Processing Order

The module dependencies are processed in the order in which they appear in the corresponding list. In the user interface, this list is shown on the Dependencies tab of the Modules page in the Project Structure dialog.

At compile time, the order of items in the list defines the order in which the compiler (javac) looks for the classes to resolve the corresponding references. At runtime, this list defines the order in which the JVM searches for the classes to be loaded.

In certain situations, the order of libraries may be important. For example, if you want to use patches to certain libraries, these patches should be placed in the list before (above) the main libraries. Otherwise, new versions of the classes (contained in the patches) will be ignored.

For compilation, the position of an item in the list in relation to the module SDK (JDK) also matters, and in certain cases may be very important. (The module SDK is always shown in the list among the module dependencies.)

- All the items after the JDK are translated into the module `classpath` entries. Most of the libraries that you normally deal with should go there.
- All the items before the JDK correspond to the module `bootclasspath` entries.

  For the compilation to succeed, certain libraries (e.g. the ones having to do with the JDK itself and its extensions) must be included in the module `bootclasspath`. So, such libraries must appear in the list before the JDK.

  The use of such libraries, however, may be necessary in a very limited number of special cases. Thus, for the most of your modules, the JDK will (and should) be the first item in the list of dependencies.

See Also

Procedures:

- Configuring Module Dependencies and Libraries

Reference:

- Project Structure
- Dependencies Tab

Web Resources:

- Developer Community