Contents

*This document has been revised to align with **Archicad 28.1.0 and Library Update 2.***

Using Library Packages

Archicad 28 introduced a new library format: the library packages. Here, you can find detailed documentation on what changes this new technology presents to you as an Archicad user or a third-party library developer.

Library Packages

Up until Archicad 28, the official library was shipped in .lcf format. Each regional version had two .lcf files: Archicad Library and MEP. These files contained different library parts, each with region-specific translations, folder structure, and default settings, matching the default template files.

On the contrary, Library Packages are smaller parts of the whole library: the previous two .lcf files are substituted with several Library Packages. These packages contain the same library parts for each regional version and all localization (translation, default settings, folder structures) knowledge, so they are suitable for multiple countries.

Changes with Library Packages

By default, the official Archicad templates contain a predefined set of Library Packages, including all objects previously available for each regional version of Archicad.

If you are a general user who does not want to explore the new technology, you can continue using the Library Packages the same way you used the previously used .lcf files. However, if you are reading this page, you are likely hoping to reap the packages' benefits, so let us see how they affect more advanced work.

Changes in regular workflows

This new technology affects the appearance of the Library Manager (*Changes in Library Manager*) and the Tool Settings Dialogs (*Changes in Tool Settings Dialogs*).

You will also discover that more library parts are available for all regional versions of Archicad and that even more library packages can be loaded via the Library Manager from the Archicad Library Packages folder (e.g., different Doors and Windows sets are available by loading different library packages).

Additionally, the library-related settings on the Model View Options window are separated into more pages.

Additional workflow options

New workflows become available with library packages and their flexible localization options. These were previously hard-coded into all released library versions but now can be changed during Archicad runtime:

- **Virtual folder structure**: All localization versions can define alternate virtual folder structures, and users can switch between these in Archicad. The official library has the same folder structures defined for each regional version as users are already used to. Read more on this in *Changing .libpack language*.
- **Translation for library-related content**: This includes the folder and library part names and all strings contained in the library parts, which enables you to use the library in the language of your choice, regardless of which Archicad version you own. Custom libraries can now also have translations for a broader market. Read more on this in *Changing .libpack language*.
- **Setting default parameter values**: With Mapping Value Tables, a simple text file, you can change the default values of library part parameters, making creating custom templates much easier as the library can match the alternate values. The official templates contain a Mapping Value Table with a complete list of editable default values, so you do not have to add anything manually; only edit the existing data (read more on this in *Setting library part defaults*). This workflow can be achieved with custom libraries, too.

For third-party library developers

Suppose you would like to incorporate any localization knowledge (mentioned above in the *Additional workflow options* section) into your custom-made libraries; you can upgrade your .lcf files to behave like a library package by following the step-by-step documentation in Create a new libpack from scratch / upgrade .lcf to libpack. You can find links to detailed descriptions of each required additional file there, but reading through all the user guides (linked above) is highly recommended.

Setting library part defaults

One of the main benefits of using the library package technology is setting library part defaults in Archicad runtime. While previously you could achieve this only by using Favorites as a workaround, with library packages you can now set the default values of most* of the parameters in each library part. Thus, from now on library defaults can match any custom template.

***The main exceptions are:**

- Hidden parameters, because changing them can cause errors in the library
- Most macro parameters are set by caller objects, so changing those default values has no effect
- Array parameter values, because changing the array size via the mapping table could lead to errors

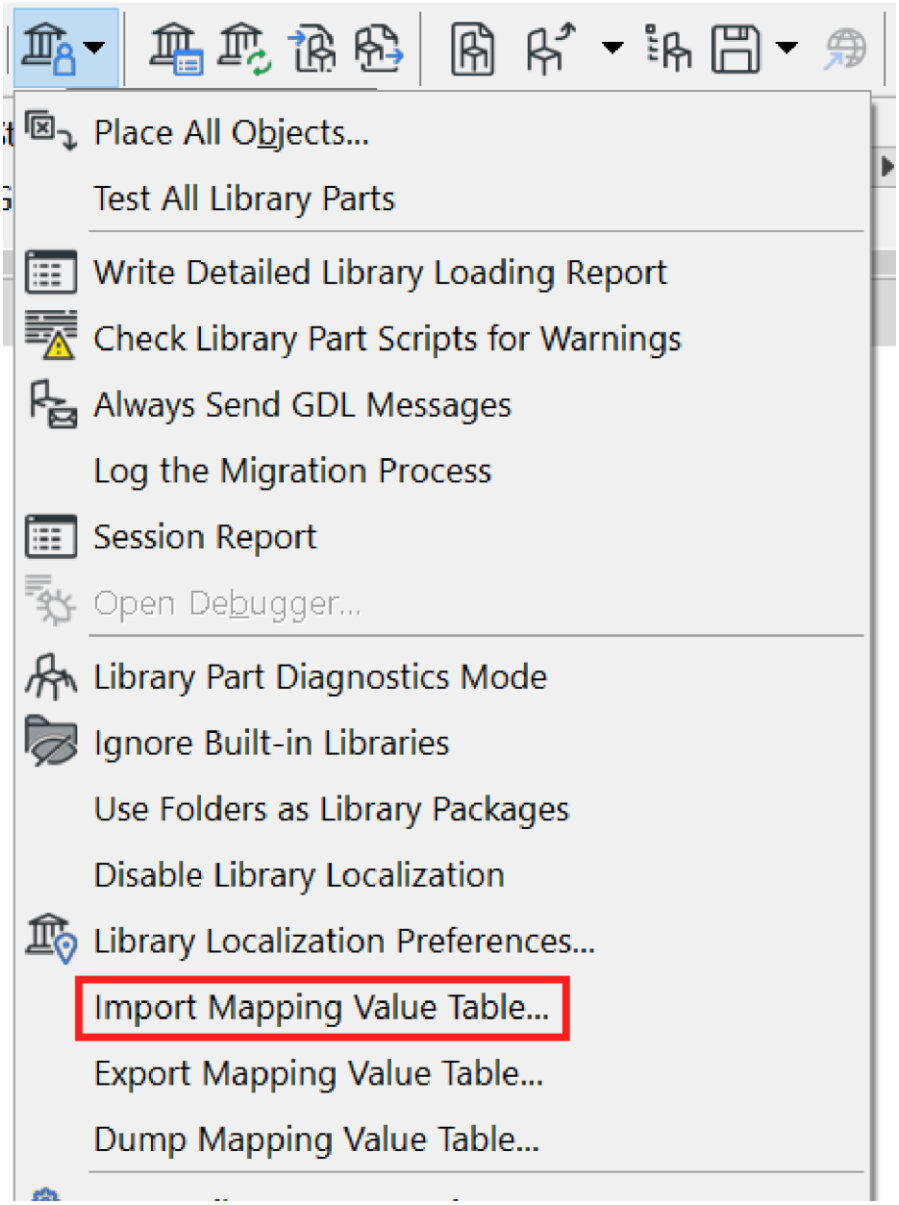
Also note that for string parameter values, changing the default via mapping values has priority over the translation (symbolStrings.po).

