

# Defining Blocks in Block Definition Diagram

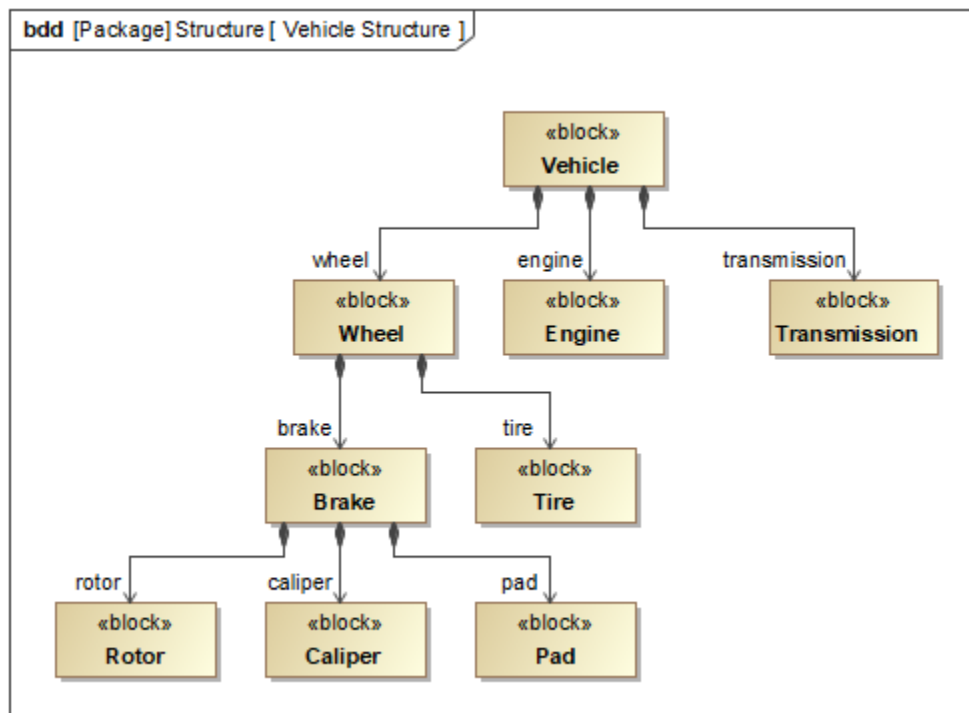
On this page

- [Introduction](#)
- [Creating Blocks](#)
  - [Use the Create Element dialog in the Model Browser](#)
  - [Use the palette on the diagram pane](#)
  - [Paste a list in the Containment tree or on a diagram pane](#)
  - [Drag .fmu file on the diagram pane](#)
- [Creating Association Block](#)
- [Relating Blocks](#)

## Introduction

When describing your system structure, you should start by defining [Blocks](#) in a [SysML Block Definition Diagram](#). Blocks represent the system hierarchy in terms of systems and subsystems. You can model either the logical or physical decomposition of a system, and the specification of software, hardware, or human elements.

The notation for a Block is a rectangle with the stereotype «block» preceding the name. Blocks are the elements of definition because they have a name only. The figure below illustrates the vehicle structure of defined Blocks in the SysML Block Definition Diagram. The figure is created from the [VehicleStructure.mdx](#) sample model.



The vehicle structure in a SysML Block Definition Diagram.

## Creating Blocks

You can create [Blocks](#) in a [SysML Block Definition Diagram](#) in the following ways:

- [Use the Create Element dialog in the Model Browser](#)
- [Use the palette on the diagram pane](#)
- [Paste a list in the Containment tree or on a diagram pane](#)
- [Drag .fmu file on the diagram pane](#)

### Use the Create Element dialog in the Model Browser

You can create all elements including [Blocks](#) in the [Model Browser](#) from the shortcut menu.

