Date: 97 April 10
To: J3
From: Loren P. Meissner

Subject: Edits for M.5: Extend Max and Min Intrinsics to Character Type

References: X3J3/96-131r1

PROPOSAL:

Extend the intrinsic functions Max, Min, MaxLoc, MinLoc, MaxVal, and MinVal to accept arguments of character data type.

EDITS:

MAX (13.14.64):

Change "Result Characteristics" to:

Result Characteristics. Type and kind type parameter are the same as the arguments. For arguments of character type, the length of the result is the length of the longest argument.

MAXLOC (13.14.66):

Change "shall be of type integer or real" to "shall be of type integer, real, or character"

MAXVAL (13.14.67):

Change "shall be of type integer or real" to "shall be of type integer, real, or character"

Change "The result is of the same type and kind type parameter as ARRAY" to "The result is of the same type and type parameters as ARRAY"

Replace Result Value Case(i) and Case (ii) as follows (no change to Case(iii)):

Case(i): If the size of ARRAY is not zero, the result of MAXVAL (ARRAY) has a value equal to the maximum value of all the elements of ARRAY. If ARRAY has size zero and type integer or real, the result of MAXVAL (ARRAY) has the value of the negative number of the largest magnitude supported by the processor for numbers of the type and kind type parameter of ARRAY. If ARRAY has size zero and type character, the result of MAXVAL (ARRAY) has the value of a string of characters of length LEN (ARRAY) with each character equal to CHAR (0, KIND = KIND (ARRAY)).

Case(ii): The result of MAXVAL (ARRAY, MASK = MASK), if MASK has at least one true element, has a value equal to the maximum value of the elements of ARRAY corresponding to true elements of MASK. If MASK has no true elements and the type of ARRAY is integer or real, the result of MAXVAL (ARRAY, MASK = MASK) has the value of the negative number of the largest magnitude supported by the processor for numbers of the type and kind type parameter of ARRAY. If MASK has no true elements and the type of ARRAY is character, the result of MAXVAL (ARRAY, MASK = MASK) has the value of a string of characters of length LEN (ARRAY) with each character equal to CHAR (0, KIND = KIND (ARRAY)).

MIN (13.14.69):

Change "Result Characteristics" to:

Result Characteristics. Type and kind type parameter are the same as the arguments. For arguments of character type, the length of the result is the length of the longest argument.

MINLOC (13.14.71):

Change "shall be of type integer or real" to "shall be of type integer, real, or character"

MINVAL (13.14.72):

Change "shall be of type integer or real" to "shall be of type integer, real, or character"

Change "The result is of the same type and kind type parameter as ARRAY" to "The result is of the same type and type parameters as ARRAY"

Replace Result Value Case(i) and Case (ii) as follows (no change to Case(iii)):

Case(i): If the size of ARRAY is not zero, the result of MINVAL (ARRAY) has a value equal to the minimum value of all the elements of ARRAY. If ARRAY has size zero and type integer or real, the result of MINVAL (ARRAY) has the value of the positive number of the largest magnitude supported by the processor for numbers of the type and kind type parameter of ARRAY. If ARRAY has size zero and type character, the result of MINVAL (ARRAY) has the value of a string of characters of length LEN (ARRAY) with each character equal to CHAR (*n* - 1, KIND = KIND (ARRAY)) where *n* is the number of characters in the collating sequence for characters with the kind type parameter of ARRAY.

Case(ii): The result of MINVAL (ARRAY, MASK = MASK), if MASK has at least one true element, has a value equal to the minimum value of the elements of ARRAY corresponding to true elements of MASK. If MASK has no true elements and the type of ARRAY is integer or real, the result of MINVAL (ARRAY, MASK = MASK) has the value of the negative number of the largest magnitude supported by the processor for numbers of the type and kind type parameter of ARRAY. If MASK has no true elements and the type of ARRAY is character, the result of MINVAL (ARRAY, MASK = MASK) has the value of a string of characters of length LEN (ARRAY) with each character equal to CHAR (*n* - 1, KIND = KIND (ARRAY)) where *n* is the number of characters in the collating sequence for characters with the kind type parameter of ARRAY.

BACKGROUND:

Specification and Syntax were approved at Meeting 138 (August 1996) - see X3J3/96-131r1.

This Minor Technical Enhancement adds text to the specifications of several Max and Min intrinsic functions mentioned, to permit ordering of character arguments in a manner that is consistent with character relational operators: for example, when A and B are character strings Max (A, B) returns a string equivalent to A if the relational expression A > B is true.

Max and Min are applied elementally with padding to the longest argument, and are consistent with relational operators. For example, the value of

Max ((/ A, Z /), (/ BB, YY /))

is

(/ BB, Z# /)

where # denotes a blank character.

For integer or real arrays with no elements (after masking), Fortran 95 defines MaxVal as "the negative number of the largest magnitude supported by the processor" and MinVal as "the positive number of the largest magnitude supported by the processor" {and not as Huge, the largest model number}. A similar concept is needed for character arrays with no elements. For the implementation suggested here, the "most negative" value corresponds to a string of characters all equal to Char (0) and the "most positive" value corresponds to a string of characters all equal to Char (n - 1) where n is the number of characters in the collating sequence for the kind type parameter of the argument array. Similar terminology appears in the current definition of Char.

Regularity is the primary reason for this extension. In Fortran 95, these intrinsic functions accept arguments of the other two ordered intrinsic types, integer and real. This extension recognizes character data type as a first-class type and extends the ordering intrinsics so that they accept all intrinsic types for which they make sense. This extension facilitates constuction of families of similar routines in applications (such as sorting) for integer, real, and character data.