

v8::internal::RegExpQuantifier  
::RegExpQuantifier

v8::internal::anonymous  
\_namespace{experimental  
\_compiler::cc}::CanBeHandledVisitor  
::VisitQuantifier

v8::internal::anonymous  
\_namespace{experimental  
\_compiler::cc}::CompileVisitor  
::VisitQuantifier

v8::internal::RegExpQuantifier::min

```
graph LR; A["v8::internal::RegExpQuantifier::RegExpQuantifier"] --> D["v8::internal::RegExpQuantifier::min"]; B["v8::internal::anonymous_namespace{experimental_compiler::cc}::CanBeHandledVisitor::VisitQuantifier"] --> D; C["v8::internal::anonymous_namespace{experimental_compiler::cc}::CompileVisitor::VisitQuantifier"] --> D;
```

The diagram illustrates a dependency or inheritance structure. Three source nodes on the left point to a single target node on the right. The first source node is 'v8::internal::RegExpQuantifier::RegExpQuantifier'. The second source node is 'v8::internal::anonymous\_namespace{experimental\_compiler::cc}::CanBeHandledVisitor::VisitQuantifier'. The third source node is 'v8::internal::anonymous\_namespace{experimental\_compiler::cc}::CompileVisitor::VisitQuantifier'. The target node is 'v8::internal::RegExpQuantifier::min'. All arrows are blue and point towards the target node.