

# Configuring Module Dependencies and Libraries

---

On this page:

- [Creating a module library and adding it to module dependencies](#)
- [Adding existing libraries to module dependencies](#)
- [Creating a module, project or global library and adding it to module dependencies](#)
- [Creating a library in the Project tool window](#)
- [Adding dependencies on modules](#)
- [Changing the order of dependencies](#)
- [Configuring the library contents](#)
- [Moving a module library to the project or global level](#)
- [Downloading missing library sources from the Maven repository](#)

See also, [Dependencies, Library and Configuring Project and Global Libraries](#).

## Creating a module library and adding it to module dependencies

1. [Open the Project Structure dialog](#) (e.g. **Ctrl+Shift+Alt+S**).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, click **+** and select **Jars or directories**.
6. In the [dialog that opens](#), select the necessary files and folders. These may be individual `.class`, `.java`, `.jar` and `.zip` files, and the directories containing such files. Click **OK**.
7. If necessary, select the [Export option](#) and change the [dependency scope](#).
8. Click **OK** in the **Project Structure** dialog.

See also, [Adding existing libraries to module dependencies](#) and [Adding a project or global library to module dependencies](#).

## Adding existing libraries to module dependencies

1. [Open the Project Structure dialog](#) (e.g. **Ctrl+Shift+Alt+S**).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, click **+** and select **Library**.
6. In the **Choose Libraries** dialog, select one or more libraries and click **Add Selected**.

The libraries you can choose from include:

- Global libraries
  - Project libraries existing in the current project
  - Application server libraries for the servers registered in IntelliJ IDEA
7. If necessary, select the [Export option](#) and change the [dependency scope](#).
  8. Click **OK** in the **Project Structure** dialog.

See also, [Adding a project or global library to module dependencies](#).

## Creating a module, project or global library and adding it to module dependencies

1. [Open the Project Structure dialog](#) (e.g. Ctrl+Shift+Alt+S).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, click **+** and select **Library**.
6. In the **Choose Libraries** dialog, click **New Library**.
7. If suggested, select:
  - **Java** to create a Java library. Select the files and folders to be included in the library in the [dialog that opens](#).
  - **From Maven** to download a Java library from a public [Maven repository](#). Specify the artifact to be downloaded and the associated settings in the **Download Library from Maven Repository** dialog.
8. In the [Configure Library dialog](#):
  1. Specify the library name.
  2. Select the library level (global, project or module).
  3. Configure the library contents using **+**, **☐** and **-**.
  4. Click **OK**.
9. In the **Choose Libraries** dialog, click **Add Selected**.
10. If necessary, select the [Export option](#) and change the [dependency scope](#).
11. Click **OK** in the **Project Structure** dialog.

See also, [Creating a project or a global library](#) and [Adding a project or global library to module dependencies](#).

## Creating a library in the Project tool window

Sometimes, you don't need to open the [Project Structure dialog](#) to create a library and add it to the dependencies of a module. This is the case when there are `.jar` files within your project [content roots](#).

You can select such `.jar` files in the [Project tool window](#) and use these files to create a library. When doing so, you will be able to select the library level and the module to whose dependencies the new library should be added.

In a similar way, you can use a directory containing `.jar` files to create a library.

1. [Open the Project tool window](#) (e.g. **View | Tool Windows | Project**).
2. Select the `.jar` file or files to be included in the library, or a directory that contains the `.jar` files of interest.
3. Select **Add as Library** from the context menu.
4. In the **Create Library** dialog, specify the library name, level and the module in which this library will be used.

## Adding dependencies on modules

1. Open the [Project Structure dialog](#) (e.g. Ctrl+Shift+Alt+S).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, click **+** and select **Module Dependency**.
6. In the **Choose Modules** dialog, select one or more modules the current module should depend on, and click **OK**. (Your selection is limited to the modules of the current project. That is, you cannot select the modules from a different project.)
7. If necessary, select the [Export option](#) and change the [dependency scope](#).
8. Click **OK** in the **Project Structure** dialog.

