

Dependencies Management (Build Path)



Redirection Notice

This page will redirect to <https://www.jetbrains.com/idea/help/configuring-dependencies-for-modular-applications.html>.

An IntelliJ IDEA Flash module contains one or more Flash build configurations. A build configuration is a set of parameters that define how to produce a SWF or SWC, an HTML wrapper or AIR descriptor, a desktop, Android or iOS package from the module sources.

At the module level, it is possible to configure only the content roots and source folders. Unlike other IntelliJ IDEA module types, Flash modules can't have any dependencies and can't depend on each other.

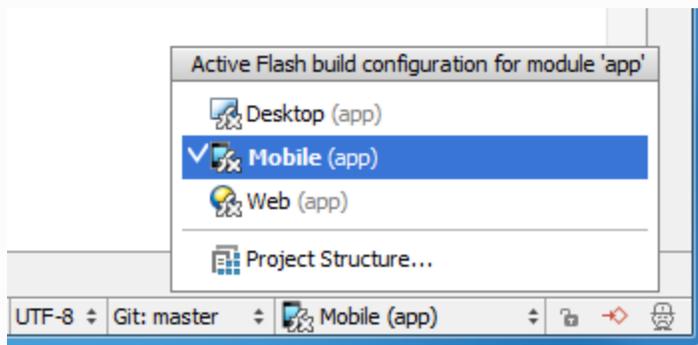
All dependencies (a.k.a. a build path) including Flex/AIR SDK and libraries are configured at the build configuration level. A Flash module can contain several build configurations each having its own set of dependencies. For example, you can have 3 build configurations that produce a web-, a desktop- and a mobile-targeted app from the same source base.



A module can have several contradictory build configurations. How is source code highlighted in such cases?

The source code is highlighted according to an active build configuration. (One of the build configurations is set active for a module.)

The name of the active build configuration is shown in the right-hand part of the status bar. To make a different build configuration active, click this name and select the build configuration of interest.



The alternative way of changing the active build configuration is by using the context menu in the editor. When working with a .as or .mxml, right-click the editing area and select Active Flash Build Configuration. Then, select the necessary build configuration.

In this tutorial, you'll find out how to:

- Configure a dependency on an external library
 - if the library is available in binary format (*.swc)
 - if the library is available only as raw ActionScript sources
- Configure a dependency on AIR Native Extensions (*.ane)
- Configure a dependency on a library that belongs to a project

1. Configure a dependency on an external *.swc library

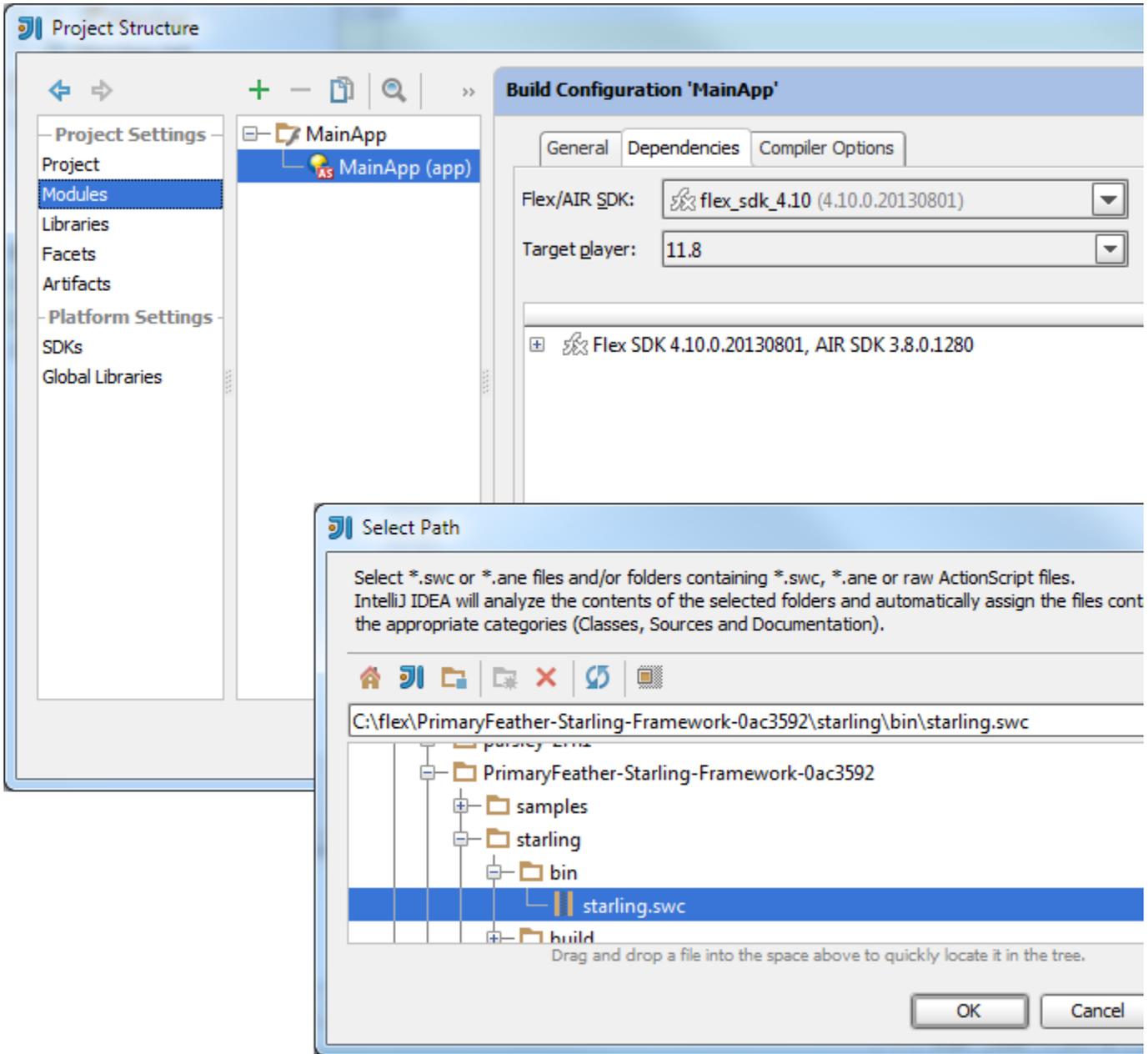
As an example, let's configure a dependency on the [Starling Framework](#).

