To: **WG5**
From: **Lawrie Schonfelder**
Date: **27-Jul-1997**
Subject: **Fortran 95 upgrade to the Varying String Standard and Module**

Now that Fortran 95 has been put to bed, a serious look at what this makes possible in improvements to the part 2, Varying String standard, has been initiated. In fact, a number of very significant improvements are possible. The most significant for the user is that most of the functions defined by the standard can either be made PURE or ELEMENTAL. The former allows them to be used within specification expressions and hence allows a number of useful programming methods to be employed. The latter allows automatic extension to apply to arrays in ways entirely analogous to those permitted for similar intrinsic functions. Both of these greatly enhance the completeness of the data-abstraction defined by the standard. The other improvement involves no change to the standard itself but makes use of the type-initialisation possibilities to improve the module implementation in a way that reduces the memory leakage. Below is a summary of the basic changes proposed. The summary is based on the contents page of the HTML version of the current standard. Under each heading it gives an outline of the nature of the proposed changes. The full text of the proposed standard, in HTML can be found at the attached web page.

---

1. Introduction
   Editorial Changes only, updating text and references to refer to Fortran 95
2. General
    1. Scope
       Editorial Changes only, updating text and references to refer to Fortran 95
    2. Normative References
       Editorial Changes only, updating text and references to refer to Fortran 95
3. Requirements
    1. The Name of the Module
       No change
    2. The Type
       No change
    3. Extended Meanings for Intrinsic Operators
        1. Assignment
           Extended to describe ELEMENTAL semantics
        2. Concatenation
           Extended to describe ELEMENTAL semantics
        3. Comparisons
           Extended to describe ELEMENTAL semantics
    4. Extended Meanings for Generic Intrinsic Procedures
        1. The LEN procedure
           Extended to describe PURE semantics. Note, it would be possible to make this version of LEN ELEMENTAL since when applied to an array of strings it would produce an array of lengths. This is not being proposed since for CHARACTER arrays the LEN function produces a scalar length.
        2. The CHAR procedure
           Extended to describe PURE semantics. Note, these versions of CHAR are not made ELEMENTAL since in general an array of strings will have an array of different lengths and such an object does not have a sensible CHARACTER form.
        3. The ICHAR procedure
           Extended to describe ELEMENTAL semantics
        4. The IACHAR procedure
           Extended to describe ELEMENTAL semantics
        5. The TRIM procedure
           Extended to describe PURE semantics
        6. The LEN_TRIM procedure

Extended to describe ELEMENTAL semantics

7. The ADJUSTL procedure
   Extended to describe ELEMENTAL semantics
8. The ADJUSTR procedure
   Extended to describe ELEMENTAL semantics
9. The REPEAT procedure
   Changed to avoid explicit error exit, negative ncopies produces zero length string result in common with a number of existing intrinsic function implementations
10. Comparison procedures
    Extended to describe ELEMENTAL semantics
11. The INDEX procedure
    Extended to describe ELEMENTAL semantics
12. The SCAN procedure
    Extended to describe ELEMENTAL semantics
13. The VERIFY procedure

5. Additional Generic Procedure for Type Conversion
   1. The VAR_STR procedure
      Extended to describe PURE semantics. Note, it would be possible to make this function ELEMENTAL but since CHAR cannot it was considered preferrable to preserve the strict identity of CHAR(VAR_STR(ch))=ch
6. Additional Generic Procedures for Input/Output
   1. The GET procedure
      No change
   2. The PUT procedure
      No change
   3. The PUT_LINE procedure
      No change
7. Additional Generic Procedures for Substring Manipulation
   1. The INSERT procedure
      No change
   2. The REPLACE procedure
      No change
   3. The REMOVE procedure
      No change
   4. The EXTRACT procedure
      No change
   5. The SPLIT procedure
      No change

Annex A : Module ISO_varying_string
As well as implementing the above changes, the type in the module is initialised to nullify the pointer component and both the assignment and get procedures check and if necessary deallocate the existing string before allocating it a new value.

Annex B : Examples
No change

These changes have been made to produce a new HTML version of the standard and a new version of the module. At the time of writing I have not been able to obtain access to an adequate F95 compiler and therefore the module code has not been fully tested. This will be done as soon as possible.

I regret I do not have time to complete the detailed editorial work necessary to render the text into the format and style required by ISO. I am happy to continue to maintain the Web version and the module source code, but to enable a published update to be produced WG5 will have to appoint an editor who can convert the HTML into ISO style.