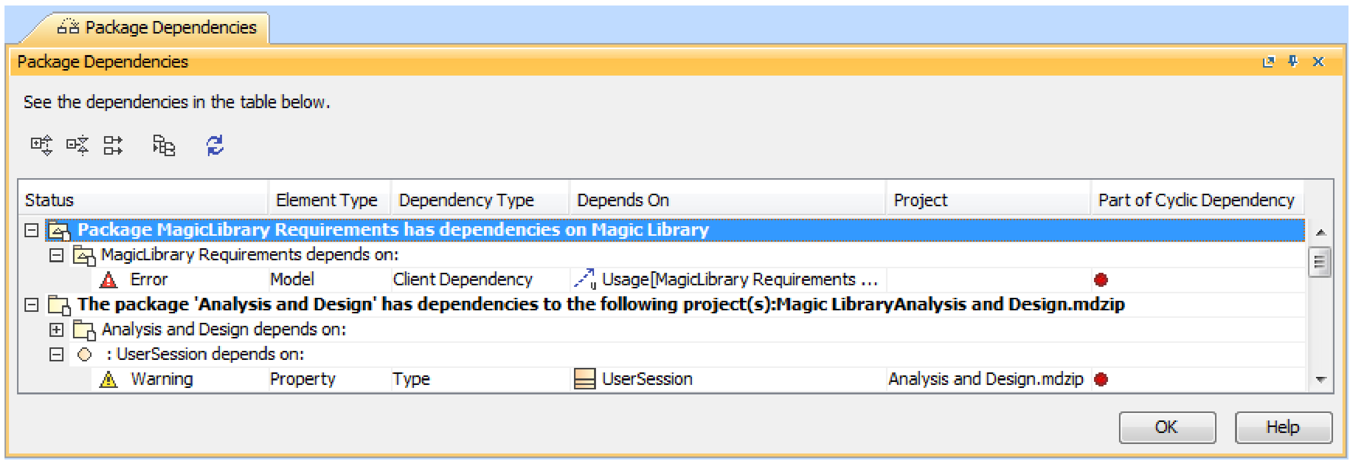
Package Dependencies panel

Before exporting or sharing a package as a independent used project, you need to identify and analyze package dependencies first. The **Package Dependencies** panel will help you with that. The **Package Dependencies** panel is displayed when you select to share or export packages.

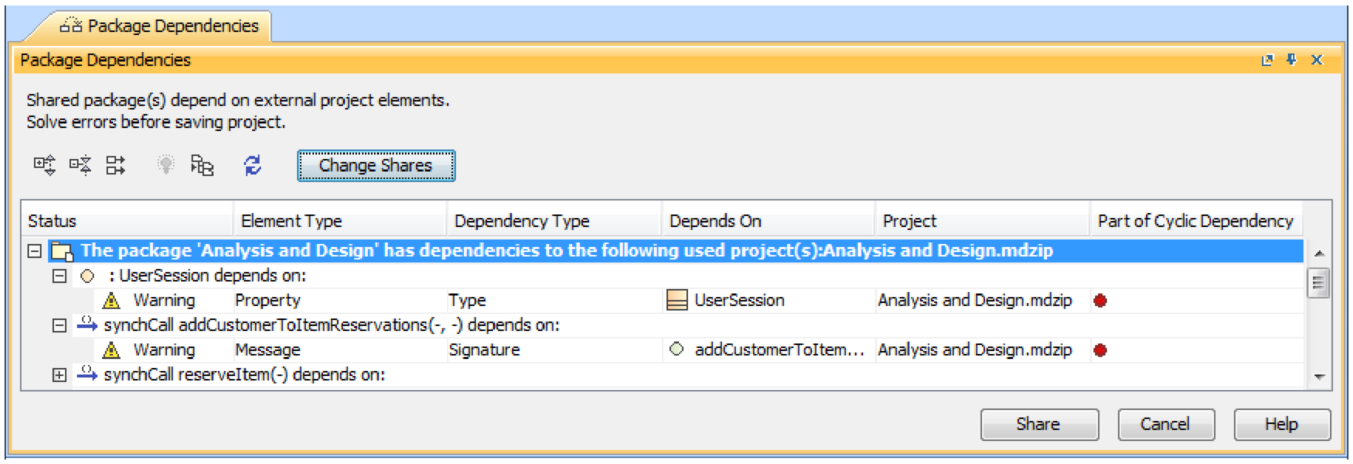
Before hiding a package, you can identify and analyze package usages and usages of all elements in the package first. The **Package Usage** window will help you with that. The **Package Usage** panel is displayed when you choose to check usages of elements in the selected packages for hiding.



