

# Compiler and Builder

---

The Java builder of IntelliJ IDEA builds, or brings together source files, external libraries, properties files and configurations to produce a living application. The builder uses a compiler that works according to the Java specification.

The compiler treats encountered problems as errors and warnings. Unlike errors, the warnings do not prevent successful compilation. Both warnings and errors are reported in the Messages window.

The following key principles underlie the notion of compilation:

- All compilation tasks are performed in a separate process, fully independent from the main IDE's process, to get rid of the "out of memory" and other resources contention.
- Java compiler is used "in-process" via Java API for better performance and more flexible annotation processors management.
- Compilation is automatically triggered by events from the file system of the IDE, if [such option is enabled](#).

To build your project, you do not have to leave your editing environment: with IntelliJ IDEA you can compile, build and run your source code straight away.

Note that the way the module dependencies are ordered may be very important for the compilation to succeed. See [Processing Order](#).

In this part:

- [Supported Compilers](#)
- [Build Process](#)
- [Compilation Types](#)

## See Also

Procedures:

- [Compiling Applications](#)
- [Annotation Processors Support](#)

Reference:

- [Compiler](#)

Web Resources:

- [Developer Community](#) 