

Applying metamodel customizations to a model package

i This information is intended for advanced users who are familiar with changing metamodel configurations.

To make the metamodel customizations available to only a single model package, apply the customizations to an existing model package with embedded metamodels. The configuration changes will remain confined to this model package.

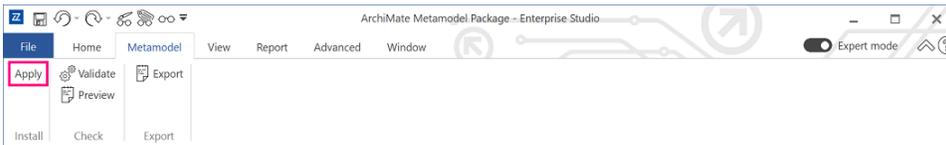
If you choose to apply your customizations to a single model package, remember to migrate this model package to the new tool configuration when updating to a new Enterprise Studio software release. Please refer to [Updating to a new Enterprise Studio release](#) for more information.

The model package you want to apply the customizations to can be the model package you initially started with when opening the metamodel but can also be also a different model package, as long as it has the same tool configuration embedded as the model package the metamodel package was started from.

i You cannot apply customizations to a regularly stored model package (XMA), or to a project. A project is a collaboration based on a model package. Any desired changes in its configuration need to be made based on the master model package. After that, a project can be updated with the configuration contribution of the master model package.

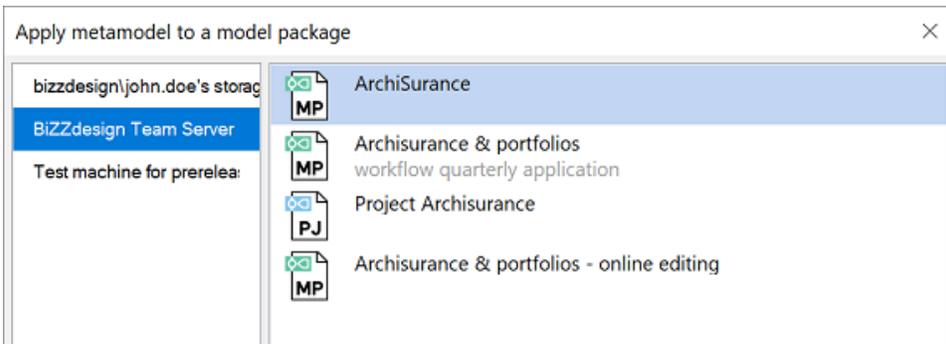
To apply metamodel customizations to a model package:

1. On the **Metamodel** tab, in the **Install** group, click **Apply**.



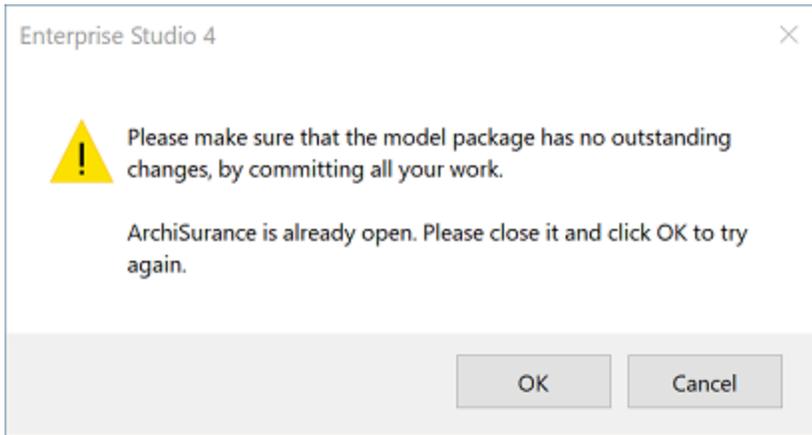
If you have not yet [validated](#) the customizations, validation is automatically performed first. If validation fails, a window appears showing the number of errors and warnings. Details of the errors and warnings are shown in the **Messages** window. Fix the problems, and validate your customizations before applying them.

