

Date: 10 June 1999
To: J3
From: Jerry Wagener
Subject: Edits to Section 1

1. change the following sentence in section 1.1 (page 1)

A processor conforming to part 1 need not conform to ISO/IEC 1539-2; however, conformance to ISO/IEC 1539-2 assumes conformance to this part.

change to:

The third part, ISO/IEC 1539-3, defines a standard conditional compilation preprocessing facility for Fortran. A processor conforming to part 1 need not conform to either ISO/IEC 1539-2 or ISO/IEC 1539-3; however, conformance to either assumes conformance to this part.

2. Replace section 1.5.1 (page 3) with the following:

1.5.1 Fortran 90 compatibility

Except as noted in this section, this standard is an upward compatible extension to the preceding Fortran International Standard, ISO/IEC 1539:1997, informally referred to as Fortran 95. Any standard-conforming Fortran 95 program remains standard-conforming under this standard.

This standard has more intrinsic procedures than did Fortran 95. Therefore, a standard-conforming Fortran 95 program may have a different interpretation under this standard if it invokes an external procedure having the same name as one of the new standard intrinsic procedures, unless that procedure is specified in an EXTERNAL statement or an interface body.

3. Replace section 1.5.2 (pages 3-4) with the following (much of which simply updates "Fortran 95" to "Fortran 2000", but there are some other minor updates as well):

1.5.2 FORTRAN 77 and Fortran 90 compatibility

Except for the deletion features noted in Annex B.1, and except as noted in this section, the Fortran 2000 Standard is an upward compatible extension to the earlier Fortran International Standards, ISO 1539:1980 (FORTRAN 77) and ISO/IEC 1539:1991 (Fortran 90). Any standard-conforming FORTRAN 77 or Fortran 90 program that does not use one of the deleted features remains standard conforming under the Fortran 2000 Standard; however, see item (4) below regarding intrinsic procedures. The Fortran 2000 Standard restricts the behavior for some features that were processor dependent in FORTRAN 77. Therefore, a standard-conforming FORTRAN 77 program that uses one of these processor-dependent features may have a different interpretation under the Fortran 95 Standard, yet remain a standard-conforming program. The following FORTRAN 77 features have different interpretations in the Fortran 2000 Standard:

(1) FORTRAN 77 permitted a processor to supply more precision derived from a real constant than can be represented in a real datum when the constant is used to initialize a data object of type double precision real in a DATA statement. The Fortran 2000 Standard does not permit a processor this option.

(2) If a named variable that was not in a common block was initialized in a DATA statement and did not have the SAVE attribute specified, FORTRAN 77 left its SAVE attribute processor dependent. The Fortran 2000 Standard specifies (5.3.13) that this named variable has the SAVE attribute.

(3) FORTRAN 77 required that the number of characters required by the input list was to be less than or equal to the number of characters in the record during formatted input. The Fortran 2000 Standard specifies (9.5.4.4.2) that the input record is logically padded with blanks if there are not enough characters in the record, unless the PAD= specifier with the value 'NO' is specified in an appropriate OPEN statement.

(4) The Fortran 2000 Standard has more intrinsic procedures than did either FORTRAN 77 or Fortran 90. Therefore, a standard-conforming FORTRAN 77 or Fortran 90 program may have a different interpretation under the Fortran 2000 Standard if it invokes an external procedure having the same name as one of the new standard intrinsic procedures, unless that procedure is specified in an EXTERNAL statement as recommended for nonintrinsic functions in both the FORTRAN 77 and Fortran 90 standards.

(5) A value of 0 for a list item in a formatted output statement will be formatted in a different form for some G edit descriptors. In addition, the Fortran 2000 standard specifies how rounding of values will affect the output field form, but FORTRAN 77 did not address this issue: therefore, some FORTRAN 77 processors may produce a different output form than Fortran 2000 processors for certain combinations of values and G edit descriptors.

(6) If the processor can distinguish between positive and negative real zero, the behavior of the SIGN intrinsic function when the second argument is negative real zero is changed by this standard.