Changing defaults in Archicad

Library packages use **Mapping Value Tables** to define each parameter's default value. These tables are saved into Archicad plan or template files; Graphisoft's default templates also contain such a table.

To import Mapping Value Tables:

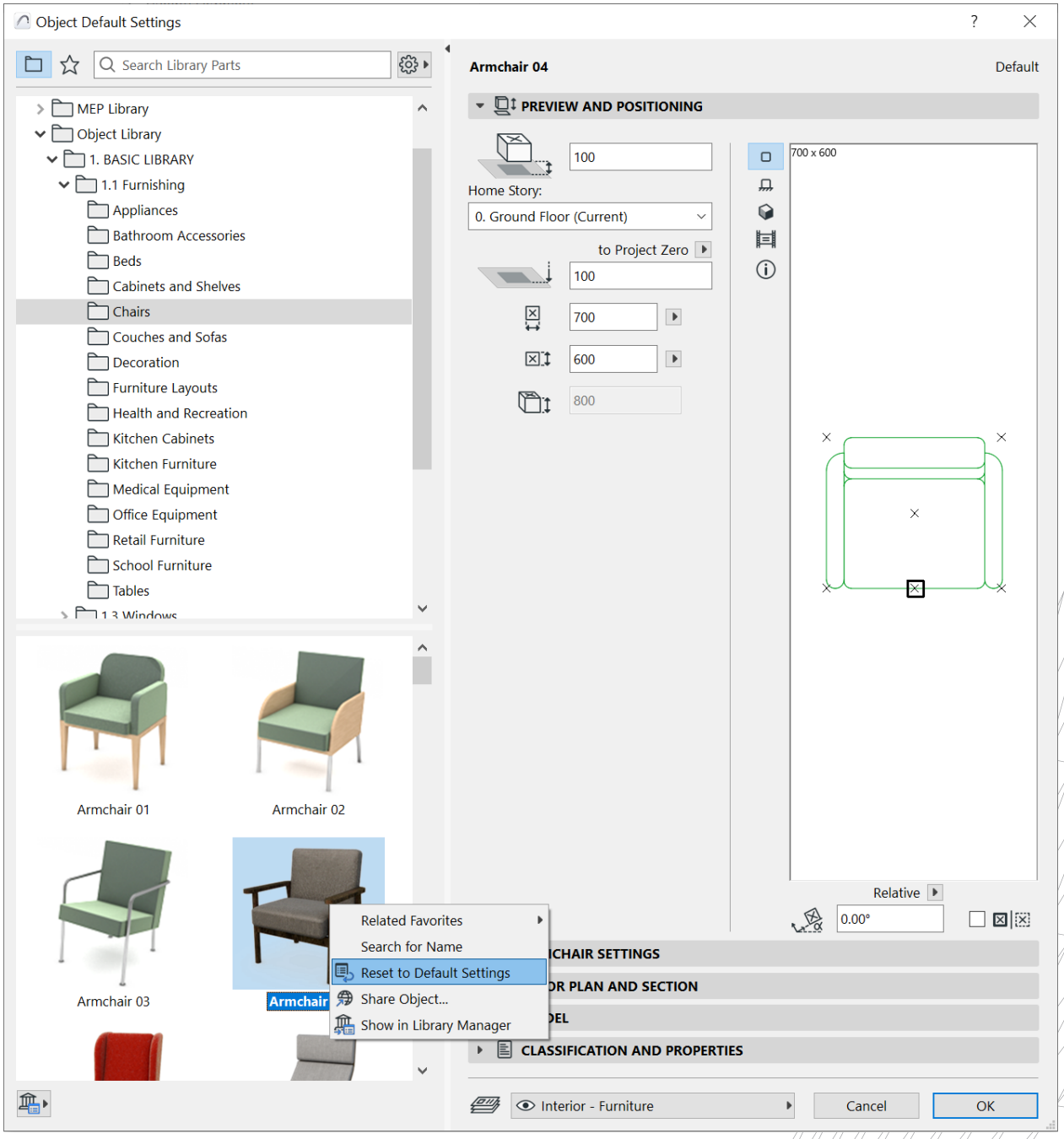
- Open the Edit GDL Library Parts toolbar through *Window/Toolbars/Edit GDL Library Parts.*
- Click on the dropdown button on the left end of the toolbar.
- From the dropdown, select Import Mapping Value Table…



- Browse for a Mapping Value Table .json file.
- Click Open.
- Decide what you would like to do with the already existing Mapping Value Table:
    - Replace current table → removes previously loaded Mapping Value Table and loads the currently selected one.
    - Keep original on conflict → merge the previous and current tables; the previously loaded table will define the duplicate values.
    - Replace the original on conflict → merge the previous and current tables; the new table will define duplicated values.
- Click on OK.

This only affects library part *defaults*, so **already placed library parts will not change**. For the changes to take effect, you may have to **reset the currently selected library parts** in the tool settings dialog:

- Either select another library part, then change back.
- Or, in the tool settings dialog, right-click on the preview picture of the library part and select Reset to Default Settings.



