	Use defaults
Resistors R1 – R3	R1 = R2 = R3 = 1 Ohm

Component Instance	Properties
Inductors L1 – L3	L1 = L2 = L3 = 1e-5 H
PMSM	Phi_e = 0 Free = false
Rotational torque meter	Show pin for torque
Rotational angular velocity source	show pin for value
Rotational mass	J = 1 kgm2

To display the editable parameters and properties of a component:

1. Click the component to open its **Properties** window.
2. Select the various tabs at the bottom of the Properties window to browse and find parameters and properties you need to edit. All other properties retain their default values.
3. Enter values or select options from the table above as needed. For example, to set properties for the **sine1** component:

    a. Click on the **sine1** component to display its **Properties** window.



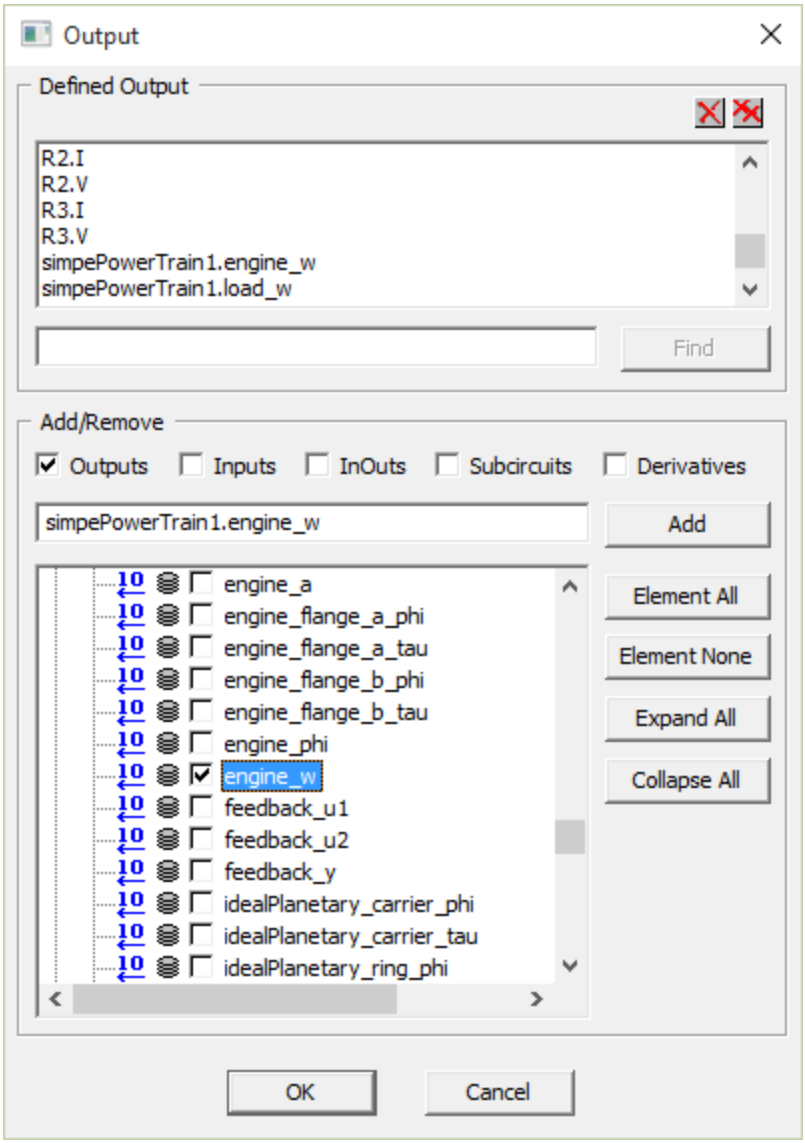
    b. On the **Quantities** tab, per the information in the above table, enter 84.6 for the **ampl** property; and **1**Hz for **freq**. No other changes are needed.

Next, you will define outputs and create reports for the simple power train design.

Defining Outputs and Creating Reports for the Simple Power Train Design

When all of the simple power train components are ready to use in the Twin Builder design, you can define outputs and set up reports.