It is assumed that you have an IntelliJ IDEA project with one Flash module that has one build configuration, and the Starling Framework has been downloaded and unpacked on your computer.

1.1. Add a *.swc file as a library

To open the Project Structure dialog, click the Project Structure icon on the main toolbar, or select File | Project Structure. Select Modules in the leftmost pane, then select your build configuration (shown as a subnode of the Flash module node) and click the Dependencies tab.

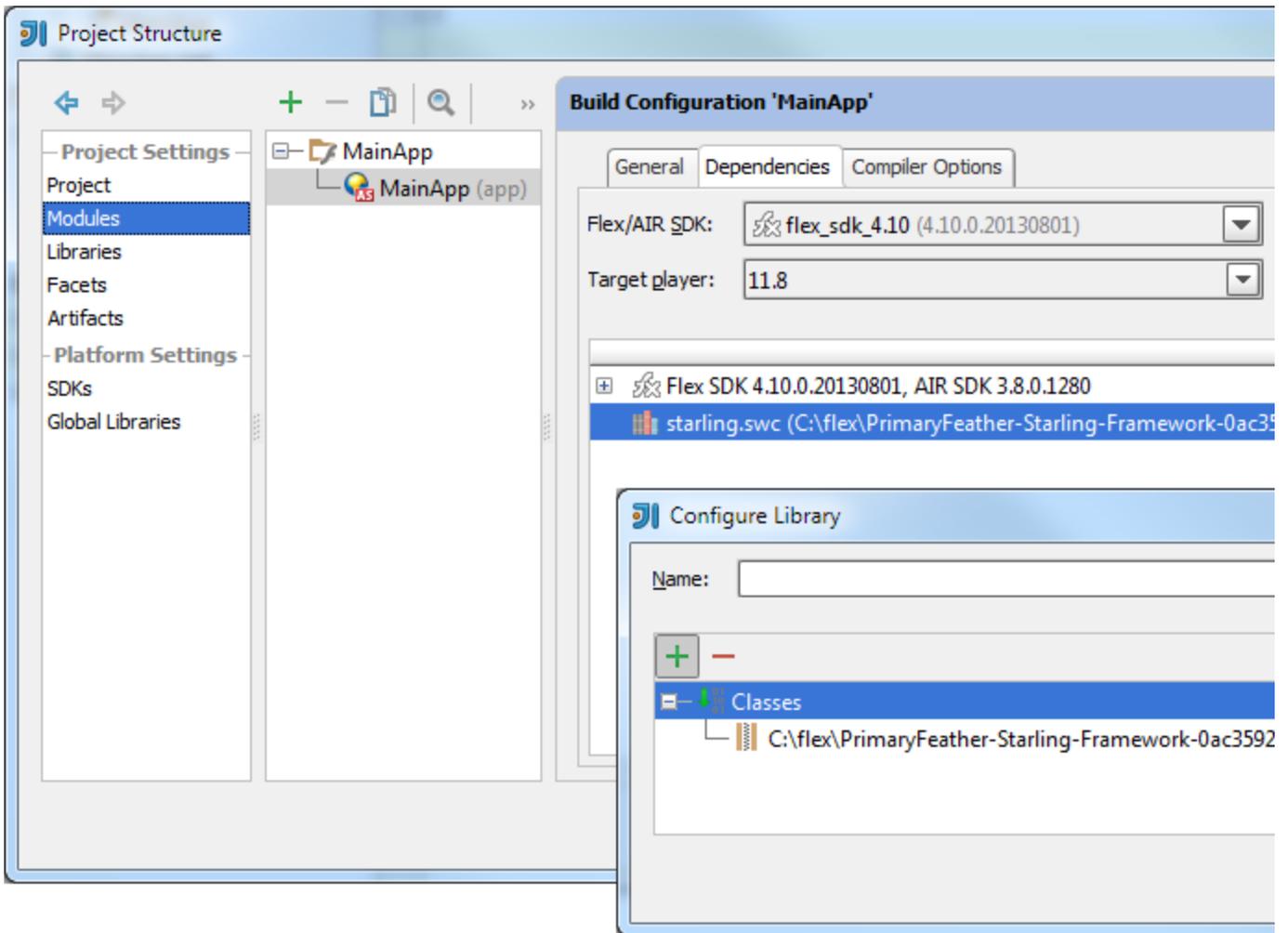
On the Dependencies tab, click [+] and select Add dependency on New Library. Then select starling.swc in the file chooser.