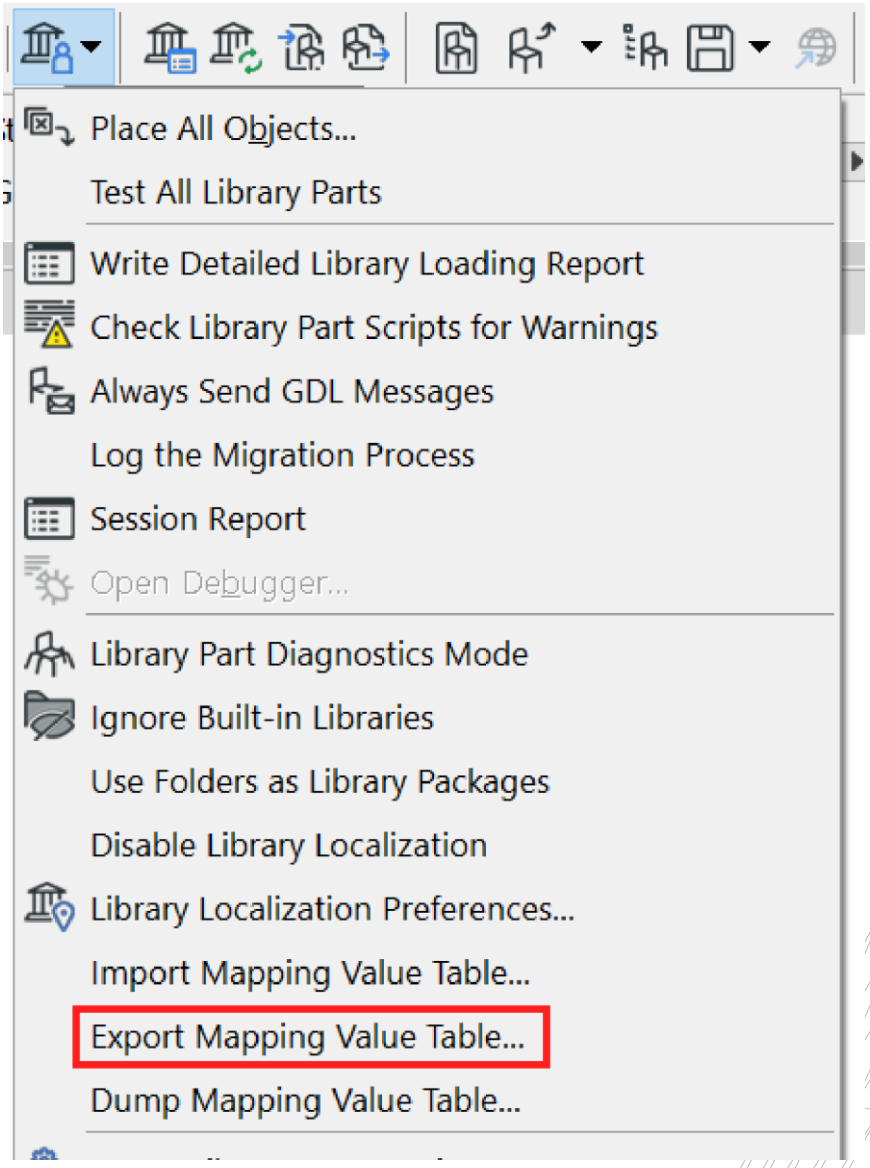
Editing Mapping Value Tables

Now that you know how to exchange the Mapping Value Tables in Archicad, you may be eager to create your own. The easiest way to do this is to modify an existing table.

For the official library provided by Graphisoft, you can download the Mapping Value Tables from here the [Dowload site](#).

Another way to obtain an editable Mapping Value Table is to export it from Archicad:

- Start a new project using the default template.
- Open the same dropdown on the Edit GDL Library Parts toolbar as before.
- Select Export Mapping Value Table…



- Save the MappingValue.json file.

Now open the saved .json file in a text editor. You can see it is long: it contains all* parameters with their meaning name, parameter type, and value.

*The same exceptions apply as before. Also, "*all*" means this list cannot be expanded without editing the library.

A single entry in the .json file looks like this:

```
{
  "meaning": "Armchair 01/gs_fill_pen/Furnishing_FillPen",
  "type": "PenColor",
  "value": 91
},
```

As you can see, each mapping value has a meaning name, a type, and a value. You can use any of these to search for what you need. (e.g., you can look for PenColor type parameters with the value of 91; with a find and replace, you can change them all to a different value).

You can also set Windows or Mac-specific values using "macValue" and "winValue". This is mainly used for font types, as the two operating systems have different font type sets:

```
{
  "meaning": "2-way Exterior Door/Font/gs_FontStyle",
  "type": "String",
  "macValue": "Lucida Grande",
  "winValue": "Arial Unicode MS"
},
```

Edit only the values after the "value" tag (with an appropriate value for their type). Otherwise, the file may not work properly!

If you wish to know how to add parameters to this list - either for creating your own library or improving the official one - see the detailed development workflow in How Editable Parameter Defaults Work.

When you are done editing the Mapping Value Table, save the new version. After that, you can import it into Archicad, as described before.

Changing .libpack language

One of the main benefits of using library packages is the ability to change their display language in Archicad runtime. The display language includes the string appearing on the user interface of each object and how they are organized in the folder structure (because different regions might have different preferred folder structures).

Archicad includes 29 predefined language options for which translations and folder structures can be given. The official library packages created by Graphisoft include knowledge of each of these languages, but third-party packages may only contain some of these languages. This is why the language selection is not necessarily a single choice but a priority list. If the library package includes data for the first language in the list, it will be applied. If not, it checks for the second priority language, and so on.

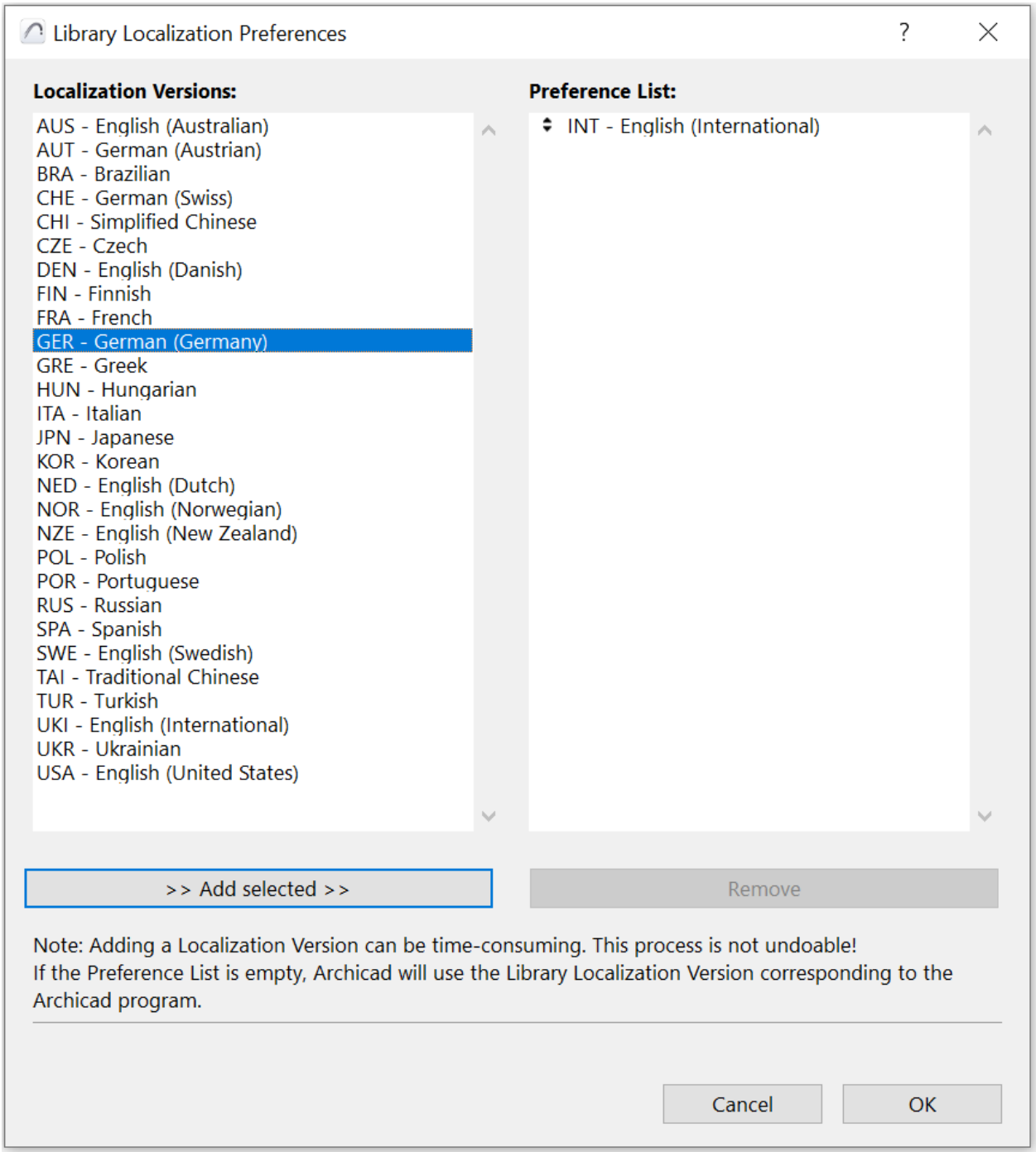
The language selection will not be applied if a library package does not contain dictionaries or pathnametables. Read these documents on creating each of the required files:

