Unresolved dependencies

Before exporting or sharing a package as an independent used project, you need to identify and solve package dependencies first. More specifically, you have to make sure that the package does not depend on any external elements (except other used projects) in a project.

The same situation occurs when you edit the used project inside the current project (the used project permissions should be set to read-write) and introduce dependencies from the used project back to the project where it is being used. In this case, you will be asked to solve these dependencies when saving the used project.

However, the modeling tool allows for continuing without solving these dependencies right away. The elements, which were referenced, but are missing in the used project will be shown as recovered elements (see Maintaining Decomposed Model Integrity).

You can choose how the dependency checking will be carried on in your projects by changing the **Check Dependencies on Package Export/Sharing** environment option (from the **Options** menu, select **Environment** > **General** > **Save/Load**). You can choose from one of the following:

- **Do not allow dependencies**. If this option is selected, then unsolved dependencies should not be allowed while exporting or sharing packages. The **Export** (or **Share**) button is not available until the dependencies are solved.
- **Allow dependencies**. If this option is selected, then the dependencies are identified, but it is not required to solve them and the **Export** (or **Share**) button is available.
- **Do not check**. If this option is selected, then the dependencies will not be checked at all. In this case, the selected package will be exported or shared as is. Be aware, that this might cause an increasing number of the recovered elements in your project (see Maintaining Decomposed Model Integrity).

Related pages

- Identifying Package Dependencies
- Package Dependencies panel
- Analyzing dependencies among elements