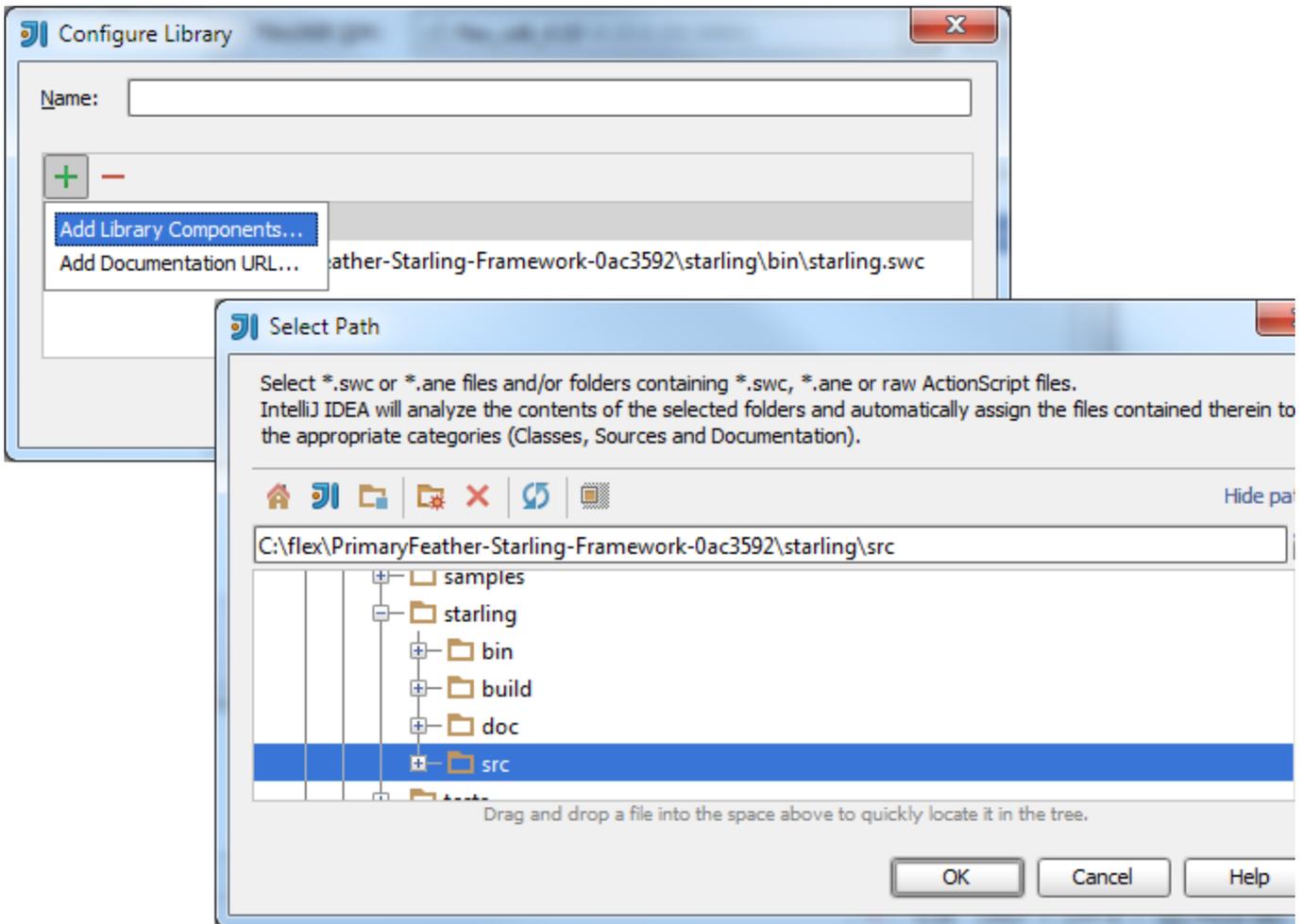
1.2. Add library sources (optional)

