

Subject: Issue 313, part of issue 290
 From: Van Snyder

1 Edits

Edits refer to 01-007. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by immediately following text, while a page and line number followed by + (-) indicates that immediately following text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

1.1 Part of issue 290

Generic resolution for user-defined derived-type input/output has turned out to be so different from “ordinary” generic resolution that there is no hope of combining the discussions.

[Editor: Replace the text of the note by “Repair the subclause to account for defined operators and assignment.” and move it to [345:3+]. But don’t bother if 01-115 passes, because it deletes the note entirely.] 346:24-27

1.2 Issue 313

A binding of a type and a binding of an extension of that type correspond if the latter binding is the same binding as the former, overrides a corresponding binding, or is an inherited corresponding binding. 54:38 New ¶

[The meaning of “in” isn’t clear. Editor: Delete “in the derived-type definition” twice.] 253:1-2, 8-9

[The current text doesn’t require the inherited binding to be the same as the one specified by *binding-name* or one that overrides it. Use the newly-coined “corresponds” term:] 253:10-12

A procedure reference with a *binding-name* refers to the binding of the dynamic type of the *data-ref* that corresponds to the specified binding of the declared type of the *data-ref*. The binding of the dynamic type shall not be deferred, and the reference is to the procedure identified by that binding.

A suitable generic interface for user-defined derived-type input/output of an effective item has a *dtio-generic-spec* that is appropriate to the direction (read or write) and method (formatted, list-directed, namelist or unformatted) of the data transfer as specified in 9.5.4.4.3, and has a specific interface whose **dtv** argument is compatible with the effective item according to the rules for argument association in 12.4.1.2. 346:23+

if the input/output statement has a *format-specification*, the effective item’s corresponding edit descriptor is a DT edit descriptor, user-defined derived-type input/output occurs if a suitable generic interface is accessible. If the interface is type-bound, the reference is to the procedure identified by the binding of the dynamic type of the effective item that corresponds to the specified binding of the declared type of the effective item. The binding of the dynamic type shall not be deferred. Otherwise the reference is to the procedure identified by the specific interface in the interface block. 346:29-31

[Editor: Delete unresolved issue note 313.] 346:32-36

[Editor: Delete.]	346:37-5
[Editor: Replace sentence with “The reference shall not be to a deferred binding, a dummy procedure or dummy procedure pointer that is not present, or a disassociated procedure pointer.”]	347:7-8
[Editor: “	347:8
[Editor: Delete (covered above).]	347:9-10
C.10.2 Generic resolution and dynamic dispatch (14.1.2.4.3)	453:13+

A type-bound generic interface consists of a family of generic interfaces, one per type in the inheritance heirarchy. A declaration of an extensible type may override specific procedures in the generic interfaces inherited from its parent type, add new specific procedures, or add new type-bound generic interfaces.

When a procedure is invoked by way of a type-bound generic interface, one of the specific procedures of the generic interface bound to the declared type of the invoking object (or the **dtv** argument in the case of user-defined derived-type input/output) is selected according to the usual rules for generic resolution (14.1.2.4.1).

Once a specific procedure is selected for the declared type, the corresponding (4.5.3.2) procedure for the dynamic type is invoked. For polymorphic objects, it is expected that the former process is performed when the program is translated, and the latter occurs during program execution. For nonpolymorphic objects, the dynamic and declared types are the same, so the selection is completed during translation.

One possible method to support the polymorphic case is for the processor to construct a *dispatch table* for each type. The elements in the dispatch table are effectively procedure pointers. Either the specific name of a nongeneric binding or the generic identifier and the combination of characteristics used for generic resolution chooses a *slot* in the dispatch table.

During execution, the dynamic type of the object through which the procedure is invoked is used to select which dispatch table to use. One is assured that the slot selected in the first step exists in the dispatch table selected during execution because type extension can only add slots, it cannot delete them.

There is the possibility that a slot will be occupied by an indication that the procedure is deferred (4.5.1.5), in which case the program is in error. This indication is effectively equivalent to a null procedure pointer, and may in fact be represented in the same way.

1.3 Only vaguely related, but in the same place as issue 313

[Editor: To get all of the stuff introduced in 14.1.2.4.0 together, exchange subclauses 14.1.2.4.3 and 14.1.2.4.4 (after doing the above edits).]	346-347
---	---------