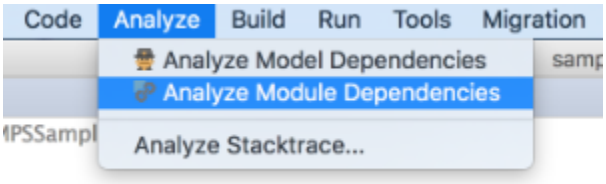
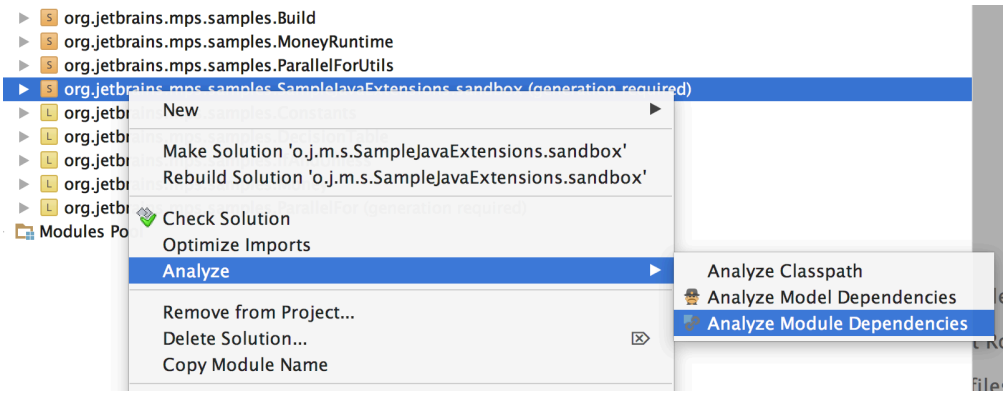
Module Dependencies Tool

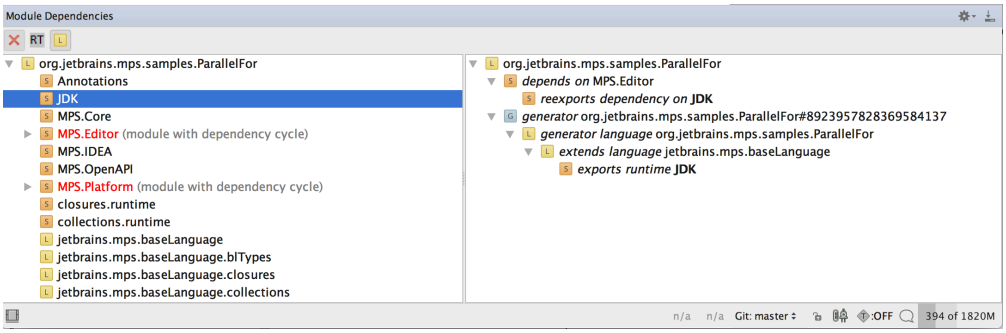
The Module Dependencies Tool allows the user to overview all the dependencies and used languages of a module or a set of modules, to detect potential cyclic dependencies as well as to see detailed paths that form the dependencies. The tool can be invoked from the menu as well as from the project pane when one or more modules are selected.



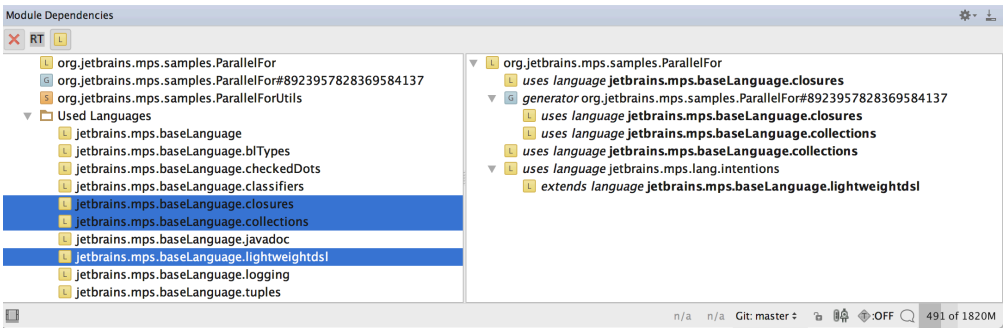


Module Dependency Tool shows all transitive dependencies of the modules in the left panel. Optionally, it can also display all directly or indirectly used languages. It is possible to expand any dependency node and get all dependencies of the expanded node as children. These will again be transitive dependencies, but this time for the expanded node.