To create a Block in the Model Browser

1. In the Model Browser, right-click the [Package](#) in which you want to store the new Block.

2. Do one of the following:
  - From the shortcut menu, select **Create Element**.
  - Press Ctrl+Shift+E.
3. In the **Create Element** dialog, select Block.
4. Type its name.  
The new Block is created in the Model Browser. You can represent it in the SysML Block Definition Diagram by dragging it from the [Model Browser](#) onto the [diagram pane](#).

## Use the palette on the diagram pane

You can create [Blocks](#) directly on the [SysML Block Definition Diagram](#) pane. The Blocks created on the [diagram pane](#) are created in the [Model Browser](#), too.

To create a Block on the diagram pane

1. Open the existing SysML Block Definition Diagram or create a new one. [How to create a new diagram >>](#)
2. Do one of the following:
  - From the [diagram palette](#), select the **Block** button.
  - Press B.
3. Click on the diagram pane.
4. Type its name.  
The new Block is created in the Model Browser and represented in the [SysML Block Definition Diagram](#).

## Paste a list in the Containment tree or on a diagram pane

You can create Blocks by copying a listed text from other resources (e.g. Word, Excel, HTML, etc.) and pasting it either in the Containment tree or diagram pane.

Detailed procedures are provided in the [Creating elements from other resources](#) page.

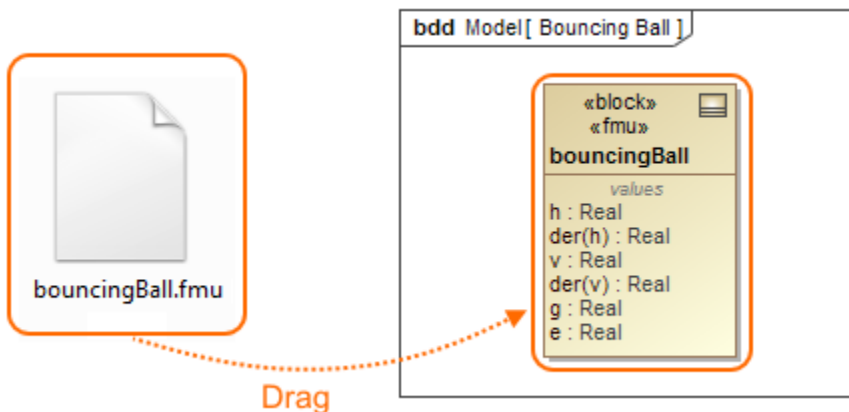
## Drag .fmu file on the diagram pane

You can represent any model components that are exported to Functional Mock-up Interface (FMI) standard (.fmu files) as Blocks in the [SysML Block Definition Diagrams](#).

 The SysML Plugin supports FMI 1.0 and 2.0 versions.

To represent an .fmu file as the Block

1. Drag the .fmu file onto the Block Definition Diagram pane.  
The Block with name, value properties, their types, and applied fmu stereotype is created.



You can execute one fmu Block or co-simulate cooperated fmu Blocks using the [Cameo Simulation Toolkit](#).