Creating symbolStrings.po

Creating pathNameTable.json

Creating fileNames.po and folderNames.po

To change the language preference in Archicad, open the dropdown menu at the left end of the Edit GDL Library Parts toolbar (Window/Toolbars/Edit GDL Library Parts, and select Library Localization Preferences.



Here, you can see two boxes: the left contains all available language versions (which are not moved to the Preference List), and the right includes the ranked list of language versions that affect your plan file.

To add a language version to the Preference List, select it and click the Add Selected button at the bottom. Once the language versions are added to the box on the right, you can rank them with drag-and-drop.

Be warned (as the text on the bottom notes) that applying these changes takes time and cannot be undone. You can edit it again manually in the same dialog.

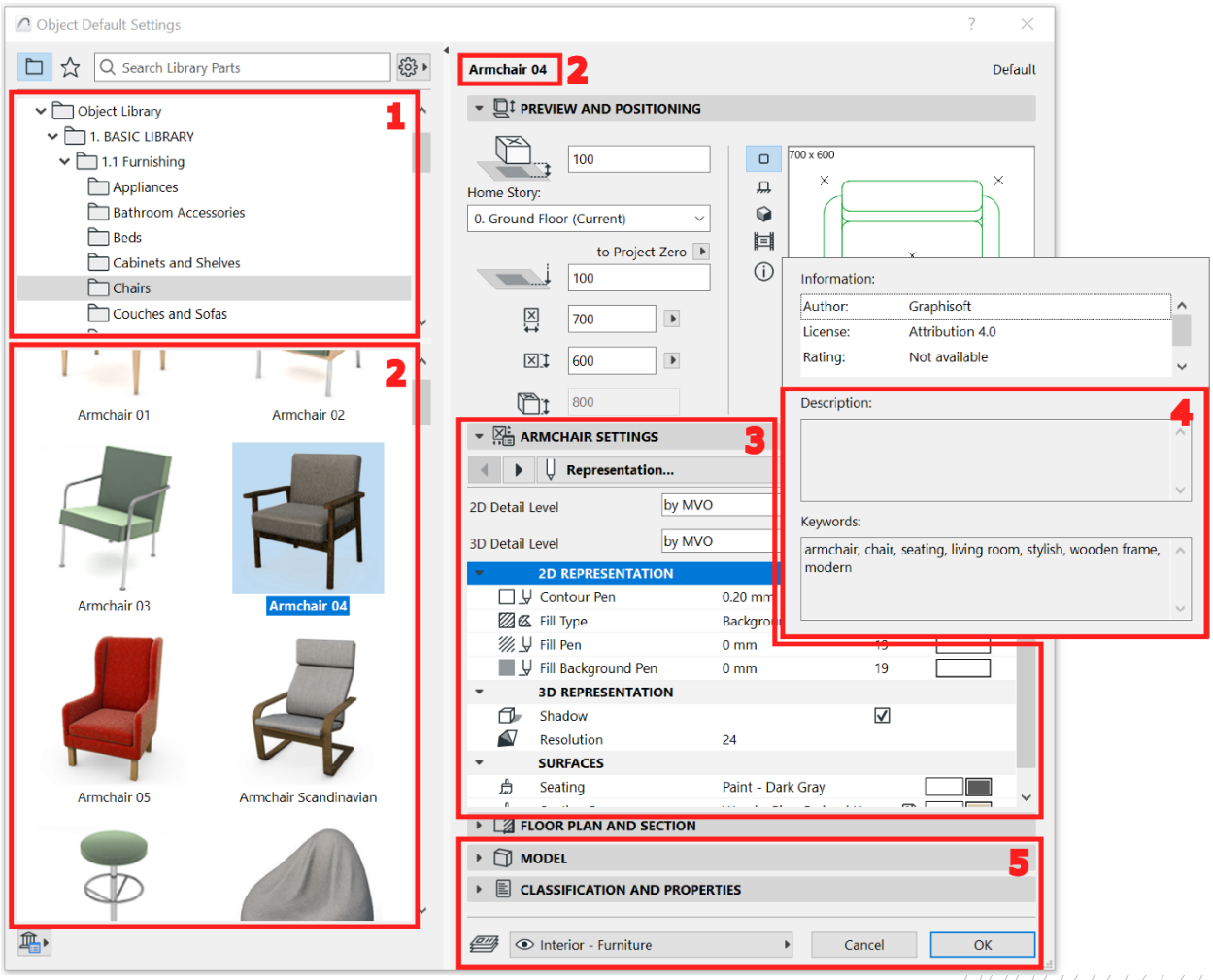
The changes affect the following:

