Date: 5 August 1998
To: J3
From: Van Snyder
Subject: Remarks on Derived Type I/O
References: 98-007r2, 98-179

While pondering generic procedures and type-bound procedures, I noticed the following problems with derived type I/O (DTIO):

- The restriction added to the generic resolution rules at [305:34-35] in 98-007r2 makes it impossible to have DTIO routines for an extensible type and also a type extended from it.
- Invocation of DTIO routines does not dispatch when used with polymorphic variables, i.e. the appropriate routine is not selected dynamically at run-time.
- It is possible to access a type, but exclude the associated DTIO routines, by USE, ONLY.

These problems could be repaired by binding DTIO routines to the type, as has been proposed for operators, assignment and ordinary procedures in 98-179 and the papers it references. E.g.

```
TYPE, EXTENSIBLE :: T1
  ! Components
CONTAINS
  READ, FORMATTED => T1_READ_FORMATTED ! Similar for WRITE and UNFORMATTED
END TYPE T1
TYPE, EXTENDS(T1) :: T2
  ! More components
CONTAINS
  READ, FORMATTED => T2_READ_FORMATTED ! Over-rides T1_READ_FORMATTED
END TYPE T1
```

If type-bound procedures are adopted, I propose type-binding ought to be the *only* way to declare that routines are to be used for DTIO. That is, the newly defined interface block syntaxes to support DTIO should be removed.

The DTIO binding should be considered to have the PASS_OBJ annotation, and to be invoked with obj% before its name.