abc command-line options

April 2, 2008

Contents

1 General Options

`-h` , `-help` Prints the usage screen for abc.

`-v` , `-version` Print the abc version number. The underlying versions of soot and polyglot, which form part of abc, are also printed.

`-verbose` Display information about what abc is doing as it runs.

`@`⟨*filename*⟩, `-argfile` ⟨*filename*⟩ Read a list of arguments from the file *filename* and behave as if they had been passed directly on the command-line.

2 Input Options

`-sourceroots` ⟨*path*⟩ Compile all .java files found in the directories given by *path* and any of their subdirectories.

`-injars` ⟨*jar list*⟩ Use all the class files in the jar files specified by *jar list* as source.

`-inpath` ⟨*dir list*⟩ Use all the class files in the directories specified by *dir list* as source.

`-cp` ⟨*classpath*⟩, `-classpath` ⟨*classpath*⟩ Specify a list of zips, jars and directories that will be used when searching for libraries referred to in the code being compiled. The default value is the classpath abc is invoked with.

`-main-class` ⟨*class*⟩ When performing interprocedural optimizations (with -O3), abc needs to know the main class (entry point) of the program. This option sets the main class.

3 Output Options

`-dava` (default value: `false`) After weaving, run the Dava decompiler to produce Java source files of the woven code rather than outputting class files.

`-outjar` ⟨*jar*⟩ Write output class files into the jar file specified by *jar*.

`-d` ⟨*path*⟩ Write output class files into the directory specified by *path*.

`-tag-instructions` (default value: `false`) Tag overhead instrutions with metadata required for calculating dynamic metrics.

`-g` (default value: `false`) Emit bytecode with debug information [currently unsupported].

4 Warning/Error Reporting Options

`-warn-unused-advice` (default value: `true`) If a piece of advice does not apply at any join point shadow, generate a warning.

`-warn-prec-ambiguity` (default value: `false`) If multiple pieces of advice apply at the same shadow, and the language specification does not dictate a precedence order, generate a warning.

5 Language Options

`-nested-comments` If this option is enabled, comments of the form /* ... /* ... */ ... */ will be allowed.

`-ext` ⟨*package name*⟩ (default value: `abc.main`) Load the AspectJ language extension defined in the package given by *package name*. For example, abc.eaj specifies the EAJ language extension supplied with abc.

`-source` ⟨*release*⟩ (default value: `1.4`) Provide source compatibility with specified release given by *release*. The default is 1.4. Use -source 1.3 if you need 'assert' as an identifier, and -source 1.5 if you intend to use 'enum' as a keyword.

`-abc101runtime` Use the abc 1.0.0/1.0.1 runtime. This disables some cflow optimisations present in versions 1.0.2 onwards.

6 Optimization Options

`-O`*arg* (default value: `1`) Set the general optimization level. 0 means no optimizations, and 1 means the standard intra-procedural options.

`-w` (default value: `false`) Allow abc to analyze your program and the Java runtime library in whole-program mode.

`-around-force-closures` (default value: `false`) Force closures for around advice.

`-around-inlining` (default value: `true`) Enable inlining of around advice.

`-around-force-inlining` (default value: `false`) Inline around advice whenever possible (as opposed to adaptive inlining).

`-before-after-inlining` (default value: `true`) Enable inlining of before and after advice.

`-before-after-force-inlining` (default value: `false`) Inline before and after advice whenever possible (as opposed to adaptive inlining).

`-cflow-use-counters` (default value: `true`) Implement the cflow construct with counters rather than stacks whenever possible for performance.

`-cflow-use-sharing` (default value: `true`) Use only one stack or counter for several occurrences of the same cflow pointcut.

`-cflow-share-thread-locals` (default value: `true`) Retrieve the thread-local instance of each cflow stack only once per method.

7 Static whole-program analysis for dependent advice (abc.da) and tracematches (abc.tmwpopt)

`-laststage` ⟨*stage*⟩ (default value: `quick`) Specifies the last stage to be executed in the static whole program analysis for dependent advice and tracematches. There are three stages: (1) quick, (2) flowins (flow-insensitive analysis) and (3) flowsens (flow-sensitive analysis, tracematches only). They are executed in this order. Default is quick.

`-warn-about-individual-shadows` (default value: `false`)

If enabled, abc issues a warning for each individual shadow removed by the analyses. If disabled, only summary information about disabled shadows and tracematches will be given. Default is false.