## Creating Association Block

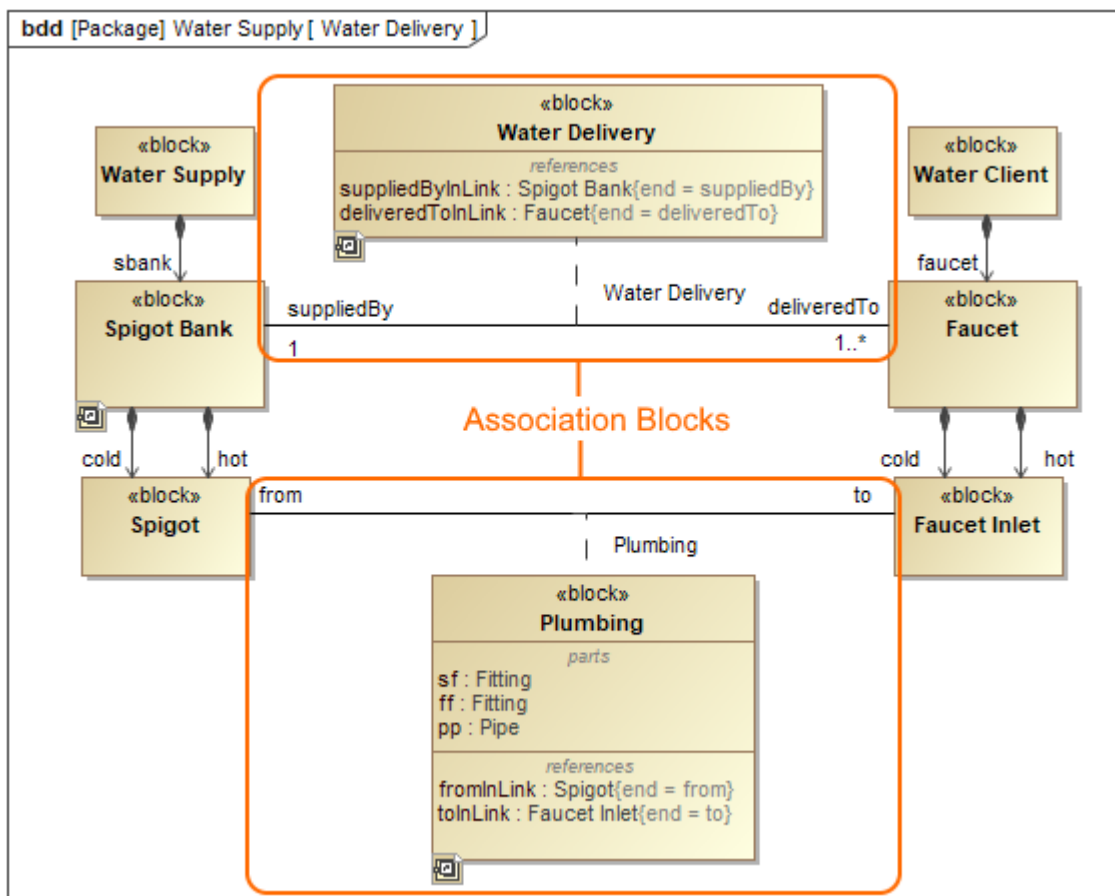
You can use an [Association Block](#) to show decomposition of the connector similar to the way that blocks show the decomposition of parts. You can create an [Association Block](#) or [Association Block with Owned Ends](#) between two [Blocks](#) in the [SysML Block Definition Diagram](#). You can represent Association Block usages in the [SysML Internal Block Diagrams](#) by [setting Association Block as Connector type](#).

To create an Association Block or Association Block with Owned Ends

1. Create two Blocks on a diagram pane.
2. From the diagram palette, select **Association Block** or **Association Block with Owned Ends** button.
3. On the diagram pane, select a [Block](#) to be used as the source.
4. Select a target by either selecting an existing Block on the diagram pane, or by clicking on empty space on the diagram to create the target Block.  
An Association Block or Association Block with Owned Ends is created between the source and target [Blocks](#). The [Participant Properties](#) are created and shown in the Association Block automatically and shown in its [references compartment](#) area.

 [Participant properties](#) are created automatically after an [Association Block](#) is created between [Blocks](#).

The following figure shows two Association Blocks: *Water Delivery* and *Plumbing*. The connection between the *Spigot Bank* Block and *Faucet* Blocks is decomposed into the *Water Delivery* Block by creating an Association Block. The *suppliedByInLink* and *deliveredToInLink* Participant Properties are created and shown in the compartment area. The same is shown with the *Plumbing* Association Block. The figure is created from the [WaterSupply\\_19.0.mdzip](#) sample model.



Water Delivery and Plumbing Association Blocks in the SysML Block Definition Diagram.

