Hiding packages

On this page

The Hiding packages utility lets you specify which model Packages, elements within the model packages, and Used projects should be hidden. Once specified, the public version of a model will be created by cutting/deleting the hidden parts. You can strip your project to reuse only a particular part of the project in another project. To strip your project, you must hide packages you do not want to share.

Hiding packages

To hide packages

1. Open a server project.
2. Do one of the following:
   - Right-click the package you want to hide. On the shortcut menu, click **Project Usages** > **Hide Packages**.
   - On the main menu, click **Collaborate** > **Hidden packages**.
3. The **Hidden Packages** dialog opens. From the **All Data** side, select the packages you want to hide and add them to the **Hidden Packages** side. Move the selected package back to the **All Data** side to unhide it.
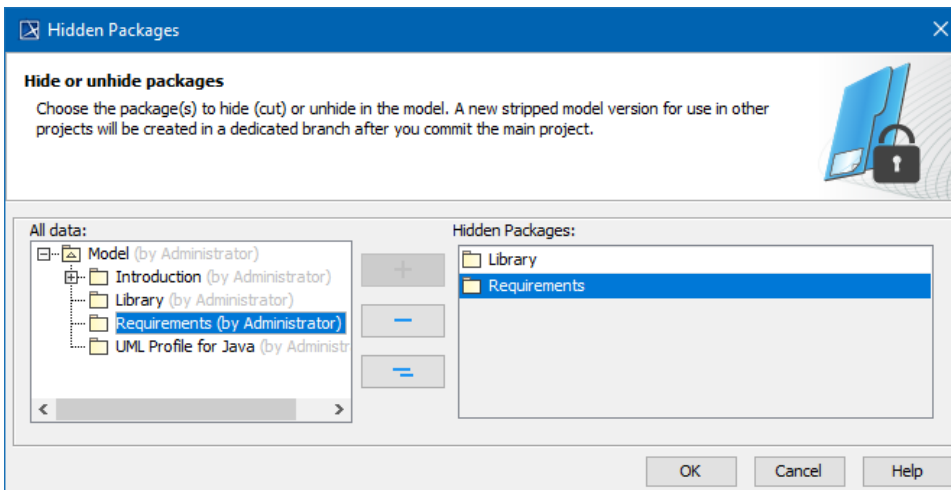
4. Click **OK** when you are done.
5. The package usage checking is suggested. Usage checking may be time-consuming; however, it helps to identify elements that are related to the soon-to-be hidden parts. If any issues are found, they are displayed and described in the **Package Usage window**.
6. After you complete or skip Usage checking, you will see a notification that hiding the selected elements will be completed only after committing your project.
7. Commit the project.
8. The stripped model version is created to be available to use in other projects. Hidden packages are marked with a lock sign ⬚🔒 in the Model Browser.

Hiding used projects

You can hide a used project the same way as other packages. When hiding a used project, you cannot hide separate used project parts: you need to hide the whole used project. After hiding is done, a dedicated branch is created with the stripped model version, where all elements of the used project are hidden from Containment tree due to being unused, and all references to these elements are cut from other model places.

What is more, you can hide the used project (directly used project) with its indirect usage (indirectly used project). When you want to hide an indirectly used project, you need to hide all directly used projects that are using the indirect usage.

> **Example**
> Your project has a structure ABC. You want to hide the "C" project inside the project "A". In this case, you need to hide the project "B" as well.

Unhiding used projects

You can unhide used projects as well. You can do so yourself manually in the **Hidden Packages** dialog (see above), or, in some cases, they may be hidden automatically. Typically, this happens when the status of the used project changes:

- when the used project is hidden/unhidden
- when the used project is updated
- when the used project is removed
- when the new used project is used
- when used or using projects are merged
- when the used project is exported
- when the used project is imported

Example
Your project has a structure ABC. You hide the project "B", and then the project "C" is hidden as well; however, if you add a new usage "D" that is using "C" and which introduces a new unhidden path to reach "C", the hidden usage "C" is automatically unhidden.