If the source code for a *.swc library is available, it is recommended that you configure the library sources in IntelliJ IDEA. This will give you additional benefits such as the ability to view ASDocs when using library classes in your code, to navigate to the library source, and to debug the library code.

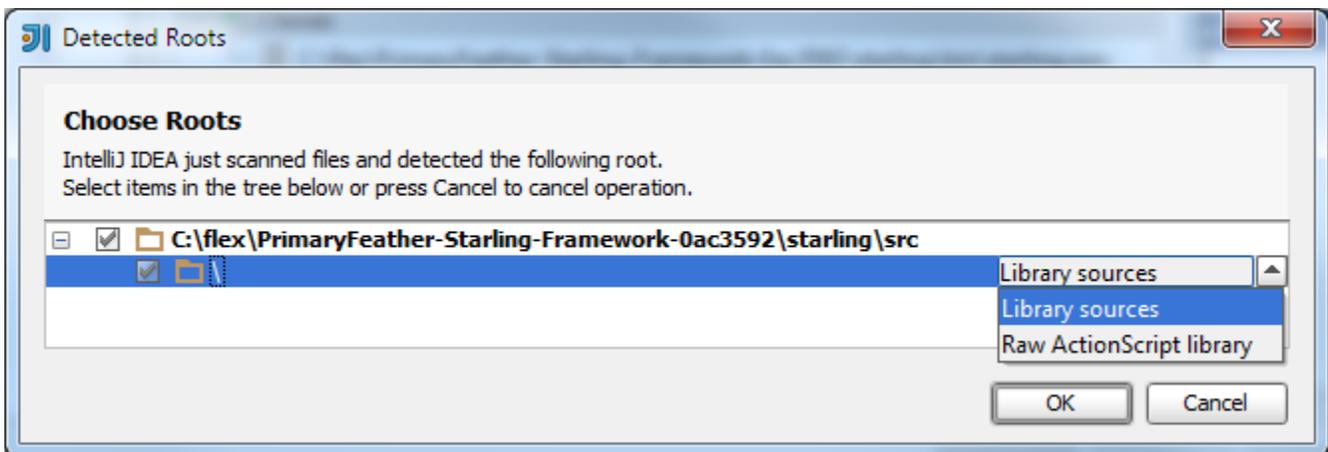
To add library sources, on the Dependencies tab, select the library that you have just configured and click the Edit icon.