## Relating Blocks

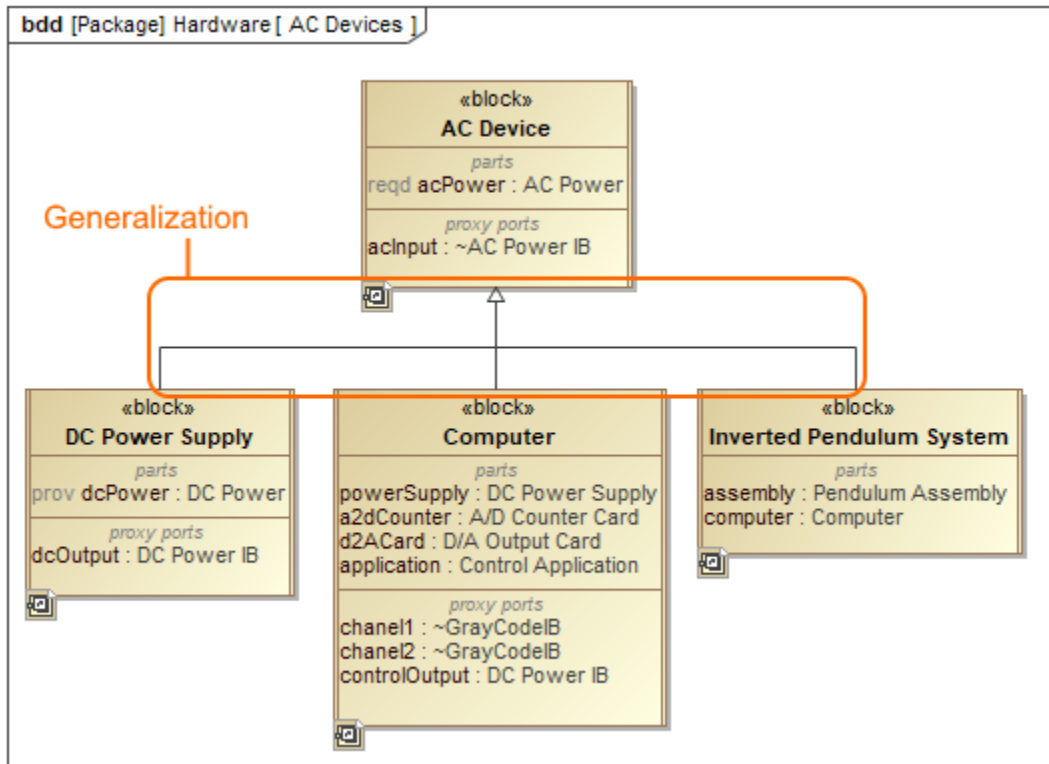
You can connect Blocks by using these main relationships:

- [Generalization](#)
- [Direct Composition or Composition](#)
- [Direct Association or Association](#)

### Generalization

The Generalization relationship conveys an inheritance between Blocks. It is usually used to create a hierarchy in your system. The notation is a solid line with a hollow, triangular arrowhead on the end.

The figure below shows that the *DC Power Supply*, *Inverted Pendulum System*, and *Computer* Blocks are types of the *AC Device* Block. All of the subtypes (*DC Power Supply*, *Inverted Pendulum System*, and *Computer*) require all the characteristics of *AC Device* but add their own specialized characteristics as well. The figure is created from the [InvertedPendulum.mdzip](#) sample model.

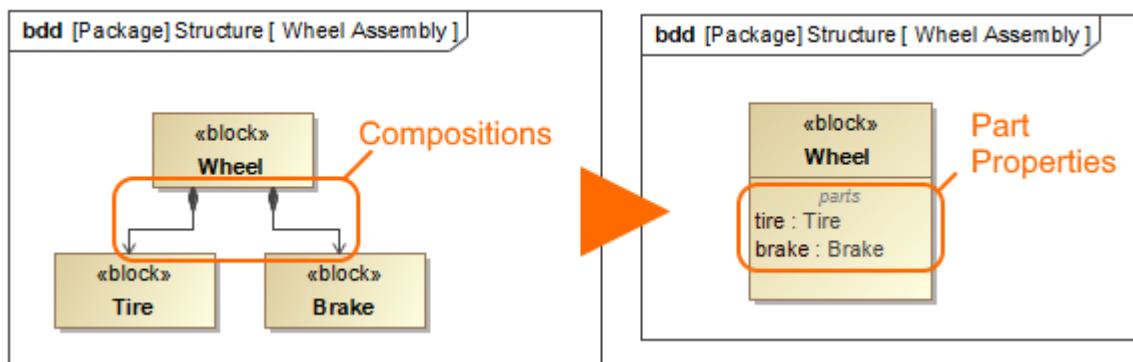


Generalization relationship between Blocks in SysML Block Definition Diagram.