Dedicated branch for stripped model

A stripped model is stored in a dedicated branch created when element hiding is performed for the first time. By default, this branch is named *Public Project*. You can rename the branch if you need to.
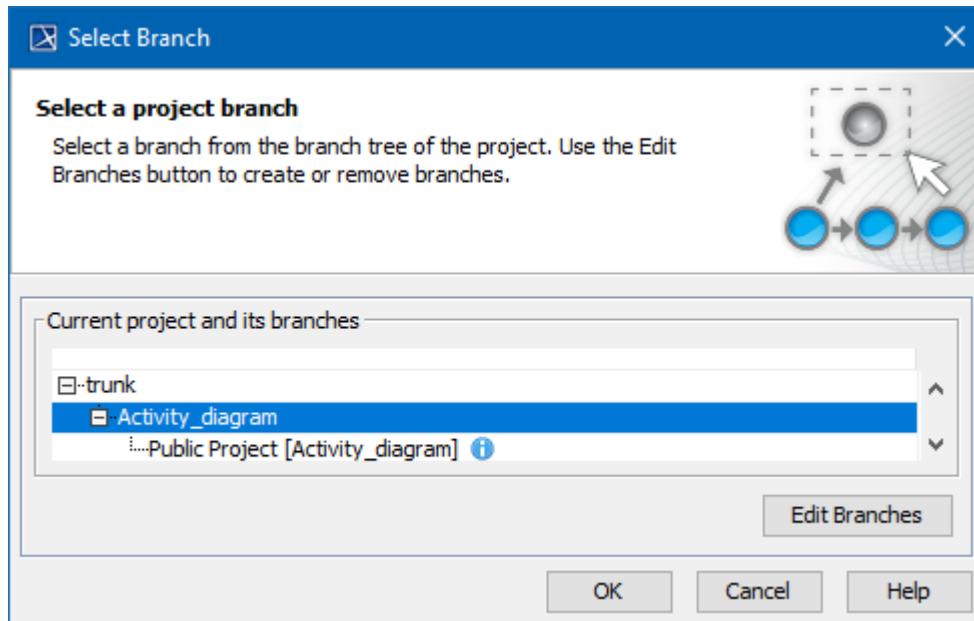
Project with it`s Public Project branch. The public project now can be reused in another project.

If all packages from the model are unhidden, the dedicated branch will be deactivated.

If the main project is using a public branch of another project and that branch gets deactivated, you will be informed about this change with a warning icon in the main project with a warning icon ⚠ in:

- Containment tree (under Project Usages root),
- Project Usages dialog,
- Advanced Project Usages Configuration dialog.

⚠ The state of the used project is checked at the same time interval as checking for a new project version in the background, or by clicking the Project Usages Update button.

Hidden package changes

After a package or a used project is successfully hidden or unhidden, an icon with a lock on a package (or used project) is shown in a Difference viewer.

The icon with a lock on a package (or on a used project package) is also shown in:
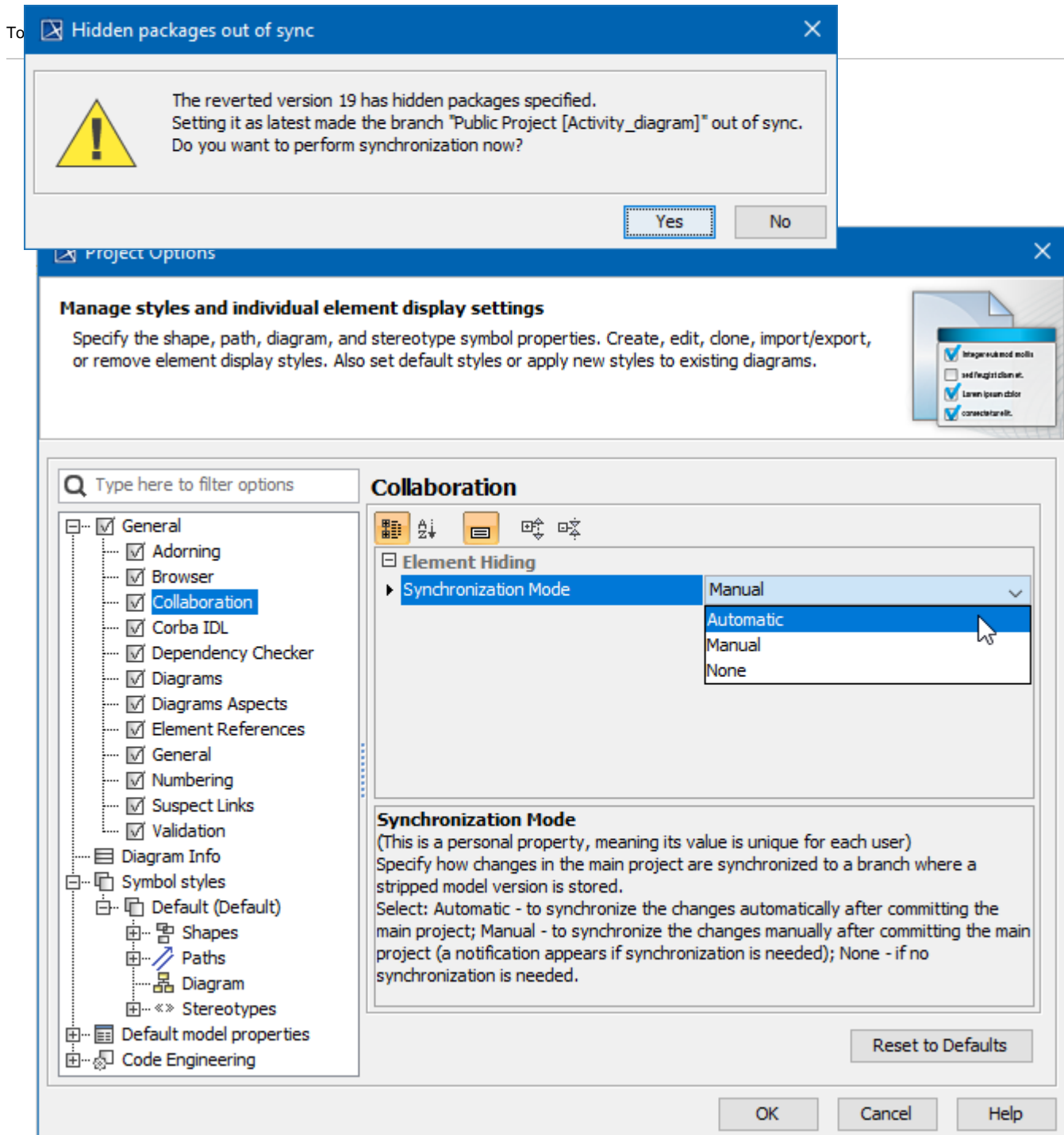
- Upcoming Changes browser panel,
- Local changes browser panel,