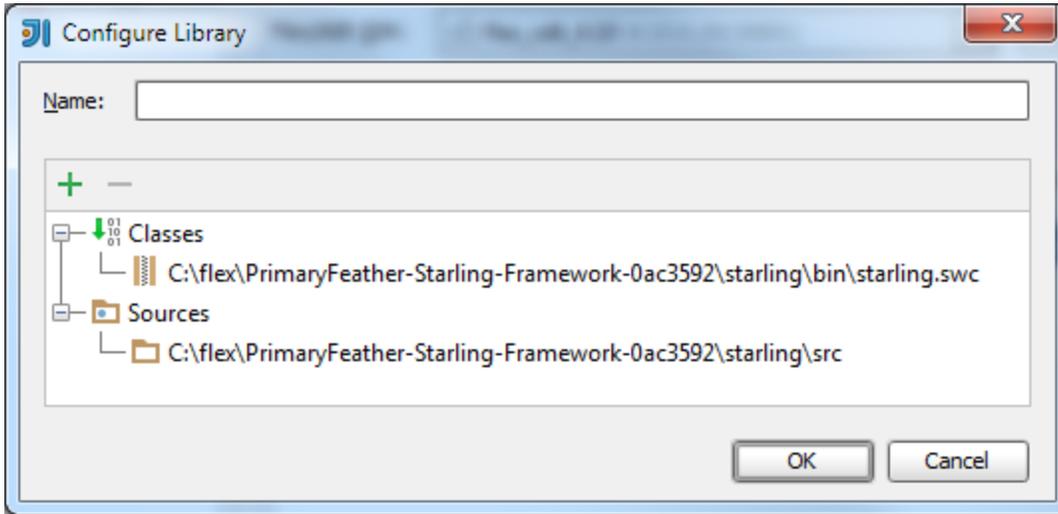
In the Configure Library dialog, click [+] and select Add Library Components. Select the location of the library sources in the file chooser.



Now, you have an option of specifying whether the found source root should be treated as a raw ActionScript library or as library sources. Select Library sources in the combo box and press OK.



As a result, the Sources category is added, and the location of the library sources is shown within it.

