Analyzing usages and dependencies

On this page

- Used By command
- Depends On command
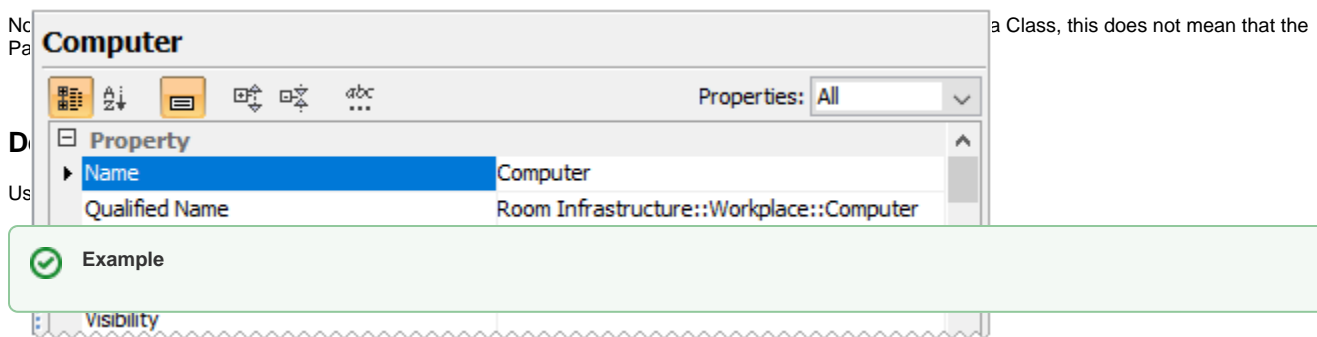- Searching for usages and dependencies

No Magic modeling tools provide a way to track and analyze element usages and dependencies in models, understand the relationships between used and dependent elements, and find the diagrams where these elements are represented. You can do all of this using the commands **Used By** and **Depends On**, as described below.

Used By command

The **Used By** command allows you to find all the elements that use a selected element (e.g., as property, tag etc.). This also includes usage in diagrams.

> ✅ **Example**
> In the figure below, the *Computer* Property has its type set to the *WorkStation* Class. In other words, the *Computer* Property uses the *WorkStation* Class as its type. If you execute the **Used By** command for the *WorkStation* Class, you will get the *Computer* Property as one of the results.

No ~~~~ a Class, this does not mean that the
Pa

D

Us

> ✅ **Example**

Searching for usages and dependencies ~~lass is used by the *Computer* Property.~~

You can find usages and dependencies of a particular model element by opening its shortcut menu from the model browser or a diagram.

To search for usages/dependencies of a model element

---

1. In the model browser or a diagram right-click the element, the usages or dependencies of which you want to find.
2. In the shortcut menu of the element click **Related Elements**, and select **Used By** or **Depends On** (depending on what you want to find).
3. In the **Usages/Dependencies Search Options** dialog, select the search options you want to use.
4. Click **OK** to start the search.

The search results are listed in the usages/dependencies search results window at the bottom of the modeling tool. The search results window also provides additional information, such as element type and how a particular element is used by/in the element for which you perform the search.

Related pages

- Analyzing usages and dependencies
- Traceability
- Specifying criteria for querying model
- Element usage in diagrams
- Identifying Package Dependencies
- Comparing projects
- Validation
- Model Visualizer
- Suspect Links

Let's say you have a model with the content tree shown in the figure below. Execute the **Depends On** command for the *Workplace Infrastructure* Package to see how its content depends on the outer elements (not owned by the Package) of the model. One of the search results shows that the *Work place Infrastructure* Package depends on the *Power Socket* Class, which is the type of several Ports owned by this package.
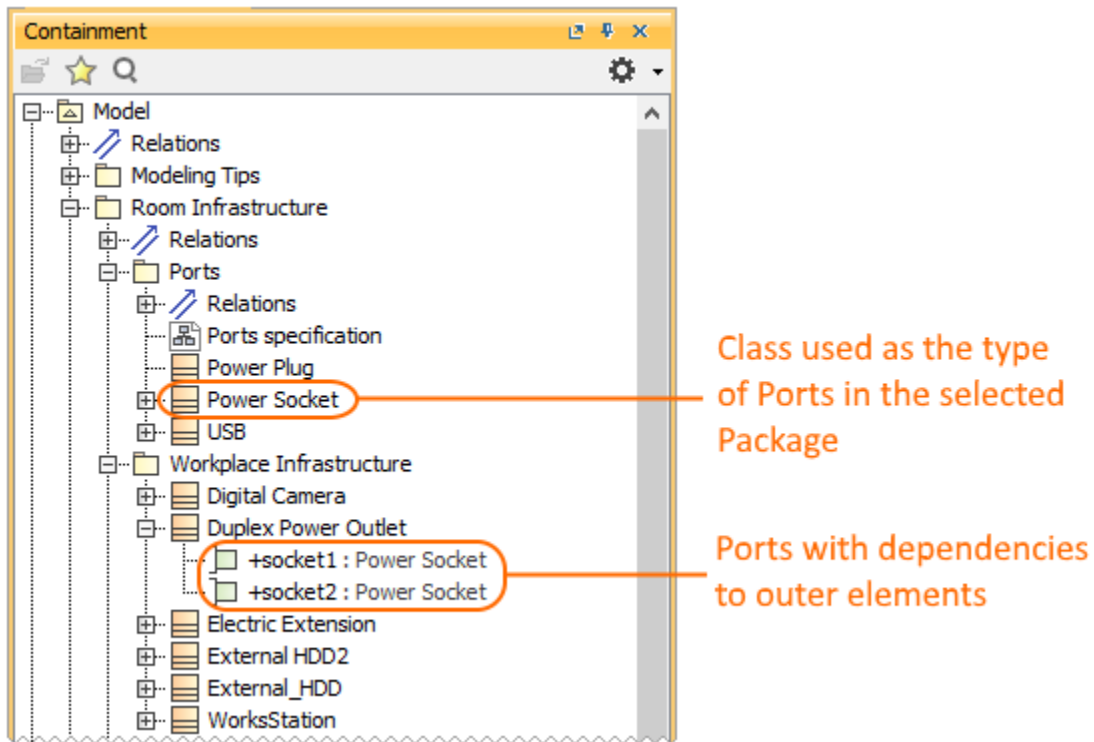
The content tree of a sample model with the *Workplace Infrastructure* Package having dependencies to the outer elements of the model.

Note that in this particular situation you must select the **Search recursively** check box in the **Usages/Dependencies Search Options** dialog. This allows you to search for dependencies of the Package content rather than the Package itself. Also, select the **Ignore inner interrelations** checkbox to eliminate the dependencies between the elements contained by the package from the search results.