See the dependencies in the table below.

Status	Element Type	Dependency Type	Depends On	Project	Part of Cyclic Dependency
Package MagicLibrary Requirements has dependencies on Magic Library					
MagicLibrary Requirements depends on:					
Error	Model	Client Dependency	Usage[MagicLibrary Requirements ...		●
The package 'Analysis and Design' has dependencies to the following project(s):Magic LibraryAnalysis and Design.mdzip					
Analysis and Design depends on:					
: UserSession depends on:					
Warning	Property	Type	UserSession	Analysis and Design.mdzip	●

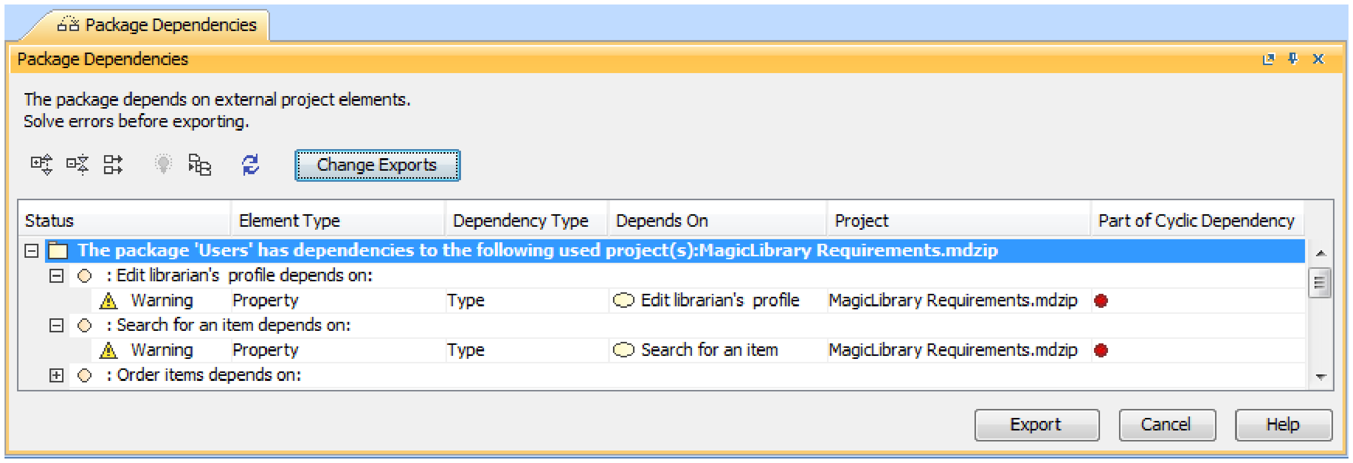
Package Dependencies panel



Shared package(s) depend on external project elements.
Solve errors before saving project.

Status	Element Type	Dependency Type	Depends On	Project	Part of Cyclic Dependency
The package 'Analysis and Design' has dependencies to the following used project(s):Analysis and Design.mdzip					
: UserSession depends on:					
Warning	Property	Type	UserSession	Analysis and Design.mdzip	●
synchCall addCustomerToItemReservations(-, -) depends on:					
Warning	Message	Signature	addCustomerToItem...	Analysis and Design.mdzip	●
synchCall reserveItem(-) depends on:					

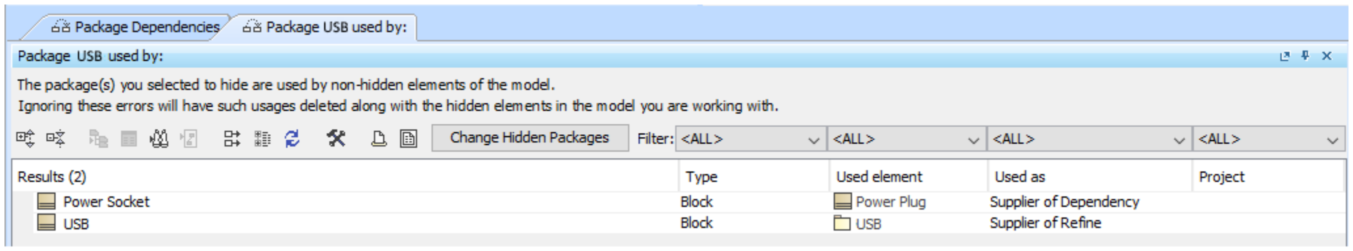
Package Dependencies panel when checking dependencies on package sharing



The package depends on external project elements.
Solve errors before exporting.

