To: J3
From: Malcolm Cohen
Subject: Interpretation Replacement Pages for 04-007.
Date: 2005/07/25

This is rather in the way of an experiment. The rest of this paper consists of the interpretation edits included in Corrigendum 1, in a form suitable for insertion into (a loose-leaf!) 04-007. It needs to be printed single-sided for this to work.

If people find this useful, I can produce a version including the interpretation requests that are still in progress.

Interp **F03/0009**, Status: *Corrigendum 1.*

Ref: 4.5.3.3, constraint C453, [53:1]

Append to constraint: "It shall not have the VALUE attribute.",
Making the whole constraint read:

C453 The passed-object dummy argument shall be a scalar, nonpointer, nonallocatable dummy data object with the same declared type as the type being defined; all of its length type parameters shall be assumed; it shall be polymorphic (5.1.1.2) if and only if the type being defined is extensible (4.5.6). It shall not have the VALUE attribute.

Interp **F03/0007**, Status: *Corrigendum 1.*

Ref: 4.5.5.2, 5^{th} paragraph, [59:30]

Replace "first executable statement"
By "executable constructs",
Making the whole paragraph:

If a specification expression in a scoping unit references a function, the result is finalized before execution of the executable constructs within the scoping unit.

Interp **F03/0007**, Status: *Corrigendum 1.*

Ref: 4.5.5.2, after the 5^{th} paragraph, [59:30+]

Insert new paragraph:

If a specification expression in a scoping unit references a structure constructor, the entity created by the structure constructor is finalized before execution of the executable constructs in the scoping unit.

Interp **F03/0013**, Status: *Corrigendum 1.*

Ref: 5.1, constraint C509, [72:23]

Append to constraint: "It shall not have the VALUE attribute.",
Making the whole constraint read:

C509 (R501) An entity declared with the CLASS keyword shall be a dummy argument or have the ALLOCATABLE or POINTER attribute. It shall not have the VALUE attribute.

Interp **F03/0014**, Status: *Corrigendum 1.*

Ref: 5.1.2.5.1, constraint C542, [78:21-22]

Replace "a dummy... procedure"
By "declared only in a subprogram or interface body",
Making the whole constraint read:

C542 (R511) An explicit-shape array whose bounds are not initialization expressions shall be declared only in a subprogram or interface body.

Interp **F03/0014**, Status: *Corrigendum 1.*

Ref: 5.1.2.5.1, paragraph after constraint C452, [78:23]

Replace "subprogram"
By "subprogram or interface body",
Making the whole paragraph:

An **automatic array** is an explicit-shape array that is declared in a subprogram or interface body, is not a dummy argument, and has bounds that are not initialization expressions.

Interp **F03/0011**, Status: *Corrigendum 1.*

Ref: 6.3.1, constraint C625, [111:11-12]

Replace “unlimited polymorphic”
By “unlimited polymorphic or is of abstract type”,
Making the whole constraint read:

C625 (R623) If any *allocate-object* is unlimited polymorphic or is of abstract type, either *type-spec* or SOURCE= shall appear.

Interp **F03/0007**, Status: *Corrigendum 1.*

Ref: 6.3.3.1, 2^{nd} paragraph after Note 6.24, [116:]

Replace "first executable statement"
By "executable constructs",
Making the whole paragraph:

If a specification expression in a scoping unit references a function whose result is either allocatable or a structure with a subobject that is allocatable, and the function reference is executed, an allocatable result and any subobject that is an allocated allocatable entity in the result returned by the function is deallocated before execution of the executable constructs in the scoping unit.

Interp **F95/0030**, Status: *Corrigendum 1.*

Ref: 7.1.6, immediately before Note 7.10, [126:19+]

Insert new paragraph:

If a specification expression in a module includes a reference to a generic, that generic shall have no specific procedures defined in the module subsequent to the specification expression.

Interp **F95/0030**, Status: *Corrigendum 1.*

Ref: 7.1.7, immediately before Note 7.11, [127:33+]

Insert new paragraph:

If an initialization expression in a module includes a reference to a generic, that generic shall have no specific procedures defined in the module subsequent to the initialization expression.

