# Controlling dependency creation between used projects

In earlier versions of our modeling tool, you could accidentally create unneeded or incorrect (from the user standpoint) dependencies between used projects in the read/write mode. Careless editing often caused cyclic module dependencies or inadvertent expansion of the scope of the other projects when the same used project was used in several other projects.

Automated project usages fix this problem.

Every automatically created usage must be backed by an existing user-defined used project usage. If it is not backed, the automated used project usage must be explicitly confirmed by the user. The usage cannot cause undesired effects in other projects while it is unconfirmed.

Automated used project usage *A B* is called **unconfirmed** when there is no user-defined usage path from A to B (neither direct user-defined usage *A B* nor user-defined usage path of arbitrary length *A ... X B*).

> ⚠ User-defined usages are always confirmed.

In ordinary circumstances, automated usages are not visible to the user. When the user modifies the main project's model and adds dependencies to used projects or sub-used projects, these usages are proper. They are backed by the fact that the used project or sub-used project is attached to the project in some way; thus, the user-defined usage path refers from the main project to the used project or sub-used project.

However, there are situations when automated used project usages are not backed and thus unconfirmed.

> ⚠ An unconfirmed automated used project usage is considered to be invalid.

This is just one of the project composition errors. Other project composition errors include similarity to recovered element errors, conflicting user-defined usage parameters, and cyclic dependencies.

Unconfirmed automated used project usages are caught by special automatic validation rules. They are displayed in validation results as warnings or errors depending on the project composition. The validation results prompt the user to remedy the situation by either confirming or rejecting the usage.

Unconfirmed usages can occur in several cases. They are described as follows.

> ⓘ **Case #1**
>
> When you edit a model part belonging to a used project with read/write permissions, an additional model-level reference can be created either back to the project itself (cycle) or to another used project.
>
> These model-level usages may or may not be valid from the user point of view. Since the creation of usages is very easy, you can inadvertently create undesired usages. On the other hand, usages can be entirely benign. Our modeling tool cannot decide whether the usage is good or bad. For example, creating references from the used project *Implementation* to the used project *Requirements* is valid, while creating references in the opposite direction may be invalid since requirements usually are standalone, and implementation refers to them.

> ⓘ **Case #2**
>
> Reorganization or removal of sub-used projects may affect the projects or other used projects using the used project. This happens only in cases when a new refactored used project does not carry all of the contents that the old used project contained.
>
> Consider a used project *ElectricComponents* with three sub-used projects: *Capacitors, Inductances,* and *Resistances.* Also consider a project *RadioSet,* which uses the used project *ElectricComponents.* Now let's use one model element from each sub-used project in the project *RadioSet.* As a result, the three automated usages
>
> *RadioSet Capacitors,*
>
> *RadioSet Inductances,* and
>
> *RadioSet Resistances*
>
> will be created.
>
> If the used project *ElectricComponents* was refactored by re-importing the sub-used project *Capacitors* and *Inductances* and completely separating the sub-used project *Resistances,* then when loading the project *RadioSet,* the automated usages *RadioSet Capacitors* and *RadioSet Inductances* will be automatically updated. However, the usage *RadioSet Resistors* will remain unaddressed. It will appear as an unconfirmed used project usage, and you will have to confirm it, i.e., explicitly attach the used project *Resistances* to the project by creating the user-defined usage.