Subject: Extensible modules
From: Van Snyder
Reference: 98-105

1 Number

TBD

2 Title

Extensible modules.

3 Submitted By

J3

4 Status

For consideration.

5 Basic Functionality

Provide for modules to be extended, in a way analogous to type extension. An extension modules would access its parent by host association, and be accessible by use association. Private components and bindings of extensible types do not — cannot — become public in an extension type. Similarly, private entities of an extensible module would not — could not — become public entities of an extension module.

6 Rationale

If one needs to extend a module, and extending it requires access to private entities of the module, one needs to modify the module. The probability of encountering a need for this is increased by the existence of extensible types.

In addition to all of the undesirable results laid out in the Modules Technical Report 19767, modifying a module is impossible if one doesn't have the source text for it.

7 Estimated Impact

This is a modest project, of substantially lesser scope than submodules, both from the point of view of the standard and for implementors. The most difficult problem for implementors will be to provide for an extension module to access its parent by host association, but that problem needs to be solved anyway to implement the Modules TR.

8 Detailed Specification

Provide for specification that a module either is or is not extensible.

We may ultimately decide that all modules are extensible, and that these specifications are therefore not necessary.

Provide for specification that a module is an extension of another module, and that an extension is by default extensible.

Examples:

```
  module, extensible :: MyModule

  module, nonextensible :: MyModule
```

```

module, extends(myModule) :: MyExtension

module, extends(myModule), nonextensible :: MyDeadEnd
```

Unlike type extension, provide that an extension module accesses its parent by host association.

There is a detailed example in 98-105.

8.1 Edits w.r.t. 03-007r2

A few more edits than these, but not many more, may be necessary. The primary purpose for providing these is to give an idea of the scope of the project.

R1105 *module-stmt* **is** MODULE [[, *module-attrib-list*] ::] *module-name* 250:11
R1105a *module-attrib* **is** EXTENSIBLE
or NONEXTENSIBLE
or EXTENDS (*module-name*)

C1103a (R1105a) If EXTENSIBLE appears, NONEXTENSIBLE shall not appear.

C1103b (R1105a) The *module-name* shall be the name of an extensible module.

A module that is introduced by a MODULE statement in which the NONEXTENSIBLE attribute does not appear is an **extensible module**. A module that is introduced by a MODULE statement in which the EXTENDS attribute appears is an **extension module**. The module named in the EXTENDS attribute is the **parent** of the extension module. An extension module accesses is parent module by host association. 251:4+

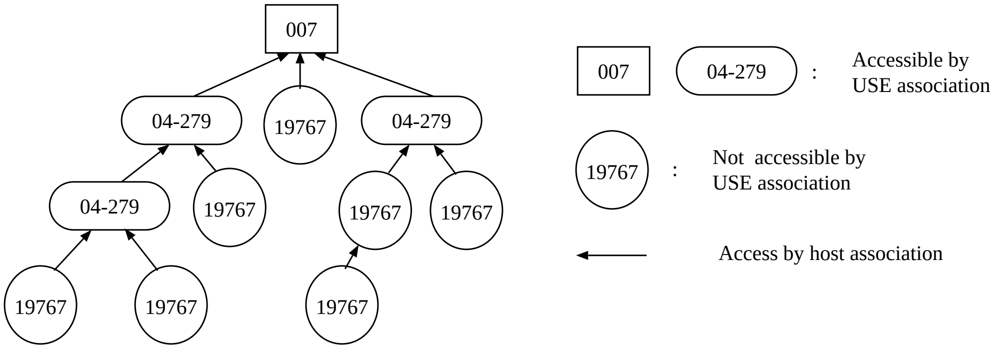
If we keep the glossary, definitions for "extension module" and "extensible module" will be needed, and the definition of "parent" given in the Modules TR will need embellishment.

An example in Annex C may be desirable. One can be plagiarized from 98-105.

In the several places in the Modules TR where it says "the ancestor module," the wording will need to be changed to "an ancestor module".

9 History

10 Illustration of relation of 007, 19767 and 04-279


007
04-279
19767
04-279
04-279
19767
19767
19767
19767
19767
19767
007
04-279
:
Accessible by
USE association
19767
:
Not accessible by
USE association
Access by host association