C++ static code analysis: 'extern "C"' should not be used with namespaces

1 minute

The C linkage declaration `extern "C"` can not be combined with a namespace. In practical terms only one function with that name can be declared as `extern "C"` because the `namespace` is functionally ignored.

Noncompliant Code Example

```
namespace ns1 {
    extern "C" void doSomething();  // Noncompliant
    // ...
}

extern "C" {
  namespace ns2 {  // Noncompliant
    // ...
  }
  // ...
}

ns1::doSomething();
doSomething(); // Works too, same as above
```

Compliant Solution

```
extern "C" void doSomething();

namespace ns1 {
  // ...
}

extern "C" {
  // ...
}

namespace ns2 {  // Noncompliant
  // ...
}

doSomething();
```

Exceptions

`extern "C"` can prefix `typedef`.

```
namespace ns1 {
    extern "C" typedef void c_function();  // Compliant, type named
'c_function' exists only in 'ns1' and not in the global namespace
    // ...
}
```