2. Select the model package you want to apply the customizations to, and click **OK**.



3. Depending on the situation, the customizations are directly applied to the selected model package (may take a few seconds), or first a message window appears indicating:

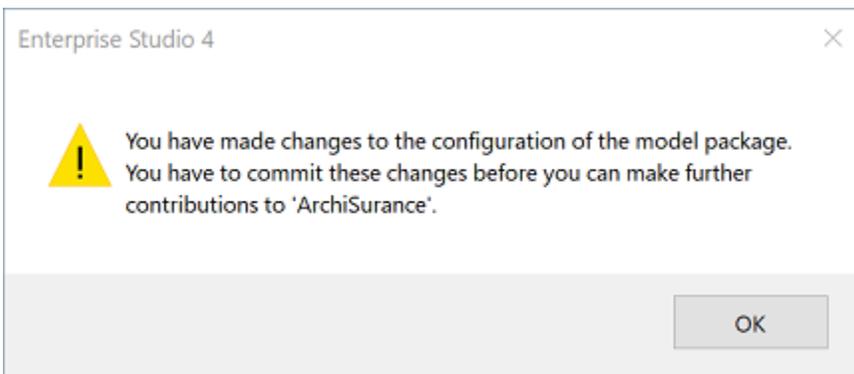
- that the selected model package is open and needs to be closed first, or
- that the selected model package has outstanding changes that need to be committed first, or
- a combination of both:



If you have not yet set a custom version for the metamodel, you will also see a message about that. However, it is optional to set a version.

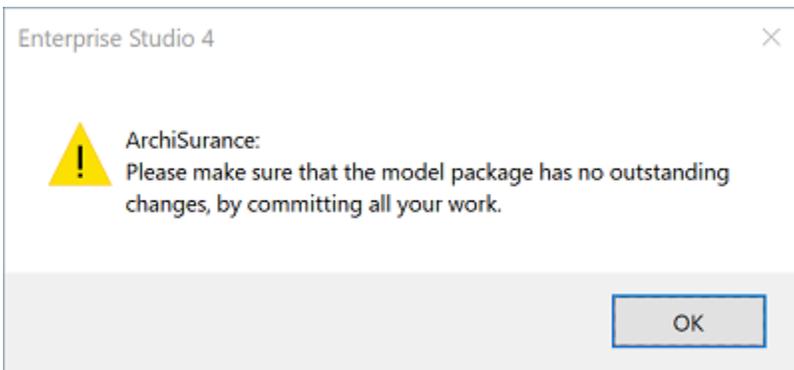
Perform the needed operation(s), and click **OK** in the message window. The customizations are now applied to the selected model package.

4. After completion, a confirmation message appears. Click **OK**.



The model package with the customized configuration is now opened in a new instance of Enterprise Studio.

If you first see the following message instead of the one above, it means that the model package you just closed still has uncommitted changes. Reopen the model package, commit the changes and close it again. Then retry to apply the changes.

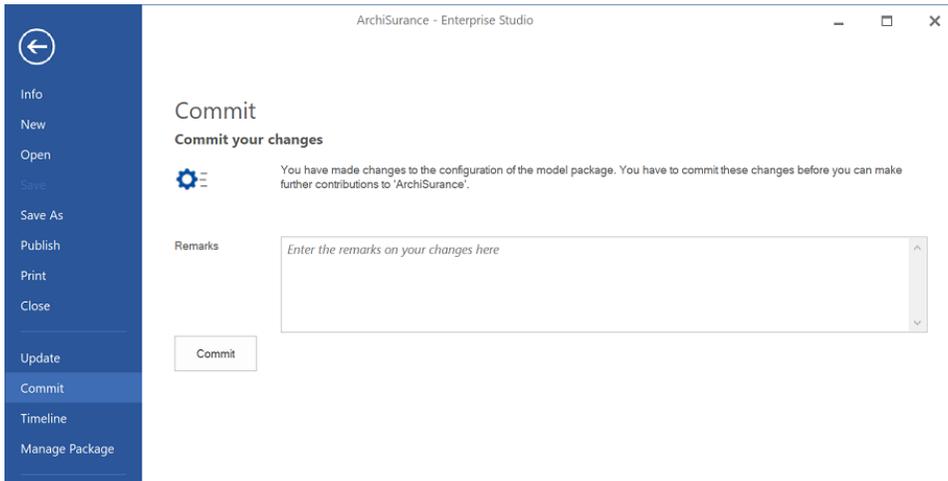


5. Depending on the type of the model package, continue as follows:

If the model package is a *tracked* package (stored in your personal storage), you are now ready. Close the metamodel package without saving, and continue working on the model package with the changed configuration.

If the model package is a *shared* package (stored on the Team Server), share the customizations with the other users:

- a. On the **File** tab, click **Commit**.



- b. In **Remarks**, type a comment about the configuration changes to clarify the commit, and then click **Commit**.

After you have committed the customizations, your contribution will be available for other users to include in their copy of the model package.

- c. Close the metamodel package without saving.

Related articles

- [Assigning a custom version to a metamodel](#)
- [Validating and previewing metamodel customizations](#)
- [Temporarily saving a metamodel package](#)
- [Exporting metamodel customizations to the tool configuration](#)

Troubleshooting

- [Metamodeler troubleshooting](#)