

# Guide to Working with Multiple Modules

(If you're working with Maven 4, please refer to the Maven 4 edition of this guide ([./guide-multiple-modules-4.html](#)))

As seen in the introduction to the POM, Maven supports project aggregation in addition to project inheritance. This section outlines how Maven processes projects with multiple modules, and how you can work with them more effectively.

## The Reactor

The mechanism in Maven that handles multi-module projects is referred to as the *reactor*. This part of the Maven core does the following:

- Collects all the available modules to build
- Sorts the projects into the correct build order
- Builds the selected projects in order

## Reactor Sorting

Because modules within a multi-module build can depend on each other, it is important that the reactor sorts all the projects in a way that guarantees any project is built before it is required.

The following relationships are honoured when sorting projects:

- a project dependency on another module in the build
- a plugin declaration where the plugin is another module in the build
- a plugin dependency on another module in the build
- a build extension declaration on another module in the build
- the order declared in the `<modules>` element (if no other rule applies)

Note that only "instantiated" references are used - `dependencyManagement` and `pluginManagement` elements do not cause a change to the reactor sort order.

## Command Line Options

No special configuration is required to take advantage of the reactor, however it is possible to customize its behavior.

The following command line switches are available:

- `--resume-from` - resumes a reactor from the specified project (e.g. when it fails in the middle)
- `--also-make` - build the specified projects, and any of their dependencies in the reactor
- `--also-make-dependents` - build the specified projects, and any that depend on them
- `--fail-fast` - the default behavior - whenever a module build fails, stop the overall build immediately
- `--fail-at-end` - if a particular module build fails, continue the rest of the reactor and report all failed modules at the end instead
- `--non-recursive` - do not use a reactor build, even if the current project declares modules and just build the project in the current directory

Refer to the Maven command line interface reference for more information on these switches.

# More information

- Chapter 6. A Multi-module Project (Maven by Example) (<http://books.sonatype.com/mvnex-book/reference/multimodule.html>)