

# ISO/IEC JTC1/SC22/WG5 N1205

To: WG5  
From: Malcolm Cohen

Date: Thu, 18 Jul 96 15:51:11 +0100  
Subject: (SC22WG5.1133) Problems with proposed change to 1539-2

There are a couple of problems with the proposed revision of the example module for ISO\_VARYING\_STRING, in paper N1183.

First off, a minor problem:

```
Given
  USE ISO_VARYING_STRING
  TYPE(VARYING_STRING) A
```

Although it is legal (by the normative text) to say:  
A = A

It causes a problem with the new module, because it deallocates the storage before using it.

Secondly, an extremely serious problem:

```
Given
  USE ISO_VARYING_STRING
  TYPE MYTYPE
    TYPE(VARYING_STRING) VALUE
  END TYPE
  TYPE(MYTYPE) A,B
```

The assignment  
A = B

will invoke \*\*\*POINTER ASSIGNMENT\*\*\* on the components, not the one defined by the module.

This is ok with the old string module in 1539-2, because it never deallocates or overwrites the hidden pointer ("chars") arrays.

But with the proposed module in N1183, after the above assignment, the statement

```
B%VALUE = "Goodbye"
```

will deallocate the storage pointed to by the hidden pointer component of A%VALUE, effectively destroying the VARYING\_STRING component of A.

There is certainly no obvious way of getting around this problem, so the N1183 module is not a conforming implementation of ISO\_VARYING\_STRING.

Thus IMO the appropriate action to be taken is not to revise the string module for Fortran 95, but to wait until sufficient facilities have been added to the language to make a more efficient solution than the one we already have possible. Some possible solutions would be allocatable components with nice assignment, a special "defined\_assignment" attribute for derived types to make them use the defined assignment even when they are components, or some clever new OO stuff.