- Name and structure of the library part folders (1)
- Name of the library parts (2)
- Settings tab of the library part (3)
- Keywords and Description (4)

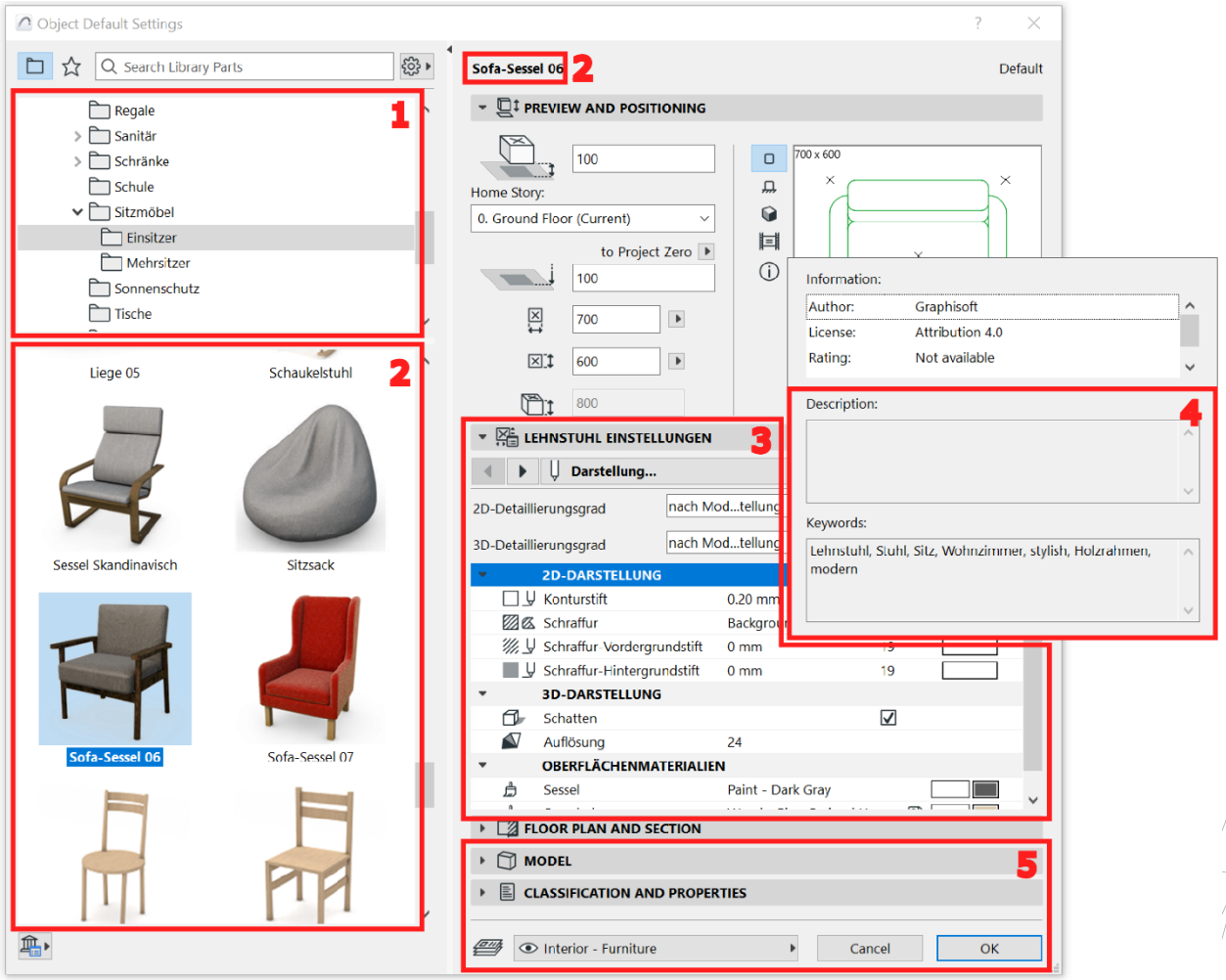
Unaffected:

- Interface parts generated by Archicad exe (5)
- String parameter values in already placed library parts (unless parameter script is run)

