

# SubjectJ BNF Grammar

The following is the BNF grammar for the .sj file that SubjectJ will use to perform the compositions. Note that the project is in the *very* pre-alpha stages, and the grammar are not just subject to, but very likely to change.

The file will contain three parts (one of which is optional):

$\langle SubjectJ \rangle ::= \langle Include \rangle [\langle Concern \rangle] \langle Subject \rangle$

**include:** This part contains the list of all packages and/or class files that SubjectJ will work with. Any package or class file not listed in this part will not be modified or otherwise touched by SubjectJ. This part is important because even if some of the included classes may not be mentioned in any mappings, SubjectJ will update every class named in this list as necessary so that they correctly interact with any newly composed class(es).

$\langle Include \rangle ::= \text{include: } \langle Module \rangle +$

$\langle Module \rangle ::= \langle File \rangle$   
                  |  $\langle Directory \rangle$

$\langle File \rangle ::= \langle Name \rangle$

$\langle Directory \rangle ::= \langle Name \rangle .* [\text{recursive}] [\langle Exclude \rangle]$

$\langle Exclude \rangle ::= \text{exclude: } \langle Module \rangle +$

$\langle Name \rangle ::= \langle Identifier \rangle ( . \langle Identifier \rangle )^*$

$\langle Identifier \rangle ::= (a | .. | z | A | .. | Z | 1 | .. | 9) +$

**concern: (optional)** This part allows the user to group individual packages or class files into ‘concerns’ to make referring to them easier when defining subjects. In other words, this is the concern mapping part. This part is optional, however; SubjectJ, like Hyper/J, creates a mapping for every included class file (see include: above) – the name of the concern will be the Java fully-qualified name. This name can be used in the subjects (see below) instead if desired, skipping any need for this part.

$\langle Concerns \rangle ::= \langle Concern \rangle \langle Concerns \rangle$   
                  |  $\langle Concern \rangle$

$\langle Concern \rangle ::= \text{package } \langle Key Value \rangle$   
                  |  $\text{class } \langle Key Value \rangle$   
                  |  $\text{interface } \langle Key Value \rangle$   
                  |  $\text{method } \langle Key Value \rangle$   
                  |  $\text{field } \langle Key Value \rangle$

$\langle Key Value \rangle ::= \langle Path \rangle : \langle String \rangle$

$\langle Path \rangle ::= \langle String \rangle . \langle Path \rangle$   
                  |  $\langle String \rangle$

$\langle String \rangle ::= c \langle String \rangle$   
                  |  $c$

**The subject (.sj) file** This is where the action happens; the actual subjects are defined in this part. This grammar is the most likely to change based on requirements and feedback as development continues. As it stands currently, there can be one or more subjects – one for each subjective view of the code. The subject names must be distinct – unlike Hyper/J, the names are not specified as commandline arguments; the names from this section will be used as the names of the folders in which the composed files will be placed. As noted above, the concerns in this part may be the concerns named in the concerns part (above), or they may simply be Java fully-qualified names. SubjectJ will only perform compositions on the named concerns in this part; all other concerns will be untouched with the exception of being updated to work correctly with the newly composed files.

$\langle Subject \rangle ::= \langle Name \rangle \langle Concerns \rangle \langle GeneralStrategy \rangle \langle SpecificStrategies \rangle$

$\langle Name \rangle ::= \langle String \rangle$

$\langle Concerns \rangle ::= \langle Concern \rangle \langle Concerns \rangle$   
 $\quad \quad \quad | \quad \langle Concern \rangle$

$\langle Concern \rangle ::= \langle Path \rangle$   
 $\quad \quad \quad | \quad \langle String \rangle$

$\langle GeneralStrategy \rangle ::= \text{merge by name}$   
 $\quad \quad \quad | \quad \text{noncorresponding merge}$   
 $\quad \quad \quad | \quad \text{override by name}$

$\langle SpecificStrategies \rangle ::= \langle Strategy \rangle \langle SpecificStrategies \rangle$   
 $\quad \quad \quad | \quad \langle Strategy \rangle$

$\langle Strategy \rangle ::= \langle Merge \rangle$   
 $\quad \quad \quad | \quad \langle Override \rangle$   
 $\quad \quad \quad | \quad \dots$   
 $\quad \quad \quad | \quad \epsilon$

$\langle Merge \rangle ::= \text{merge } \langle Method \rangle \text{ with } \langle Method \rangle$   
 $\quad \quad \quad | \quad \text{merge } \langle Method \rangle \text{ with } \langle Method \rangle \text{ via } \langle Method \rangle$   
 $\quad \quad \quad | \quad \text{merge } \langle Field \rangle \text{ with } \langle Field \rangle \text{ via } \langle Method \rangle$

$\langle Override \rangle ::= \text{override } \langle Method \rangle \text{ with } \langle Method \rangle$   
 $\quad \quad \quad | \quad \text{override } \langle Field \rangle \text{ with } \langle Field \rangle$

$\langle Class \rangle ::= \langle Concern \rangle . \langle String \rangle$   
 $\quad \quad \quad | \quad \langle Concern \rangle$

$\langle Method \rangle ::= \langle Concern \rangle . \langle String \rangle$   
 $\quad \quad \quad | \quad \langle Concern \rangle$

$\langle Field \rangle ::= \langle Concern \rangle . \langle String \rangle$   
 $\quad \quad \quad | \quad \langle Concern \rangle$

$\langle Path \rangle ::= \langle String \rangle . \langle Path \rangle$   
 $\quad \quad \quad | \quad \langle String \rangle$

$\langle String \rangle ::= c \langle String \rangle$   
 $\quad \quad \quad | \quad c$