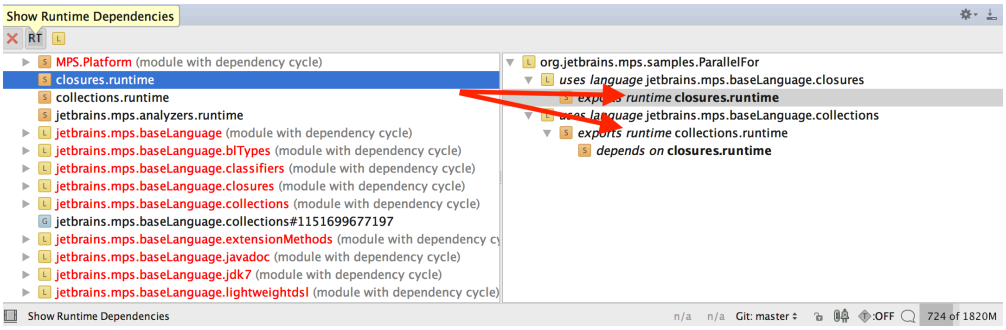
Select one or more of the dependency nodes in the left panel. The right panel will show paths to each of the selected modules from its "parent" module. You can see a brief explanation of each relation between modules in the right tree. The types of dependencies can be one of: depends on, uses language, exports runtime, uses devkit, etc. For convinience the name of the target dependent module is shown in bold.



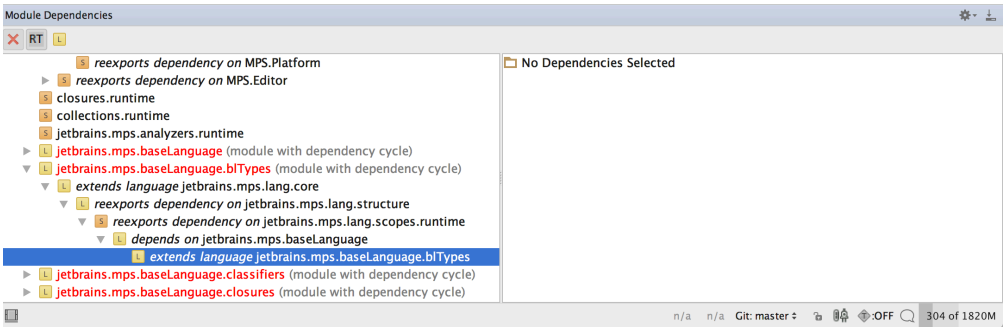
There are two types of dependency paths: Dependency and Used Language. The L button in the toolbar enables/disables displaying of Used Languages in the left tree panel. When you select a module in the Used Language folder in the left tree, the right tree shows only the dependency paths that introduce the used language relation for the given module. To show "ordinary" dependencies on a language module, you should select it outside of the Used Languages folder (e.g. the jetbrains.mps.lang.core language in the picture below). It is also possible to select multiple nodes (e.g. the same language dependency both inside and outside of the Used Language folder). In that case you get a union of results for both paths.



When you are using a language that comes with its own libraries, those libraries are typically not needed to compile your project. It is the runtime when the libraries must be around for your code to work. For tracking runtime dependencies in addition to the "compile-time visible" ones, you should check the Runtime option in the toolbar. The runtime dependencies are marked with a "runtime" comment.



The modules in the left tree that participate in dependency cycles are shown in red color. It is possible to expand the tree node to see the paths forming the cycle:



For some types of dependencies the pop-up menu offers the possibility to invoke convenience actions such as Show Usages or Safe Delete. For the "depends on" dependencies (those without re-export) Dependencies Analyzer will be invoked for the Show Usages action.