English/International (INT) version



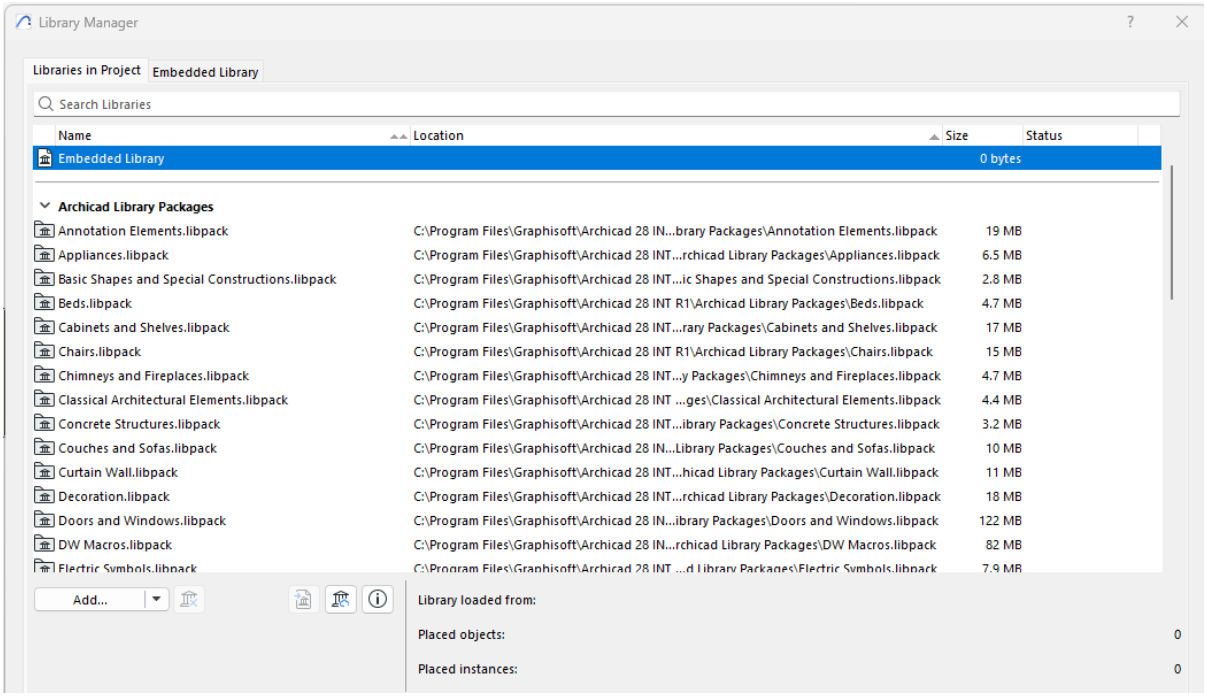
German (GER) version



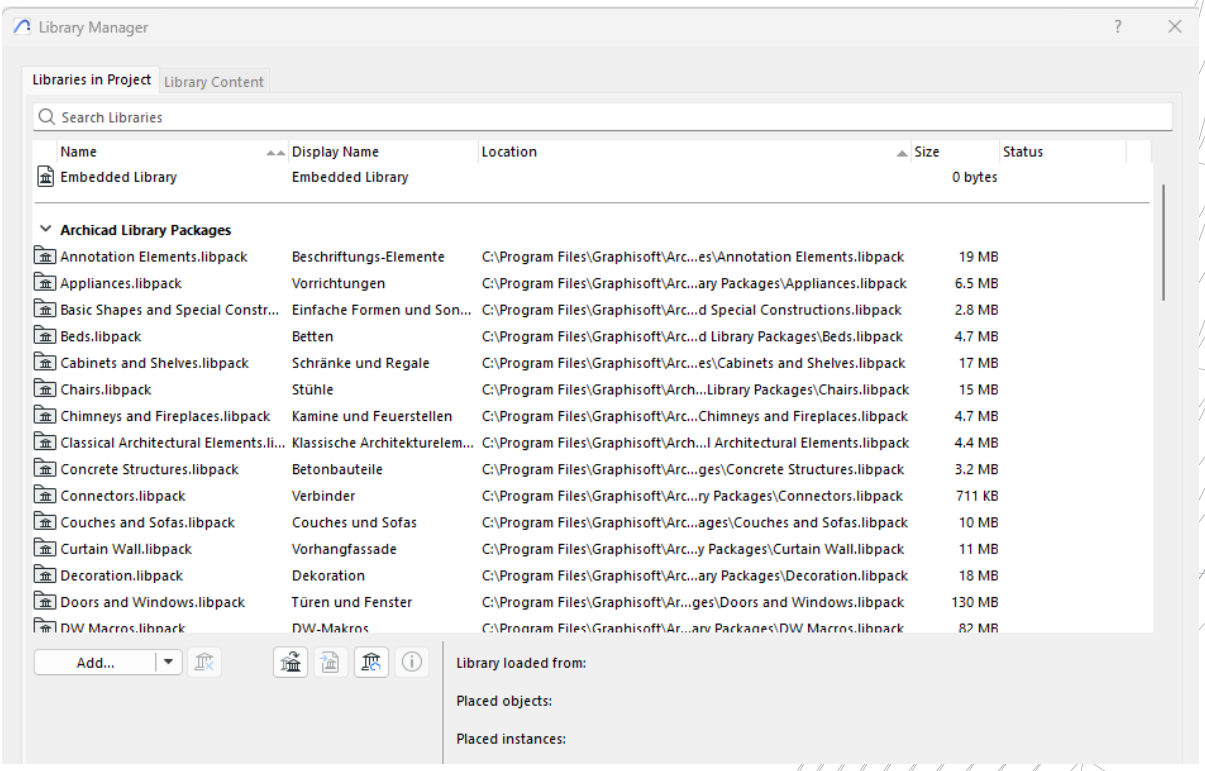
Archicad 28.1.0 and Library Update 2:

The Library Localization Preferences setting also affects the library package names in the Library Manager and in the Tool Settings Dialogs.

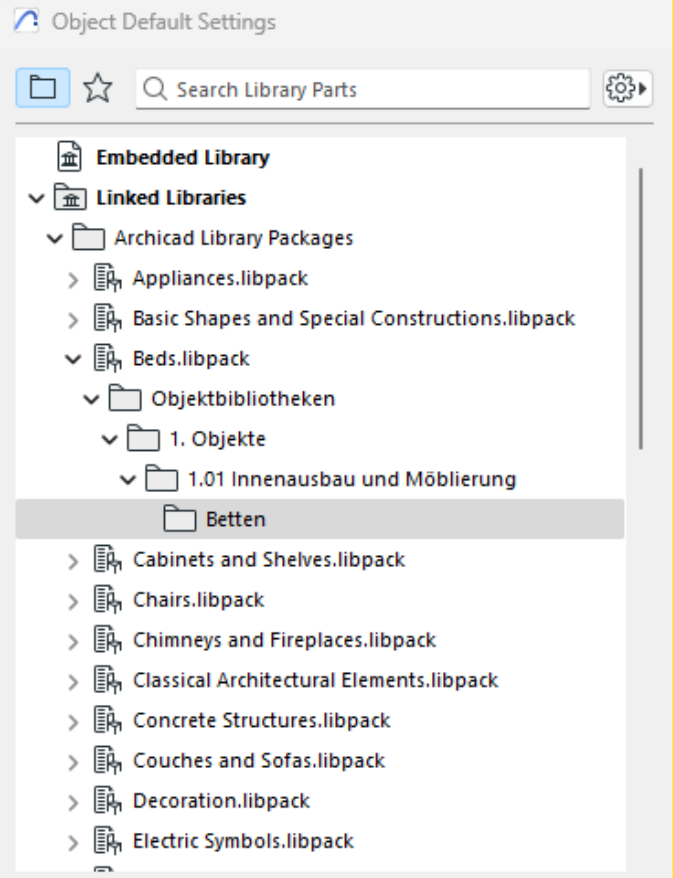
Library Manager German (GER) version – before update



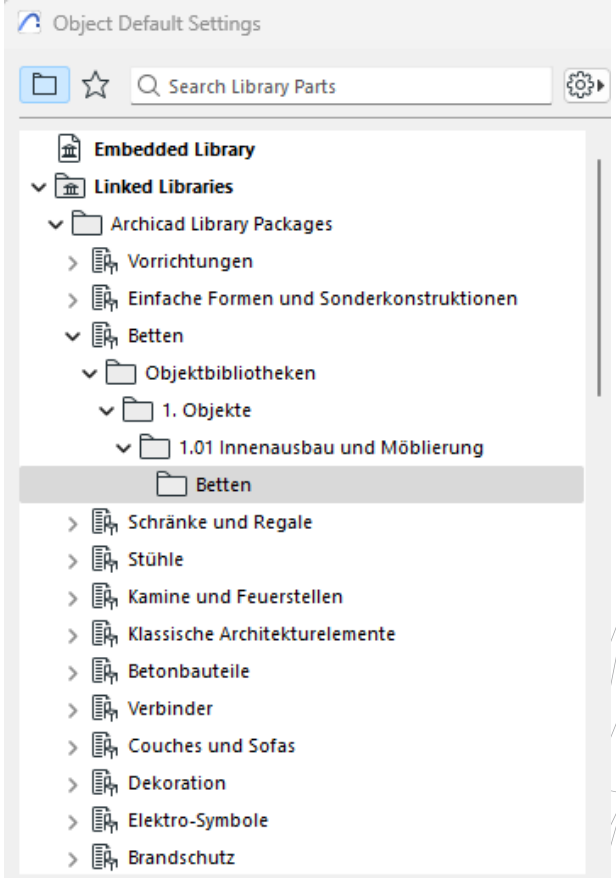
Library Manager German (GER) version – after update



