

# Compiler

You can mix your groovy and java code and didn't think about combined (possible, cyclic) compilation.  
Click [Compile Module](#) (or [Make Module](#)) will compile (make) whole module.

The image shows an IDE window with a project structure on the left, a context menu in the center, and Groovy code on the right.

**Project Structure (Left):**

- module
  - src (C:\Documents and Settings\Dmitry.Krasn...)
    - src (C:\groovy\groovy-1.0\src)
    - bin
    - conf
    - examples
      - commandLineTools
        - AntMap.groovy
        - BigTests.groovy
        - ListFiles.groovy
        - Reflections.groovy
        - SimpleWebServer.groovy
      - console
      - groovy2d
      - searchEngine
      - swing
      - webapps
    - latex
    - main
      - groovy
        - inspect
          - swingui
            - TableMap
            - TableSorter
            - ObjectBrowser.groovy
          - Inspector
        - lang
        - mock
          - interceptor
            - MockProxyMetaClass
            - Demand.groovy
            - LooseExpectation.groovy
            - MockFor.groovy
            - MockInterceptor.groovy
            - StrictExpectation.groovy
            - StubFor.groovy
            - package.html
          - ClosureConstraintMatcher
          - GroovyMock
          - package.html

**Context Menu (Center):**

- New
- Jump to Source F4
- Cut Ctrl+X
- Copy Ctrl+C
- Copy Paths Ctrl+Shift+C
- Copy Reference Ctrl+Alt+Shift+C
- Paste Ctrl+V
- Paste... Ctrl+Shift+V
- Paste Simple Ctrl+Alt+Shift+V
- Find Usages... Alt+F7
- Find in Path... Ctrl+Shift+F
- Replace in Path... Ctrl+Shift+R
- Analyze
- Refactor
- Add To Favorites
- Move to Changelist...
- Reformat Code... Ctrl+Alt+L
- Optimize Imports... Ctrl+Alt+O
- Delete Delete
- Toggle Method Breakpoint
- Make Module 'module'
- Compile Module 'module' Ctrl+Shift+F9**
- Local History
- Synchronize selected files
- Compare with Editor
- Module Settings...
- Edit Scopes

**Code (Right):**

```

def classLabel = '<
label(classLabel)
}

dPane(constraints:
f (inspector.objec
scrollPane(name
itemTable =
int i =
def dat
tableMo
clo
clo
}
}
}
f (inspector.objec
scrollPane(name
itemTable =
int i =
def dat
tableMo
clo
clo
clo
}
}
}
scrollPane(name: '
fieldTable = ta
def data =
data.addAll
tableModel(
closure
closure
closure
closure
closure
closure
}
}
}
scrollPane(name: ' (
methodTable = t
def data =
data.addAll

```