

v8::internal::MacroAssembler
::LoadZeroIfConditionNotZero

v8::internal::MacroAssembler
::LoadZeroIfConditionZero

v8::internal::Assembler::neg

```
graph LR; A["v8::internal::MacroAssembler::LoadZeroIfConditionNotZero"] --> C["v8::internal::Assembler::neg"]; B["v8::internal::MacroAssembler::LoadZeroIfConditionZero"] --> C;
```

The diagram illustrates a code reuse pattern. Two methods from the `v8::internal::MacroAssembler` namespace, `LoadZeroIfConditionNotZero` and `LoadZeroIfConditionZero`, are shown on the left. Both methods have arrows pointing to a single method in the `v8::internal::Assembler` namespace, `neg`, which is highlighted in a gray box on the right. This indicates that both macro assembler methods delegate the actual negation operation to the `neg` method of the base assembler class.