Interp **F03/0006**, Status: *Corrigendum 1.*

Ref: 7.4.1.3, 1^{st} paragraph, [139:17]

Replace "the evaluation of all operations in *expr* and *variable*"
By "the evaluation of *expr* and the evaluation of all expressions in *variable*",
Making the whole paragraph:

Execution of an intrinsic assignment causes, in effect, the evaluation of the expression *expr* and all expressions within *variable* (7.1.8), the possible conversion of *expr* to the type and type parameters of *variable* (Table 7.9), and the definition of *variable* with the resulting value. The execution of the assignment shall have the same effect as if the evaluation of *expr* and the evaluation of all expressions in *variable* occurred before any portion of *variable* is defined by the assignment. The evaluation of expressions within *variable* shall neither affect nor be affected by the evaluation of *expr*. No value is assigned to *variable* if *variable* is of type character and zero length, or is an array of size zero.

Interp **F03/0006**, Status: *Corrigendum 1.*

Ref: 7.4.1.3, list item (2) after Note 7.39, [141:20,21,22]

Insert "the value of" before each of the three occurrences of "*expr*",
Making the whole paragraph:

(2) If the component of the value of *expr* is allocated, the corresponding component of *variable* is allocated with the same dynamic type and type parameters as the component of the value of *expr*. If it is an array, it is allocated with the same bounds. The value of the component of the value of *expr* is then assigned to the corresponding component of *variable* using defined assignment if the declared type of the component has a type-bound defined assignment consistent with the component, and intrinsic assignment for the dynamic type of that component otherwise.

Interp **F03/0015**, Status: *Corrigendum 1.*

Ref: 8.1.4.3, 1^{st} paragraph, [161:18-19]

Delete ", TARGET,", and after "the attribute.", insert new sentence:
"The associating entity has the TARGET attribute if and only if the selector is a variable and has either the TARGET or POINTER attribute.",
Making the whole paragraph:

Within a SELECT TYPE or ASSOCIATE construct, each associating entity has the same rank as its associated selector. The lower bound of each dimension is the result of the intrinsic function LBOUND (13.7.60) applied to the corresponding dimension of *selector*. The upper bound of each dimension is one less than the sum of the lower bound and the extent. The associating entity has the ASYNCHRONOUS or VOLATILE attribute if and only if the selector is a variable and has the attribute. The associating entity has the TARGET attribute if and only if the selector is a variable and has either the TARGET or POINTER attribute. If the associating entity is polymorphic, it assumes the dynamic type and type parameter values of the selector. If the selector has the OPTIONAL attribute, it shall be present.

Interp **F95/0096**, Status: *Corrigendum 1.*

Ref: 9.5.3.4.2, 8^{th} paragraph, [198:12]

Replace "input item and its corresponding data edit descriptor"
By "effective input item and its corresponding data edit descriptors",
Making the whole paragraph:

During nonadvancing input when the pad mode has the value YES, blank characters are supplied by the processor if an effective input item and its corresponding data edit descriptors require more characters from the record than the record contains. If the record is incomplete, an end-of-file condition occurs; otherwise an end-of-record condition occurs.

Interp **F95/0096**, Status: *Corrigendum 1.*

Ref: 9.10.3, list item (1), [218:6-7]

Replace "input list item (9.5.3.4.2) and corresponding data edit descriptor that requires"
By "effective input item (9.5.2) and its corresponding data edit descriptors that require",
Making the whole paragraph:

(1) If the pad mode has the value YES, the record is padded with blanks to satisfy the effective input item (9.5.2) and its corresponding data edit descriptors that require more characters than the record contains. If the pad mode has the value NO, the input list item becomes undefined.

Interp **F03/0044**, Status: *Corrigendum 1.*

Ref: 12.3.2.5, [266:8]

Change "referenced" to "invoked",
Making the whole paragraph:

In a scoping unit where the interface of a function is implicit, the type and type parameters of the function result are specified by an implicit or explicit type specification of the function name. The type, type parameters, and shape of dummy arguments of a procedure invoked from a scoping unit where the interface of the procedure is implicit shall be such that the actual arguments are consistent with the characteristics of the dummy arguments.

Interp **F03/0016**, Status: *Corrigendum 1.*

Ref: 12.4, after constraint C1224, [266:24+]

Insert new constraint:

C1224a (R1219) If *data-ref* is an array, the referenced type-bound procedure shall have the PASS attribute.

Interp **F03/0043**, Status: *Corrigendum 1.*

Ref: 12.4.1.1, [268:17]

Replace "procedure"
By "procedure or a procedure pointer component",
Making the whole paragraph:

In a reference to a type-bound procedure, or a procedure pointer component, that has a passed-object dummy argument (4.5.3.3), the *data-ref* of the *function-reference* or *call-stmt* is associated, as an actual argument, with the passed-object dummy argument.