Tool Settings Dialog German (GER) version – before update



Tool Settings Dialog German (GER) version – after update



Read the following document on how to create/modify the library package files to implement the new function:

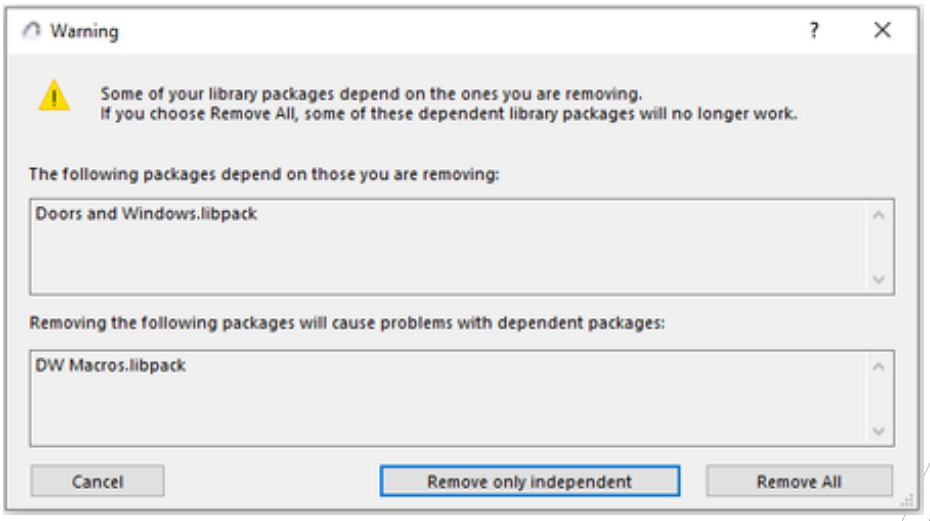
[Creating localizedPackageNames.json]

Changes in Library Manager

As mentioned, Library Packages are smaller parts of the entire Archicad Library. To avoid code duplication and increase in library size, most of the common code (macros, MVOs, and images) are stored in "central" library packages (e.g., the Essential Library), accessible by all other library packages. For these "dependent" packages to work, their dependencies (that is, the "central" packages they depend on) must also be loaded into the project. The Library Manager handles these dependencies automatically (see below), so all library packages can access all required code.

When you add a library package to your project in the Library Manager, all its dependencies are automatically added (as long as they are in the same folder). Archicad also lists the added library packages in an info window.

If you try to remove a library package on which other packages depend, a warning window pops up:



If you select multiple packages, and if some of those packages still have dependencies loaded, you can avoid removing the dependency packages by selecting **Remove only independent**.

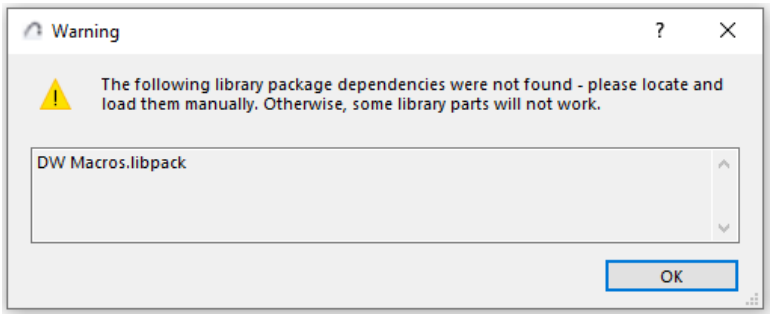
- For example, when designing an office building and wanting to remove unnecessary Beds and Medical Equipment packages, Archicad will warn you that the Furniture Layouts library package depends on the Beds package. If you have already placed some elements from Furniture Layouts, those library parts won't work if you remove Beds. In this case, select **Remove only independent**, which will remove the Medical Equipment package but keep the Beds package - so that Furniture Layouts can still work.

You can choose Remove All if you don't mind the missing dependencies (temporarily).

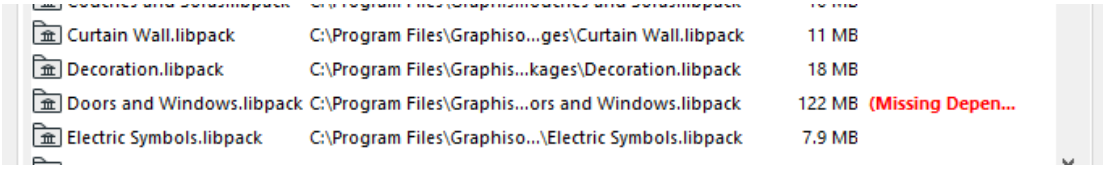
- For example, you have a modified version of the Tables package. To make it work, you had to edit the Furniture Macros package. Now you select these two packages in the Library Manager and then delete them. Archicad warns you that Furniture Macros is a dependency for other library packages as well, but you choose Remove All. After this, you can load your modified version of Furniture Macros (and Tables), which will also solve the dependency warnings.

