

# Make Like Reactor Mode

## "Make" like build behavior mode

Maven currently has a top down build approach where you start at the top of a reactor and build all children. One of the most common requests I get from my developers is an easy way to build certain artifacts from the bottom up. Often times a build, especially large ones, will contain many modules needed for a "full" build, but are actually made up of pockets of interdependencies. It's not always possible to logically group these all together with a common parent to enable easily building a subtree.

For example, you may have a project comprised of services, ui's and packages:

```
+---packages
|   +---a-package
|   +---b-package
|   \---c-package
+---services
|   +---a-service
|   +---b-service
|   \---c-service
\---ui
    +---a-ui
    +---b-ui
    \---c-ui
```

The packages inherit from the package parent, etc. Assume that "A-package" depends on "a-service" "b-service" and "a-ui"

In Maven, there is currently no easy way to make a change to "a-service" and build it and the package at once. This can be controlled to some extent with modules in profiles, but this quickly becomes unwieldy and doesn't support the granularity needed.

### Out of Scope

It is out of scope for this proposal to determine if a project actually needs to be rebuilt based on the contents. (ie checking if anything has actually changed) This is simply intended to be an extension to the reactor behavior in choosing which projects should be included in the reactor.

### Proposed Solution 1 (NOT IMPLEMENTED)

The ideal use case for this issue is:

1. Developer makes change to code in "a-service"
2. Developer goes to "a-package" and executes "mvn -A install" (-A for all)
3. Maven walks up the parent tree to see how far up the source goes. Then any dependencies in the graph that are found in the contained source on disk are ordered and built. Everything else is ignored in the build.

### **Alternate Use Case 2:**

2. Instead of going to "a-package" and executing mvn, the developer goes to the top level parent and executes "mvn -Apackages/a-package" (in this case defining the module that should be considered for dependency graphing)
3. Maven builds the graph and builds what is needed.

This use case isn't ideal but is probably easier to implement since the top level parent doesn't need to be located and everything to be built is included in the subtree.

### **Porting maven-reactor-plugin**

The [maven-reactor-plugin](#) can be used today to build stuff. We'd like to port at least reactor:resume, reactor:make, reactor:makeDependents into core. I'm thinking it would go something like this:

New options:

- -rf --resume-from: Resume reactor from specified project
- -pl --project-list: Build the specified reactor projects instead of all projects
- -am --also-make: If a project list is specified, also make projects that the list depends on
- -amd --also-make-dependents: If a project list is specified, also make projects that depend on projects on the list

Examples:

- resume: mvn -rf packages/b-package
- make: mvn -am -pl ui/a-ui,ui/b-ui
- make dependents: mvn -amd -pl packages/a-package,packages/b-package
- make both: mvn -am -amd -pl packages/b-package

At least at first, core would not include the new reactor:makeScmChanges because that would require maven-core to depend on maven-scm.