### Direct Composition or Composition

The Direct Composition or Composition relationships convey a structural decomposition of Blocks. The notation is a solid line between two Blocks with a solid diamond on the composite end.

The following figure shows that the *Wheel* is composed of *Tire* and *Brake*. The Composition can be represented in two different ways: by showing Composition relations between Blocks or by showing Part Properties on a decomposed Block *compartment*. The figure is created from the [VehicleStructure.mdzip](#) sample model.

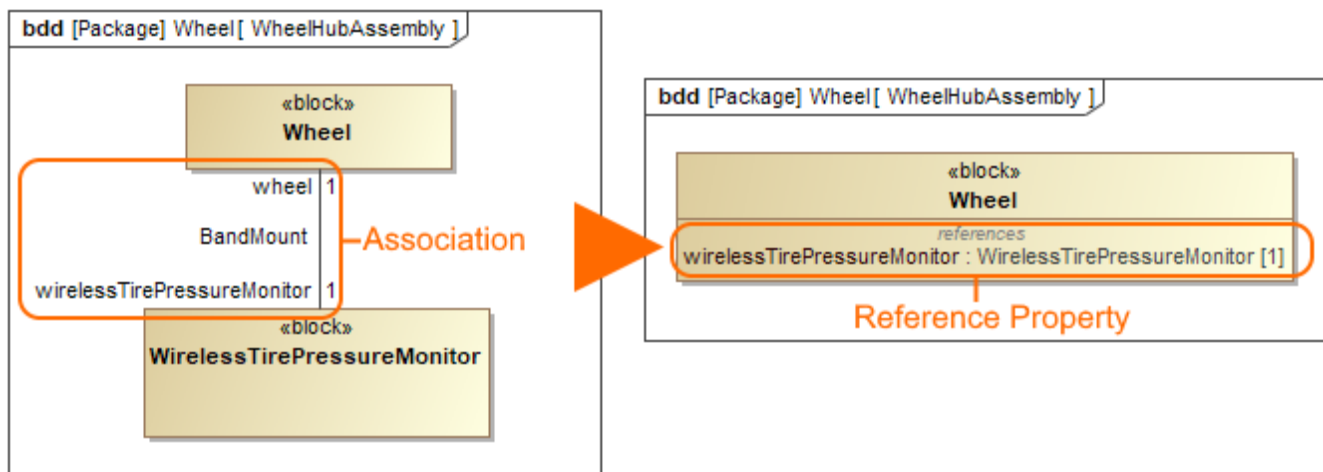


Direct Composition relationship between Blocks in the SysML Block Definition Diagram.

### Direct Association or Association

The Direct Association or Association conveys that a connection can exist between those Blocks. Those Blocks can access each other for some purpose across the connection. The Direct Association notation is a solid line with an arrowhead on the end, while the Association notation consists only of a solid line.

The figure below illustrates the Association between the *Wheel* and *WirelessTirePressureMonitor* Blocks. The name of the Association is *BandMount*, which describes the type of connection that could exist between the wheel and wireless tire pressure monitor. The Association can be represented in two different ways: by showing the Association relation between Blocks or by showing the Reference Property on a decomposed Block [compartment](#). The figure is created from the [hybrid sport utility vehicle.mdzip](#) sample model.



Association relationship between Blocks in SysML Block Definition Diagram.

For how to create, remove, change a style, route, create a line jump, insert a shape on a path, or manage path compartments, read the [Working with paths and relationships](#) page.