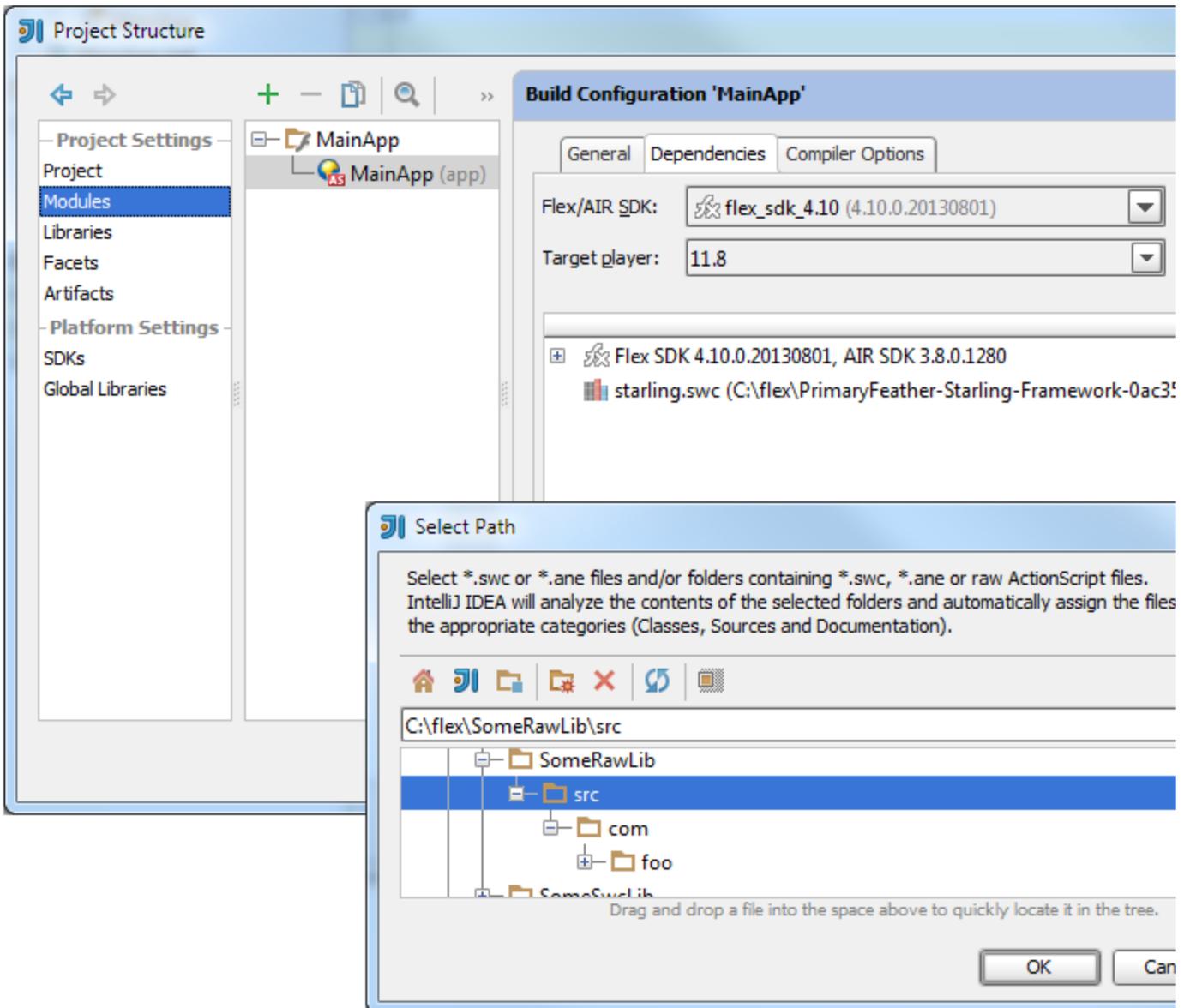


2. Configure a dependency on a raw ActionScript library

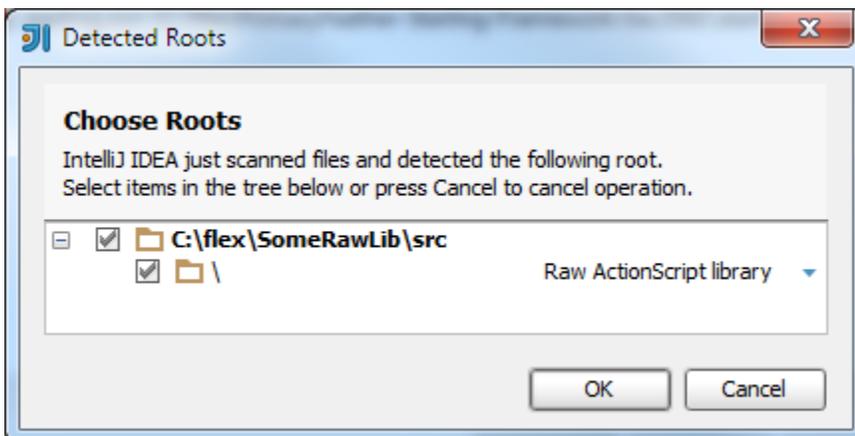
If an external library is not distributed as binary *.swc but is available as raw ActionScript sources, configure it as described in this section.

To open the Project Structure dialog, click the Project Structure icon on the main toolbar, or select File | Project Structure. Select Modules in the leftmost pane, then select your build configuration (shown as a subnode of the Flash module node) and click the Dependencies tab.

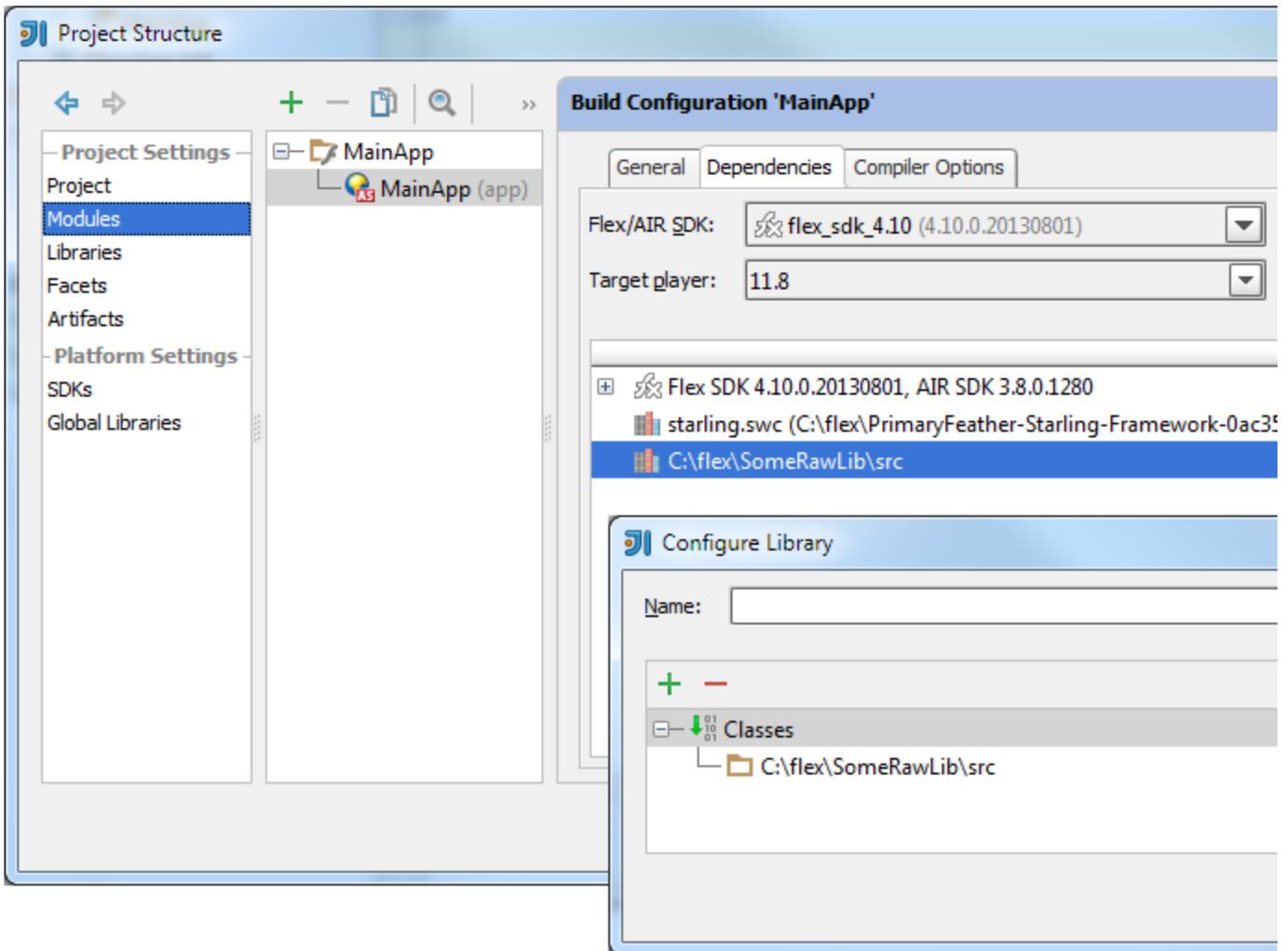
On the Dependencies tab, click [+] and select Add dependency on New Library. Then select the location of the raw AS library source folder in the file chooser.