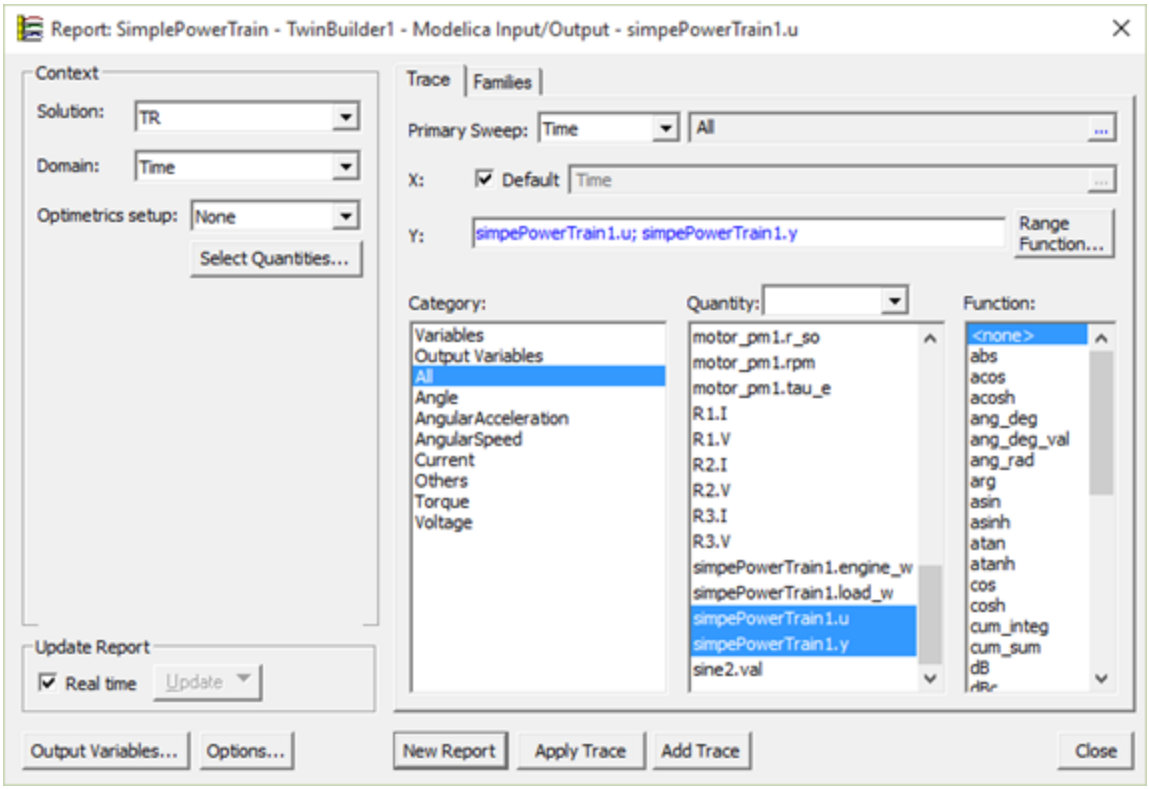
1. To create reports or graphical representations of simulation results, you must first define outputs in the **Output** dialog to make them available for plotting. This can be done by clicking on **Output Dialog** in the **Schematic** tab of the desktop ribbon. For this example, the following outputs are selected:
    - simPowerTrain1.u
    - simPowerTrain1.y
    - simPowerTrain1.engine_w
    - simPowerTrain1.load_w