If a library package is missing its dependency, multiple warnings appear:

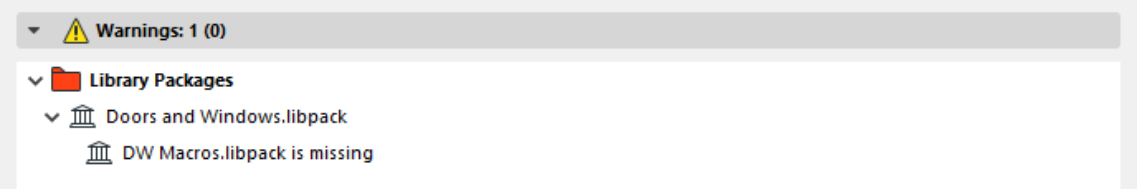
- When clicking on Reload and apply



- In the Library Manager's list of loaded libraries



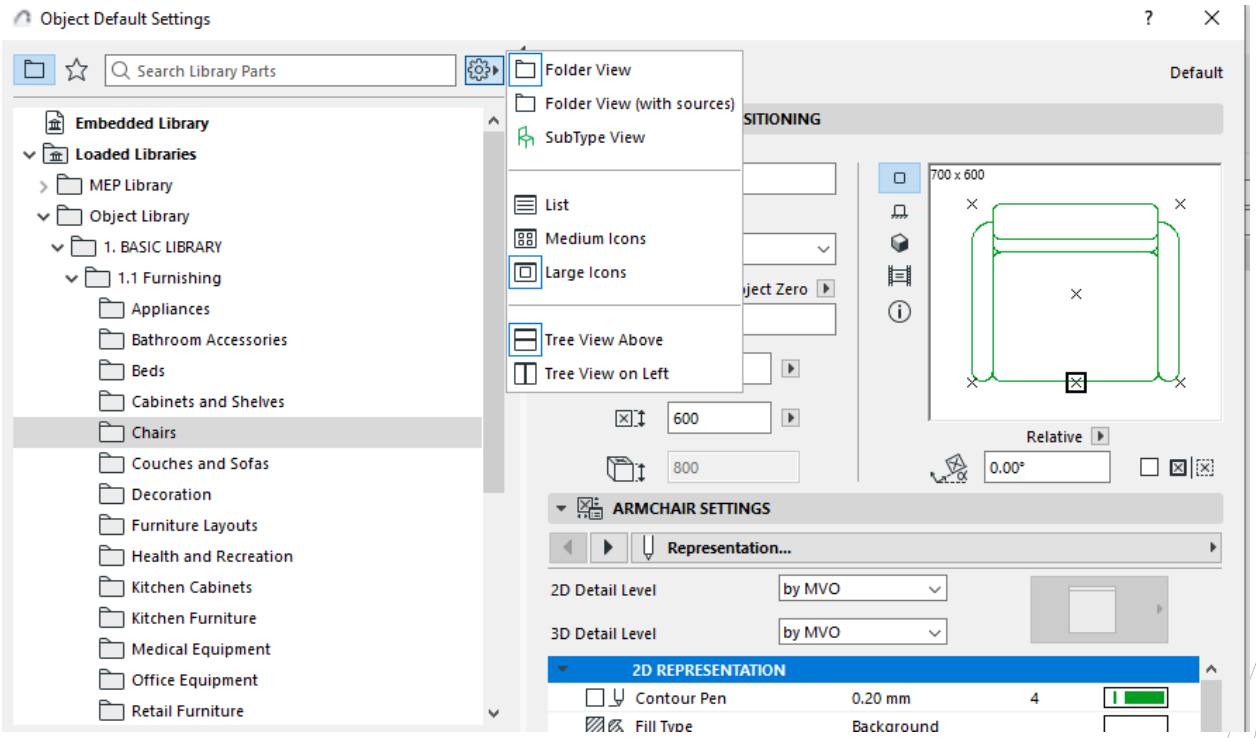
- A library warning appears at the bottom of Library Manager, and the Library component in Action Center also turns red



You can avoid these warnings by either (re-)adding the missing dependency or removing the dependent library package.

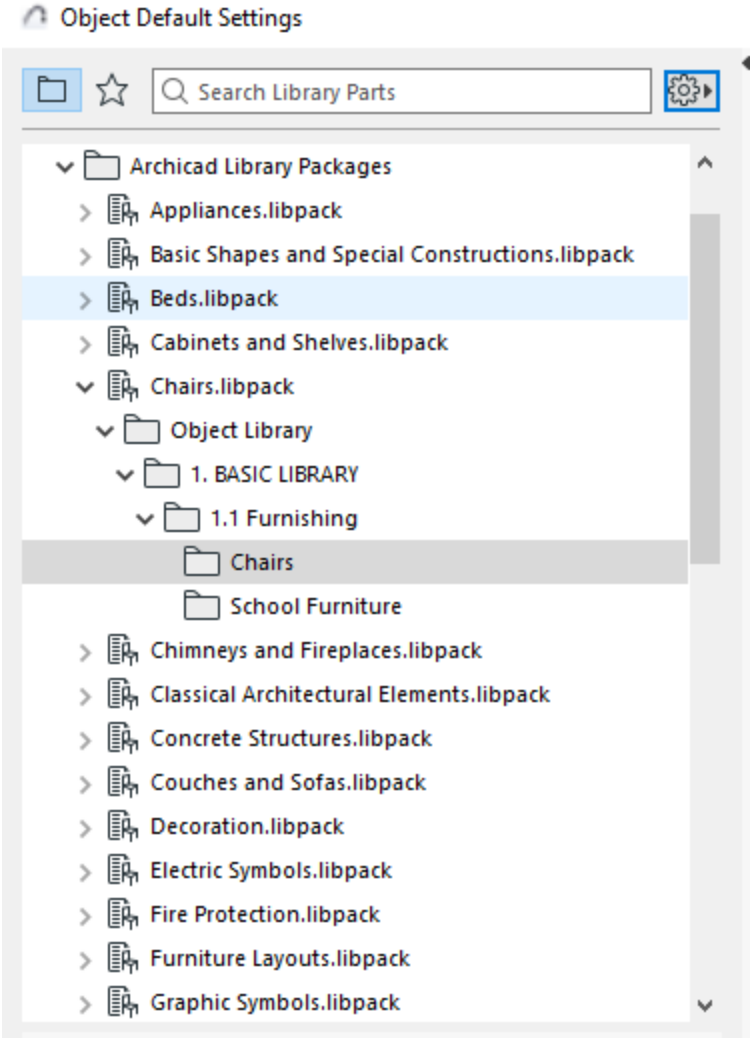
Changes in Tool Settings Dialogs

If you open any tool settings dialog, you will see the same folder structure as before. You can change the view options (e.g. folder view, list vs. icons) by clicking the cogwheel icon next to the search bar.



A new view option is available: **Folder View (with sources)** displays the structure of the library packages.

This shows you which library parts are included in which library package. If you have library packages in multiple folders, those parent folders are visible here as well (e.g., Archicad Library Packages), to help you further organize your project.



This tree is made up of two different types of data:

- **"Physical file locations"** (the folder structure on your computer), which determines the structure from the root folder to the library packages' folder level
- **"Virtual file locations"** (pathNameTables, which can be custom-made and changed with the language preferences of the library: *Changing .libpack language*), which determines the folder structure inside a single library package.

If the default option (**Folder View**) is selected, all library packages' subfolder structures are merged. In Folder view, the displayed folder structure can only be affected by the path name table files.