In the Detected Roots dialog, make sure that Raw ActionScript library is selected and click OK.



To check the library configuration, select the library on the Dependencies tab and click the Edit icon.



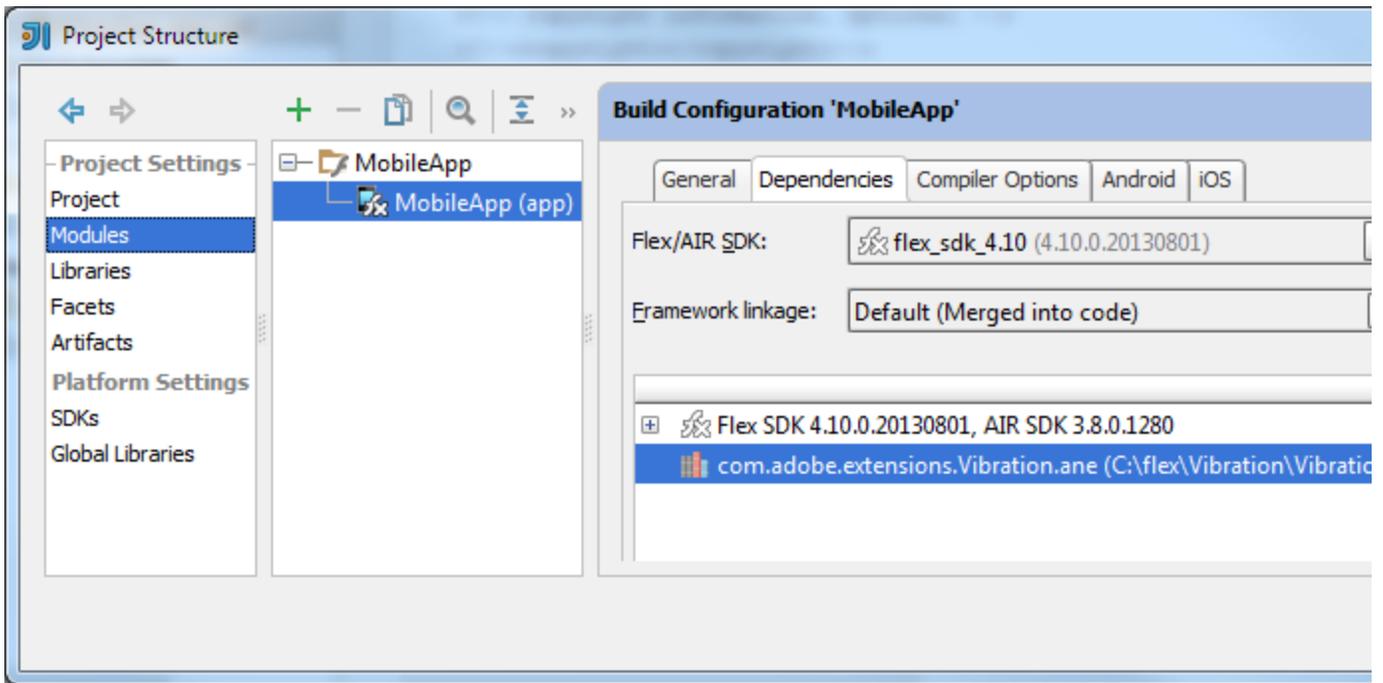
Notice that in the case of a raw AS library, the library source folder is in the Classes category. (In the example with the *.swc library, the *.swc file was under Classes, and the source folder was under Sources.