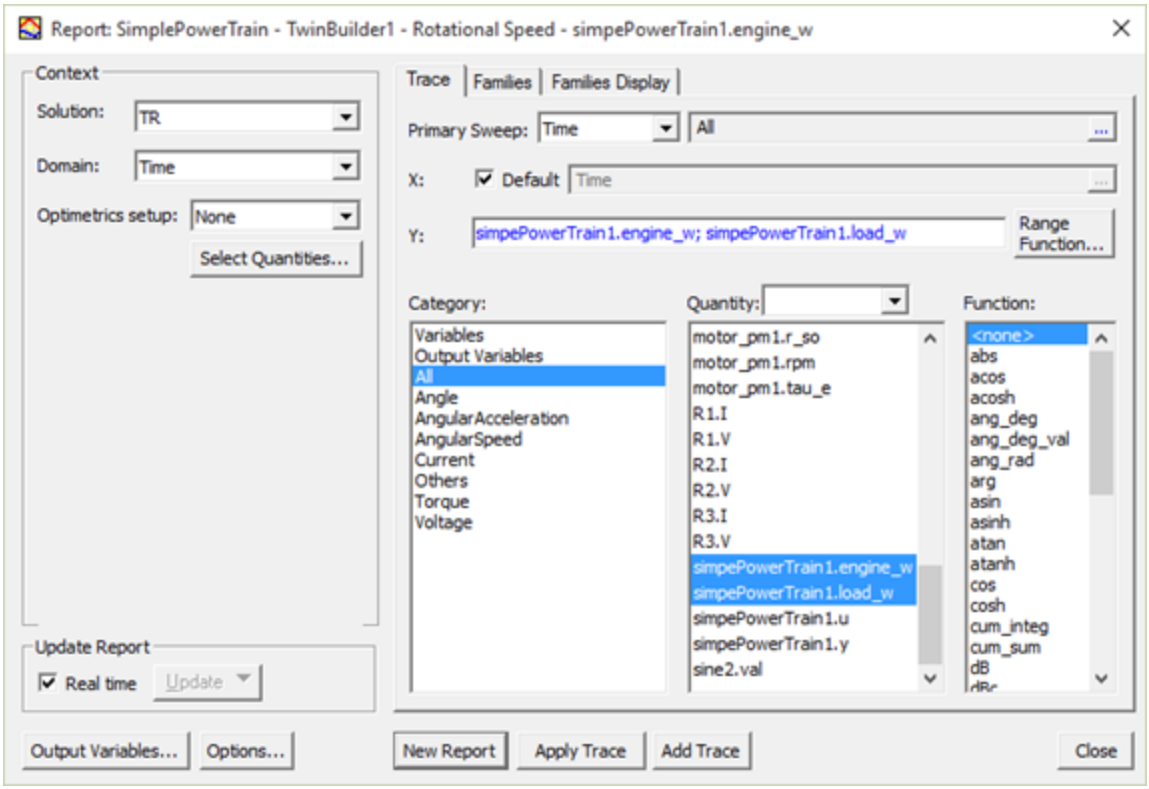


Refer to *Setting the Outputs for Simulation* in the Twin Builder help for more information.

2. In the Project Manager, you can then right-click on **Results** and choose **Create Standard Report** to add various types of reports for the design being simulated.

   The outputs you set in the **Output** dialog box will be available in the Report dialog boxes for plotting. Below are example reports dialogs for plotting the outputs mentioned above.





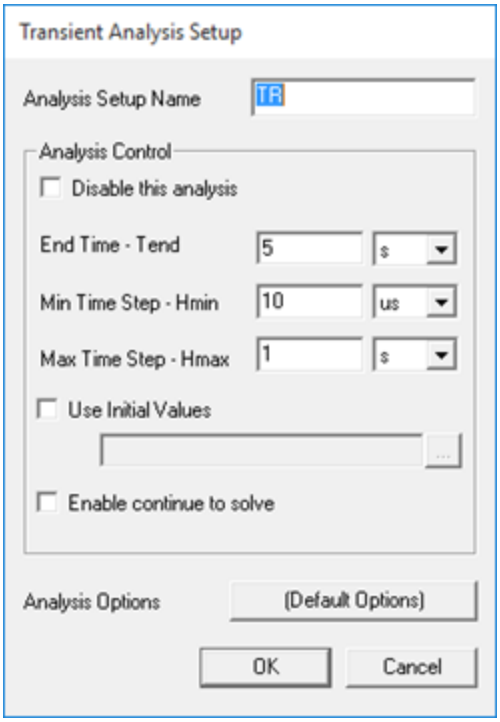
For details on the various ways for viewing simulation results, see *Generating Reports and Postprocessing* in the Twin Builder help.

Next, you will set up and run a transient analysis to generate simulation results.

Setting Up and Running a Transient Analysis for the Simple Power Train Design

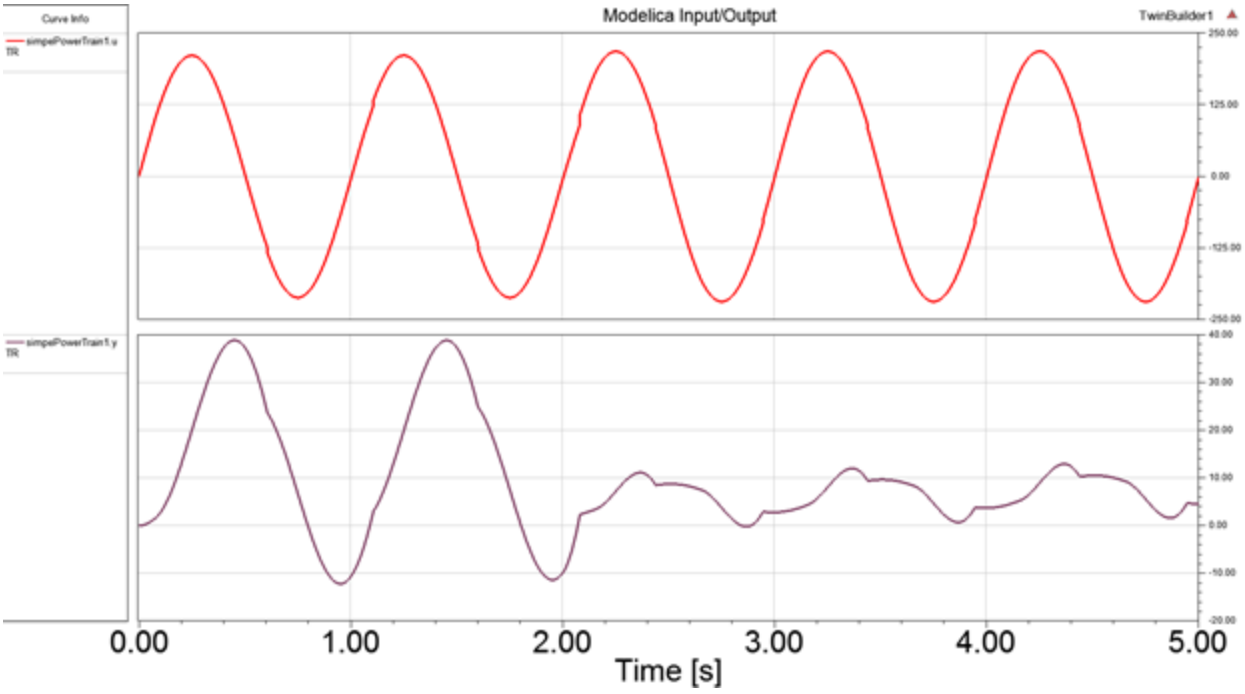
In this example, we would like to simulate the time-domain response of the simple power train design. This requires that a transient analysis be performed on the design. By default, a transient analysis, **TR**, is created for the design. You can find it in the **Project Manager** under **Analysis** of the design.

