Subject: Implicit kind, default kind
From: Van Snyder

Number

TBD

Title

Implicit kind, default kind.

Submitted By

J3

Status

For consideration.

Basic Functionality

Provide a facility to specify the kind type parameter values for entities of a particular type that are created if a declaration does not include a *kind-selector* or a literal does not include a *kind-param*. This specification is here called the **implicit kind**.

Provide a facility to specify the size of a numeric storage unit. This specification would simultaneously change the implicit kind type parameter values for integer, real, complex and logical types, and change the kind type parameter value implied by double precision so that it occupies twice as many numeric storage units as implicit real kind, which preserves storage association relationships; it would not affect the implicit kind for character type. If named constants for numeric default kind parameter values are put into ISO_FORTRAN_ENV, those values would be changed by this specification if they are accessed by use association in a scoping unit in which the default is changed. It *would not* change the representation method indicated by a particular value of a kind type parameter; such an implementation would break the one-to-one correspondence between kind type parameter values and representation methods, and thereby wreak havoc with separate compilation.

Rationale

In a carefully-written program, the author will give every entity an explicitly specified kind type parameter. Entities in each family of related entities are given their kind type parameter values by reference to a single named constant. This allows one to change the kind of each family of entities by changing the definition of a single named constant. Frequently, one family of entities appears far more frequently than any other. In these cases, it would be helpful if these entities could be declared without a *kind-selector* or *kind-param*, and yet still be able to change their kind type parameter values in one place without changing the kind type parameter values of unrelated entities. A specification of an **implicit kind** for a particular type, that applies if a declaration does not include a *kind-selector* or a literal does not include a *kind-param*, would permit this.

In many programs, especially so-called "legacy programs," entities are declared to be of default kind, or not declared at all. If and when it becomes necessary to migrate these programs to a different configuration of the platform for which they were developed, or to a platform with different characteristics, it is sometimes useful to consistently and simultaneously change the underlying meanings of "default kind" for numeric and logical entities. For example, one may need to migrate a program developed for a platform on which default integer type occupies 64 bits to a platform on which default integer type occupies 32 bits. The program may use storage association, including between different type and kind numeric variables, so it may be important to preserve the relation between storage sizes. One could

almost achieve the desired effect by carefully changing the implicit kinds for integer, real, complex and logical in a consistent way, but this is tedious and error prone, and cannot change the definition of double precision to occupy twice as many numeric storage units as implicit real kind.

Estimated Impact

This is a project of moderate size, both in terms of the standard and for implementors. Most of the changes in the standard will be in Section 4. One or two additional statements, or perhaps modifications of the IMPLICIT statement, will be needed.

Detailed Specification

Provide a facility to specify the kind type parameter values for entities of a particular type that are created if a declaration does not include a *kind-selector* or a literal does not include a *kind-param*. This specification is here called the **implicit kind**. Provide named constants in ISO_FORTRAN_ENV for the default kind type parameter values, since these cannot be gotten by reference to literals without *kind-param*s once the meaning of such is changed.

Provide a facility to specify the size of a numeric storage unit. This specification would simultaneously change the implicit kind type parameter values for integer, real, complex and logical types, and change the kind type parameter value implied by double precision so that it occupies twice as many numeric storage units as implicit real kind, which preserves storage association relationships; it would not affect the implicit kind for character type. It *would not* change the representation method indicated by a particular value of a kind type parameter; such an implementation would break the one-to-one correspondence between kind type parameter values and representation methods, and thereby wreak havoc with separate compilation.

History