Interp **F03/0010**, Status: *Corrigendum 1, MODIFIED.*

Ref: 12.4.1.2, 1^{st} paragraph, [268:23]

Before "the declared"
Insert "either both the actual and dummy arguments shall be unlimited polymorphic, or",
Making the whole paragraph:

If a dummy argument is neither allocatable nor a pointer, it shall be type compatible (5.1.1.2) with the associated actual argument. If a dummy argument is allocatable or a pointer, the associated actual argument shall be polymorphic if and only if the dummy argument is polymorphic, and either both the actual and dummy arguments shall be unlimited polymorphic, or the declared type of the actual argument shall be the same as the declared type of the dummy argument.

Interp **F03/0005**, Status: *Corrigendum 1.*

Ref: 12.4.1.2, paragraph after Note 12.22, [270:1-2]

Replace "associated with an actual argument that is"
By "used as an actual argument that is associated with",
Making the whole paragraph:

If the dummy argument does not have the TARGET or POINTER attribute, any pointers associated with the actual argument do not become associated with the corresponding dummy argument on invocation of the procedure. If such a dummy argument is used as an actual argument that is associated with a dummy argument with the TARGET attribute, whether any pointers associated with the original actual argument become associated with the dummy argument with the TARGET attribute is processor dependent.

Interp **F95/0078**, Status: *Corrigendum 1.*

Ref: 12.4.4.1, end of subclause, [278:5+]

Append new list item:

(5) If (1), (2), (3), and (4) do not apply, the name is that of an intrinsic procedure, and the reference is consistent with the interface of that intrinsic procedure, then the reference is to that intrinsic procedure.

Interp **F03/0054**, Status: *Corrigendum 1.*

Ref: 13.7.37, Result Value paragraph, [315:5-6]

Replace "model representation (13.4) for the value of X"
By "representation for the value of X in the model (13.4) that has the radix of X but no limits on exponent values",
Making the whole paragraph:

> **Result Value.** The result has a value equal to the exponent e of the representation for the value of X in the model (13.4) that has the radix of X but no limits on exponent values, provided X is nonzero and e is within the range for default integers. If X has the value zero, the result has the value zero. If X is an IEEE infinity or NaN, the result has the value HUGE(0).

Interp **F03/0054**, Status: *Corrigendum 1.*

Ref: 13.7.40, Result Value paragraph, [317:8]

Replace "model representation of X"
By "representation for the value of X in the model that has the radix of X but no limits on exponent values",
Making the whole paragraph:

> **Result Value.** The result has the value $X \times b^{-e}$, where b and e are as defined in 13.4 for the representation for the value of X in the model that has the radix of X but no limits on exponent values. If X has the value zero, the result has the value zero. If X is an IEEE infinity, the result is that infinity. If X is an IEEE NaN, the result is that NaN.

Interp **F03/0055**, Status: *Corrigendum 1.*

Ref: 13.7.100, Result Value paragraph, [347:22]

Replace "model representation of X"
By "value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken",
Making the whole paragraph:

> **Result Value.** The result has the value $|\mathrm{X} \times b^{-e}| \times b^{p}$, where b, e, and p are as defined in 13.4 for the value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken. If X is an IEEE infinity, the result is zero. If X is an IEEE NaN, the result is that NaN.

Interp **F03/0054**, Status: *Corrigendum 1.*

Ref: 13.7.107, Result Value paragraph, [351:5]

Replace "model representation of X"
By "representation for the value of X in the model that has the radix of X but no limits on exponent values",
Making the whole paragraph:

> **Result Value.** The result has the value $X \times b^{I-e}$, where b and e are as defined in 13.4 for the representation for the value of X in the model that has the radix of X but no limits on exponent values. If X has value zero, the result has value zero.

Interp **F03/0055**, Status: *Corrigendum 1.*

Ref: 13.7.113, Result Value paragraph, [353:9]

Replace "model representation of X"
By "value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken",
Making the whole paragraph:

> **Result Value.** If X does not have the value zero, the result has the value $b^{\max(e-p,e_{\mathrm{MIN}}-1)}$, where b, e, and p are as defined in 13.4 for the value nearest to X in the model for real values whose kind type parameter is that of X; if there are two such values, the value of greater absolute value is taken. If X has the value zero, the result is the same as that of TINY (X). If X is an IEEE infinity, the result is positive infinity. If X is an IEEE NaN, the result is that NaN.