****

J3/02-295r1

Date: October 2002
To: J3
From: Aleksandar Donev
Subject: Selecting the base component of polymorphic objects
Reference: J3-007R3, J3/02-294

****

Summary

Polymorphic objects of CLASS(My_Type) need to occasionally be used in places where a non-polymorphic object of the declared type TYPE(My_Type) is needed. The present standard allows these to be intermixable in certain occasions. After some discussion with DATA members, we concluded this introduced complexities and problems and should be prohibited. For example, generic resolution cannot add specifics with arguments of type TYPE(My_Type) once there is a generic with arguments of CLASS(My_Type). Kurt will deal with generics in a separate paper.

Proposed Solution

Instead, we should provide a way to cast a CLASS(My_Type) variable into a TYPE(My_Type) variable, and that can be used as needed. Introducing a base component, much alike the parent component, is a natural solution:

```
TYPE, EXTENSIBLE :: My_Type
   ...
END TYPE My_Type
CLASS(My_Type), ... :: polymorphic_variable

TYPE(My_Type) :: plain_variable
TYPE(My_Type), POINTER :: plain_pointer
INTERFACE
   SUBROUTINE MyProcedure(dummy)
      TYPE(My_Type), INTENT(INOUT) ::  dummy
   END SUBROUTINE
END INTERFACE

All of these are now valid:

plain_variable=polymorphic_variable%My_Type
plain_pointer=>polymorphic_variable%My_Type
CALL MyProcedure(plain_dummy=polymorphic_variable%My_Type)
```

This simplifies and makes the syntax more consistent. We should delete some provisions which do allow the mixing of TYPE(My_Type) and CLASS(My_Type), listed below.

Edits and Choices

Additions:

Every polymorphic variable should have a component named after its base_type, even if it is not of an extended type. So 53:13 should be replaced with (possibly split into two sentences):
‘‘An object of base/extended type has a scalar, nonpointer, nonallocatable, *base/parent component* with the type and type parameters of the base/parent type. The name of this component is the base/parent type name.’’
There is the possibility of only allowing such base component selection for polymorphic variables, but I see no reason why not to allow it for nonpolymorphic variables as well (see Note 74: 5.7)

Changes:

This is incomplete. We may heavily redo intrinsic assignment,
so this is only the beginning:
139:5 add: "variable and expression shall not be polymorphic"

265:9 Delete: "If the dummy argument is allocatable or pointer,"