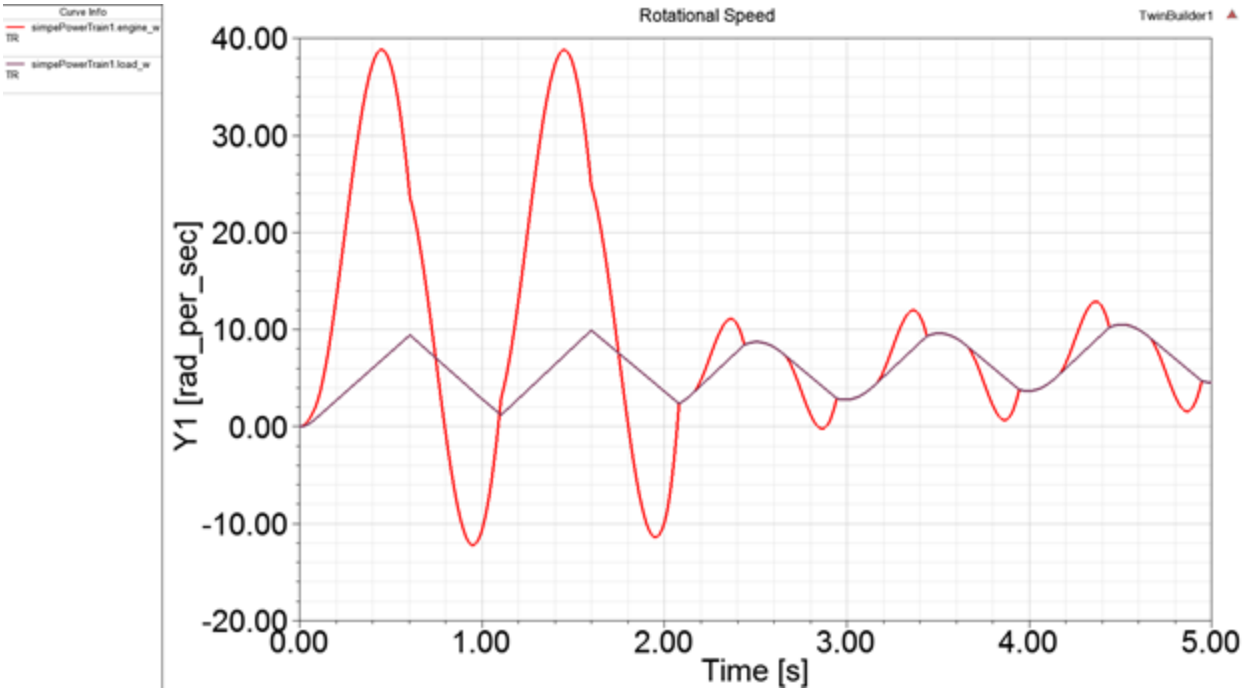
1. Double-click on **TR** to set up end time and the desired time step for the simulation.



2. For this example, set **End Time** to 5 s, **Min Time Step** to 10 us, and keep **Max Time Step** as the default 1 ms. For more details about transient analysis, see *Transient Analysis Setup* in the Twin Builder help.
3. To start the simulation, run the transient analysis by doing one of the following:
   - On the desktop ribbon, **Schematic** tab, click the **Analyze** icon.
   - In the Project Manager, right click on **TR** of the design, then select **Analyze**.

Below are sample reports of the simulation results for the design.





4 - Index

C

Component Libraries 2-2, 2-3

E

Examples 1-2

H

Help 1-2

I

Introduction 1-1

M

Manuals 1-2

S

SIMPLORER

- Documentation 1-2
- PDF manuals 1-2
- Starting simulation system 2-4

Simplorer

- Help 1-2

Simulation

- Examples 1-2

Starting, SIMPLORER 2-4