15. Separate compilation

If an implementation permits user-defined procedure and class declarations to be separately compiled, then a program should have means of making reference to such declarations as external to the program.

The following additional declarations are recommended as an optional part of the Common Base.

15.1 Syntax

```
<external item> ::= <external identifier>|
                    <identifier> = <external identifier>
<external list> ::= <external item>|
                    <external list>,<external item>
<external declaration> ::=
             external procedure <external list>|
             external <type> procedure <external list>|
             external  class <external list>
```

15.2 Semantics

An external identifier is an identification with respect to an "operating system" of a separately compiled declaration.

An external item introduces a local identifier for such a declaration. The local identifier may or may not be identical to the corresponding external identifier.

An external declaration represents a copy of each of the separately compiled procedures or class declarations identified by its external list. Each copy is modified by replacing occurrences of the original procedure or class identifier by occurrences of the given local identifier.