

Using the AspectJ (ajc) Compiler

By default, IntelliJ IDEA uses the `javac` compiler. So to use the AspectJ `ajc` compiler instead, you should change the corresponding IDE setting and specify the associated options.

Note that the `ajc` compiler is not bundled with the IDE and should be downloaded separately.

`ajc` is available as part of AspectJ Development Kit (AJDK) which you can download from the [AspectJ website](#).

- [Controlling the `ajc` aspectpath](#)
- [Specifying the AspectJ compiler settings](#)
- [Redefining the aspectpath at the module level](#)

Controlling the `ajc` aspectpath

IntelliJ IDEA lets you flexibly control the `aspectpath` command-line option for the [ajc compiler](#).

At the project level, the `aspectpath` may be defined in the [ajc compiler settings](#).

The project `aspectpath` may be [redefined](#) at the level of individual modules. This is done by configuring the AspectJ [facets](#) associated with the corresponding modules.

Depending on whether or not the `aspectpath` is specified at the project and module levels:

- The `aspectpath` is specified neither at the project nor at the module level: the `ajc` compiler uses the Java `classpath` of the project or module.
- The `aspectpath` is specified only at the project level: the `ajc` uses this `aspectpath` in all cases.
- The `aspectpath` is specified at the project level and redefined in certain modules (i.e. AspectJ facets): to compile the modules with the redefined `aspectpath`, `ajc` uses the corresponding module-specific `aspectpath`, and the project `aspectpath` in all other cases.

Note that the module `aspectpath` may be set automatically as a result of importing an appropriately configured [Maven](#) project into IntelliJ IDEA.

To specify the AspectJ compiler settings


1. Open the [Settings dialog](#).
2. In the left-hand part of the dialog, under **Project Settings**, expand the **Compiler** node and select **Java Compiler**.


3. On the [Compiler > Java Compiler page](#) that opens in the right-hand part of the dialog:

1. Select **Ajc** from the **Use compiler** list.
2. Specify the path to the compiler in the **Path to Ajc compiler** field.

You can type the path in the field, or click  and select the necessary file in the [corresponding dialog](#).

(The file that you want is `aspectjtools.jar` which is located in `<AJDK installation directory>\lib.`)

3. If necessary, specify the maximum heap size for the compiler in the **Maximum heap size (MB)** field. (The default is 128 MB.)
4. If necessary, specify the [command-line options](#)  to be passed to the compiler in the **Command line parameters** field.

You can type the parameters right in the field, or click  to open the **Command line parameters** dialog where the text entry area is larger.

Note that the `aspectpath` option may be specified separately, using the GUI provided below.

5. Click **Test** to check if the specified settings are correct.
6. If you want to specify the `aspectpath` option, select the **Explicit aspect path** check box, and then use the available controls to form the `aspectpath`:
 - Use **+** (Alt+Insert) to add project and global libraries to the list. Select the necessary library or libraries in the **Choose Libraries** dialog that opens.
 - Use **-** (Alt+Delete) to remove the selected items from the list.
 - Use **↑** (Alt+Up) to move the selected item one line up in the list.
 - Use **↓** (Alt+Down) to move the selected item one line down in the list.

Note that the `aspectpath` specified here may be redefined at the module level.

4. Click **OK** in the **Settings** dialog.

To redefine the aspectpath at the module level

To be able to redefine the `aspectpath` option at a module level, you should add an AspectJ facet to the corresponding module or modules. Once you have done that, you can control the `aspectpath` at the module level by configuring the corresponding AspectJ facet settings.

1. Add an AspectJ facet to the module of interest. For corresponding instructions, see [Adding and Removing Facets](#).
2. On the [Facet 'AspectJ' page](#) that opens in the right-hand part of the [Project Structure dialog](#), select the **Override project aspect path** check box. Use the available controls to form the `aspectpath` for the module:
 - Use **+** (Alt+Insert) to add project and global libraries, and/or the modules that the current module depends on. Select **Library** or **Module**, and then select the necessary libraries or modules in the **Choose Libraries** or the **Choose Modules** dialog.
 - Use **-** (Alt+Delete) to remove the selected items from the list.
 - Use **↑** (Alt+Up) to move the selected item one line up in the list.
 - Use **↓** (Alt+Down) to move the selected item one line down in the list.
3. Click **OK** in the **Project Structure** dialog.

See Also

Procedures:

- [Enabling AspectJ Support Plugins](#)
- [Creating a Library for aspectjrt.jar](#)
- [Compiling Applications](#)

Reference:

- [Compiler. Java Compiler](#)
- [AspectJ Facet Page](#)
- [Project Structure](#)

Web Resources:

- [Developer Community](#) 