## Changing the order of dependencies

In certain cases, the order of module dependencies may be important, see [Processing Order](#).

To reorder the items, use **↑** and **↓**. Clicking these icons moves the selected item one line up or down in the list.

## Configuring the library contents

1. Open the [Project Structure dialog](#) (e.g. Ctrl+Shift+Alt+S).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, select the library of interest, and click .
6. In the [Configure Library](#) dialog that opens, use **+**,  and **-** to manage the library contents.
7. Click **OK** in the **Project Structure** dialog.

## Moving a module library to the project or global level

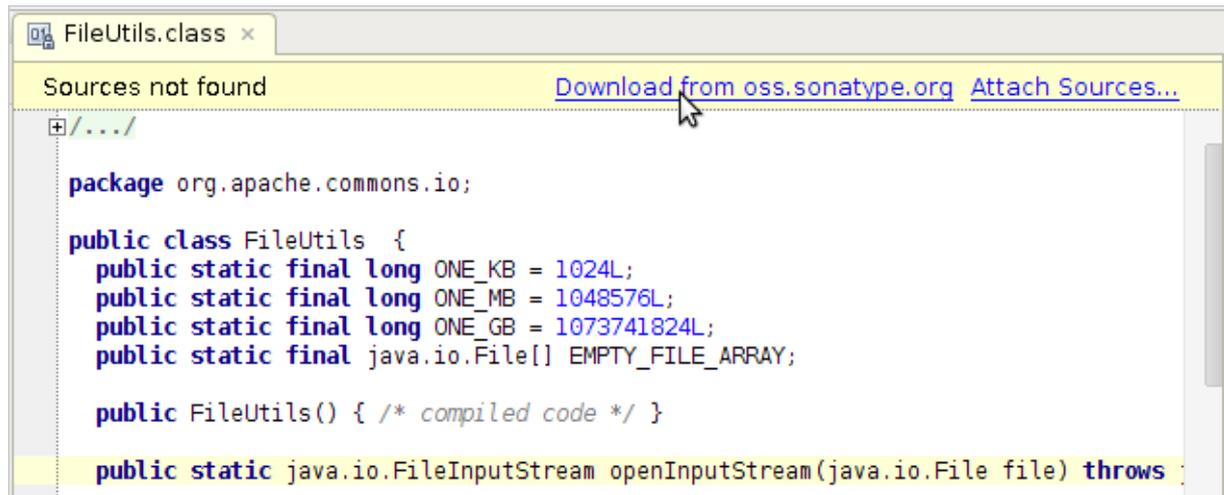
1. Open the [Project Structure dialog](#) (e.g. Ctrl+Shift+Alt+S).
2. In the left-hand pane of the dialog, select **Modules**.
3. In the pane to the right, select the module of interest.
4. In the right-hand part of the dialog, on the **Module** page, select the **Dependencies** tab.
5. On the **Dependencies** tab, right-click the library of interest, and select **Move to Project Libraries** or **Move to Global Libraries**.
6. In the **Move Library** dialog, if necessary, edit the library name. If you want to change the location of the library files, select the **Move library files** to check box and specify the corresponding location.

Note that the corresponding library is not removed from the list of the module dependencies.

## Downloading missing library sources from the Maven repository

For the library files originating from the Maven public repository, when missing sources are detected, IntelliJ IDEA suggests downloading and including those sources in the library.

To access these features, IntelliJ IDEA provides the corresponding hyperlinks in the editor.



## See Also

### Concepts:

- [Library](#)
- [Dependencies](#)
- [Processing Order](#)

### Procedures:

- [Configuring Project and Global Libraries](#)
- [Configuring Libraries of UI Components](#)

### Reference:

- [Project Structure](#)
- [Dependencies Tab](#)
- [Project Library and Global Library Pages](#)
- [Configure Library Dialog](#)
- [Application Servers](#)

### Web Resources:

- [Developer Community](#) 