Status	Element Type	Dependency Type	Depends On	Project	Part of Cyclic Dependency
The package 'Users' has dependencies to the following used project(s):MagicLibrary Requirements.mdzip					
: Edit librarian's profile depends on:					
Warning	Property	Type	Edit librarian's profile	MagicLibrary Requirements.mdzip	●
: Search for an item depends on:					
Warning	Property	Type	Search for an item	MagicLibrary Requirements.mdzip	●
: Order items depends on:					

Package Dependencies panel when checking dependencies on package exporting


Package Dependencies | Package USB used by:

Package USB used by:

The package(s) you selected to hide are used by non-hidden elements of the model.
Ignoring these errors will have such usages deleted along with the hidden elements in the model you are working with.

Change Hidden Packages   Filter: <ALL> <ALL> <ALL> <ALL>

Results (2)	Type	Used element	Used as	Project
Power Socket	Block	Power Plug	Supplier of Dependency	
USB	Block	USB	Supplier of Refine	

Package Usage search window when checking usages on package hiding

The **Package Dependencies** panel has a table which shows the list of dependencies and buttons for managing data displayed in this table.

Button	Description
Expand All Tree Branches	Expands all nodes in the package dependencies tree.
Collapse All Tree Branches	Collapses all nodes in the package dependencies tree.
Display the Full Path of the elements	Displays the element full path next to the element name.
Solve	The button is enabled, when a dependency with **Error** status is selected in the table and a solution for the problem can be found. Clicking the button opens the dialog for choosing the solution for a specific dependency problem. This button is only visible when exporting or sharing a package.
Select in Containment Tree	Shows the selected element in the Browser. The button is available, when a dependency is selected in the table.
Refresh	Performs dependency analysis and refreshes the dependency table with the new analysis results.
Change Shares	Opens the **Shared Packages** dialog wherein you can select another packages to be shared. This button is visible only when sharing a package.
Change Exports	Opens the **Export Package to New Project** dialog wherein you can select another packages to be exported. This button is visible only when exporting a package.
Share	Closes the **Package Dependencies** panel and makes the package shared. The button is available, when the **Check Dependencies on Package Export/Sharing** environment option is set to *Allow dependencies* (to open the **Environment Options** dialog, select **Options** > **Environment**). This button is visible only when sharing a package.
Export	Closes the **Package Dependencies** panel and opens the **Save as/ Commit Settings** dialog for saving/committing the package as separate project. The button is available, when the **Check Dependencies on Package Export/Sharing** environment option is set to *Allow dependencies* (to open the **Environment Options** dialog, select **Options** > **Environment**). This button is visible only when exporting a package.

Column	Description
Status	Shows severity of the element dependency problem. The status can be **Error**, **Warning**, or **Info**. Dependencies that have **Error** status: • used project depends on the project Dependencies that have **Warning** status: • cyclic used project depends on other used projects Dependencies that have **Info** status: • project element depends on elements from shared packages (shared packages can belong both to the project and external project)

Element Type	Displays element type.
Dependency Type	Displays dependency type.
Depends On	Displays the model element, on which the package/model element is dependent.
Project	Displays the name of the used project file that owns the model element, on which the package/model element depends.
Part of Cycling Dependency	Indicates that the element is a part of the *cycling dependency**.

*** Cyclic dependency**

If there is a chain of dependencies such that A M(1), M(1) M(2), M(2) M(3), ..., M(X) M(A), where:

- A is an element from the used project M(A)
- M(1..X) are other used projects
- A M(x) is element A dependency on used project M(x)
- M(y) M(x) is a dependency of at least one element in used project M(y) on used project M(x),

then this chain is called a cyclic dependency and every atomic dependency in this chain is considered as part of cyclic dependency.

Related pages

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