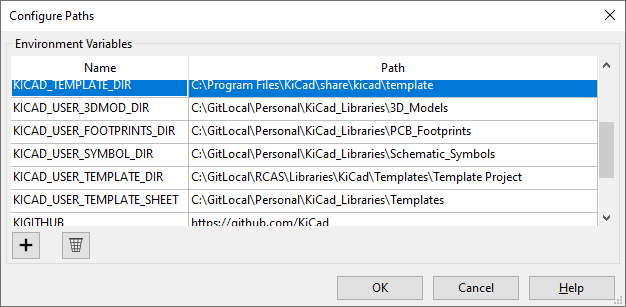
# Libraries Setup

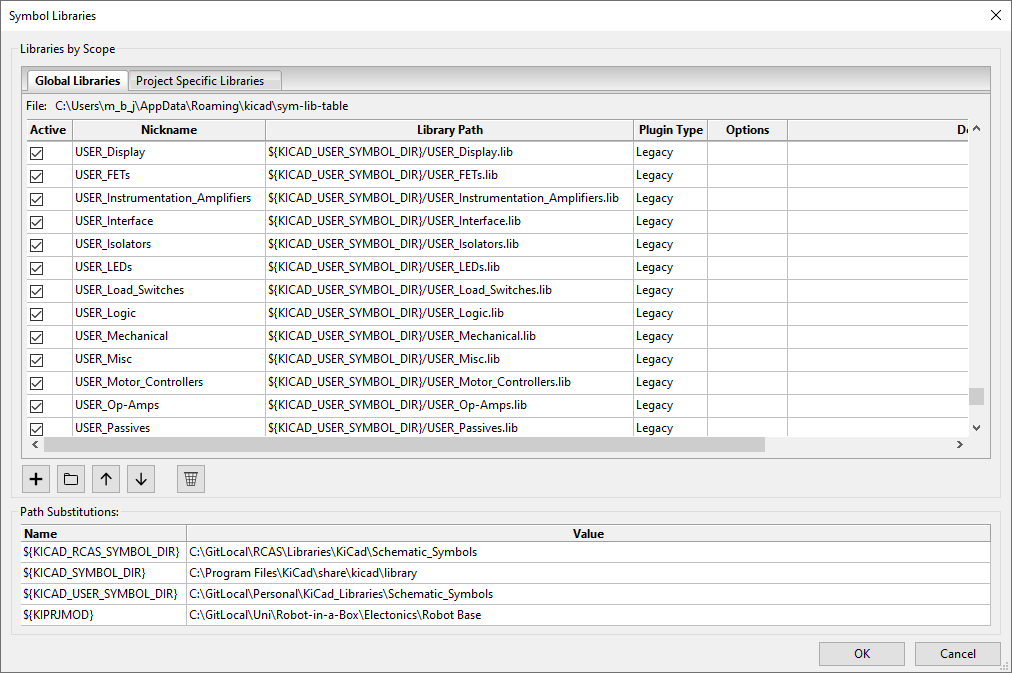
After installation of KiCad and checking out the repository for the custom [KiCad\_Libraries](https://github.com/mbjackson1/KiCad_Libraries), Environment Variables must be set up to correctly point your KiCad installation to your local copies of the libraries and templates. Open KiCad, go to Preferences > Configure Paths to open up the dialog box and insert the paths as below:

* KICAD\_USER\_3DMOD\_DIR - point to your local copy of the 3D models.
* KICAD\_USER\_FOOTPRINTS\_DIR - point to your local copy of the footprint libraries.
* KICAD\_USER\_SYMBOL\_DIR - point to your local copy of the schematic symbol libraries.
* KICAD\_USER\_TEMPLATE\_SHEET - point to your local copy of the template sheets (for PCBs and schematics).

The screenshot below shows an example Configure Paths dialog, highlighting the Paths that must be configured. In this case, the repository has been checked out under C:\GitLocal\Personal – substitute this for the location in which you have checked out the KiCad\_Libraries repository.



Once the paths are set up, within the main KiCad project window, go to Preferences>Manage Symbol Libraries. In the Global Libraries tab, use the folder icon to navigate to your local copy of the schematic symbols libraries, highlight all the .lib files and then add them. Once this is done, they should appear in your library table like this:

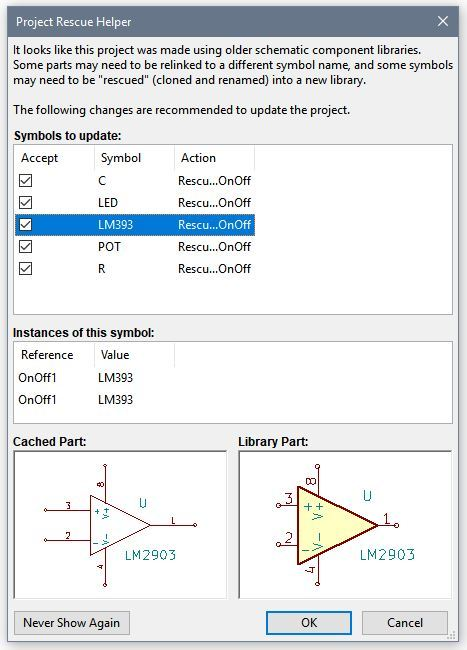


If the paths were set up correctly, then you should see the substitution of ${KICAD\_USER\_SYMBOL\_DIR} in the library table. All the libraries from the repository are prefixed with USER\_ to distinguish them from the built in KiCad libraries.

To add the footprint libraries, go to Preferences>Manage Footprint Libraries. The process is broadly similar, although adding the footprints requires highlighting the .pretty folders for each footprint (whereas for the symbols .lib files were selected).

**Things to note:**

* If upon opening Eeschema you see the rescue symbols dialog (below), then always click cancel. Check that your path variables and library tables are set up correctly (as described above) to point to the symbols. This may also have been caused by new libraries having been created that are not yet in your library table. Pull the libraries repository, then in the library table manager re-add all the libraries - it will prompt you to skip all the libraries you already have added.



# BoM Generation

* KiBoM is a much better plugin for BoM generation than the default KiCad ones, allowing variants, DNFs, and customising the fields exported. The download (use the Zip version) and documentation is here:

<https://github.com/SchrodingersGat/KiBoM>

* The plugin is added from the ‘Generate Bill of Materials’ tool. The default name is KiBOM\_CLI.
* Adding this plugin adds a bom.ini file to the project folder (if it does not already exist). This can be edited to change the settings of the BoM export file. A default bom.ini file is included in the template project in the GitHub libraries repository, and has been added to the projects. This can be adapted if different fields are needed in the BoM (see the field names used in the schematic in the ‘Edit symbol fields’ dialog.
* ‘ -d Outputs’ is recommended to be added to the command line command. This means the BoM is created in the ‘Outputs’ folder in the project directory, keeping the top level of the project tidy. A .gitignore file is set up that should prevent the majority of generated files from being added to the repository (as it should only contain source files).