Allowing the establishment of namespace scopes in an export-declaration

1 Introduction

This paper proposes to allow the establishment of namespace scopes in an export-declaration and define a syntax for doing so.

2 Motivation and Scope

While headers allow one to relatively easily use namespaces without wrapping files in namespace blocks, modules have no such equivalent. Using headers, one can wrapping the header files in namespace blocks and prepending the namespace onto the functions in the implementation file, although this is against good style.

Through this change, this paper creates a simple, compact way to use namespaces with modules, as opposed to wrapping implementation files inside namespace blocks.

(There is not currently a reference implementation, although one would be trivial to create, as a non-standard extension to the core language is a tough sell for compilers.)

3 Wording

Relative to N4910.

3.1 #[module.interface]

Change the grammar before paragraph 1:

```
  export-declaration:
+    export module-name namespace-name
     export declaration
     export { declaration-seqopt}
     export-keyword module-import-declaration
```

Change paragraph 1:

```
An export-declaration shall inhabit a namespace scope
and appear in the purview of a module interface unit.
An export-declaration shall not appear directly or indirectly
within an unnamed namespace or a private-module-fragment.
+ An export-declaration containing a module-name and a namespace-name establishes a
+ named module according to the module-name and establishes and
+ exports a namespace-scope according to the namespace-name.
```

```
- An export-declaration has the declarative effects of its declaration,
+ Any other export-declaration has the declarative effects of its declaration,
declaration-seq (if any), or module-import-declaration.
The declaration or declaration-seq of an export-declaration shall not
contain an export-declaration or module-import-declaration.

- [Note 1: An export-declaration does not establish a scope. - end note]
```