i IntelliJ IDEA protects library source code from being occasionally modified. If you navigate to the library source file, you'll see that it's read-only.

To be able to edit library code inside an IntelliJ IDEA project, you should configure the library as a separate Flash module with a build configuration producing a library, but not as an external library.

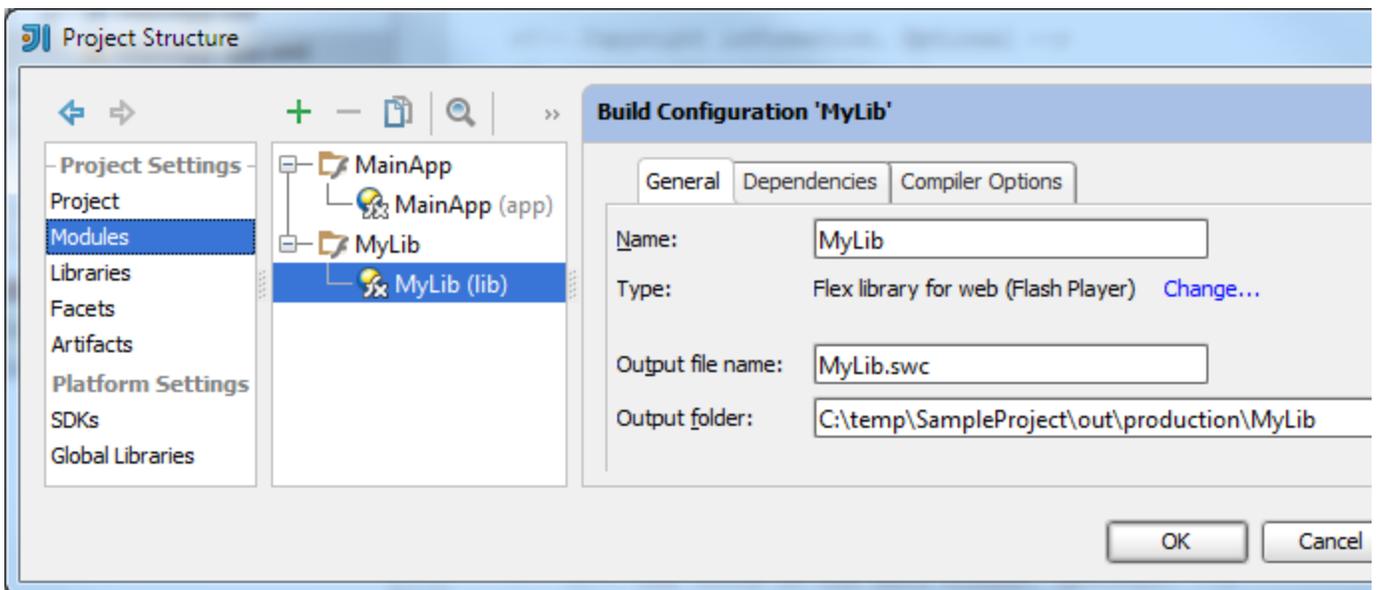
3. Configure a dependency on AIR Native Extensions (*.ane)

To develop a desktop- or mobile-targeted application that uses AIR Native Extensions (ANE) you need to add corresponding *.ane file(s) to dependencies of the corresponding build configuration. The procedure is very similar to that use for [adding a *.swc file as a library](#). The only difference is that a *.ane file should be selected in the file chooser rather than a *.swc file.



4. Configure a dependency on a library that belongs to a project

As a project grows bigger, developers usually try to split it into modules. A widely used approach is to keep common-purpose code in a separate module that has one build configuration of the Library type (i.e. one producing a *.swc).



Other build configurations may depend on this library.

To add the corresponding dependency, click [+] on the Dependencies tab of the corresponding build configuration and select Add dependency on Build Configuration. Then select your library build configuration in the Add Dependency dialog that opens.