- Commit dialog,
- Content History panel,
- Element History.

Model synchronization

A stripped model is read-only. It can only be used for reuse in other projects, not for editing. However, you can edit the main project. On committing, checking is performed to see if the changes affect hidden and public parts of the model, and model synchronization is initialized. In the **Project Options** dialog, you can define the model synchronization mode. The default mode is **Manual**; that is, after committing project changes, you will be notified about local changes affecting a public model stored in a dedicated branch. You can then decide either to synchronize changes or not.

⚠ The server project with hidden packages (and public branches) might get out of sync with the current version of the model in the public branch after set as the latest functionality in the **History** dialog is used. Therefore, you will be asked to perform hidden package synchronization.

To

⊠ Hidden packages out of sync ✕

⚠ The reverted version 19 has hidden packages specified.
Setting it as latest made the branch "Public Project [Activity_diagram]" out of sync.
Do you want to perform synchronization now?

[Yes]   [No]

⊠ Project Options ✕

Manage styles and individual element display settings

Specify the shape, path, diagram, and stereotype symbol properties. Create, edit, clone, import/export, or remove element display styles. Also set default styles or apply new styles to existing diagrams.

Q Type here to filter options

- ⊟ ☑ General
  - ☑ Adorning
  - ☑ Browser
  - ☑ Collaboration
  - ☑ Corba IDL
  - ☑ Dependency Checker
  - ☑ Diagrams
  - ☑ Diagrams Aspects
  - ☑ Element References
  - ☑ General
  - ☑ Numbering
  - ☑ Suspect Links
  - ☑ Validation
- ☰ Diagram Info
- ⊟ 🗗 Symbol styles
  - ⊟ 🗗 Default (Default)
    - ⊞ 🗗 Shapes
    - ⊞ ⁄ Paths
    - 🗗 Diagram
    - ⊞ «» Stereotypes
- ⊞ ☰ Default model properties
- ⊞ 🗗 Code Engineering

Collaboration

⊟ **Element Hiding**

▶ Synchronization Mode	Manual ∨
	Automatic
	Manual
	None

Synchronization Mode
(This is a personal property, meaning its value is unique for each user)
Specify how changes in the main project are synchronized to a branch where a stripped model version is stored.
Select: Automatic - to synchronize the changes automatically after committing the main project; Manual - to synchronize the changes manually after committing the main project (a notification appears if synchronization is needed); None - if no synchronization is needed.

[Reset to Defaults]

[OK]   [Cancel]   [Help]

Selecting a synchronization mode to specify how changes in the main project are synchronized to a branch where a stripped model version is stored

Related pages

- Branching projects
- Identifying Package Dependencies
- Committing changes to Teamwork Cloud