

Subject: Issues 287, 288, 294, 296 and part of 290: Type-bound operators, assignment and generic procedures
 From: Van Snyder
 References: 00-304r1

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.

[Editor: “A ... (12.3.2.1)” ⇒ “A subroutine and a generic interface (4.5.1.5, 12.3.2.1) whose generic specifier is ASSIGNMENT (=)”]	31:18
[Editor: Before “.” insert “(7.5.1.3).”]	31:20
[Editor: “ <i>construct</i> ” ⇒ “ <i>stmt</i> ” twice.]	41:15-16
R440 <i>proc-binding-stmt</i> is <i>specific-binding</i> or <i>generic-binding</i> or <i>final-binding</i>	41:20-21
[Editor: Move [41:33-35] to here.]	
R441 <i>specific-binding</i> is PROCEDURE [(<i>abstract-interface-name</i>)] ■	41:22
[Editor: Move [41:36] to here.]	41:23+
Constraint: The <i>abstract-interface-name</i> shall be specified if and only if the <i>binding</i> is NULL() and is not overriding an inherited binding. [This was formerly at [41:37-38].]	
[Editor: Move [42:28-29] to here.]	
R441 ¹ / ₃ <i>generic-binding</i> is GENERIC [(<i>abstract-interface-name</i>)] ■ ■ [, <i>binding-attr-list</i>] :: <i>generic-spec</i> => <i>binding-list</i>	41:24-26
Constraint: The <i>abstract-interface-name</i> shall be specified if and only if <i>binding-list</i> is a single binding which is NULL().	
Constraint: If <i>generic-spec</i> is <i>generic-name</i> , <i>generic-name</i> shall not be the name of a specific binding of the type.	
Constraint: If <i>generic-spec</i> is OPERATOR (<i>defined-operator</i>), the interface of each binding shall be as specified in 12.3.2.1.1.	
Constraint: If <i>generic-spec</i> is ASSIGNMENT (=), the interface of each binding shall be as specified in 12.3.2.1.2.	
Constraint: If <i>generic-spec</i> is <i>dtio-generic-spec</i> , the interface of each binding shall be as specified in 9.5.4.4.3. The type of the dtv argument shall be <i>type-name</i> .	
R441 ² / ₃ <i>final-binding</i> is FINAL [::] <i>final-subroutine-name-list</i>	
[Editor: Delete unresolved issue note 287.]	41:27-32
[Editor: Delete. Note that [41:33-36] have been moved upward.]	41:37-40
[Editor: Delete unresolved issue note 288.]	42:1-11
[Editor: Delete.]	42:12-17

Constraint: PASS_OBJ shall not be specified in a <i>generic-binding</i> that has a <i>dtio-generic-spec</i> .	42:34-35
Constraint: PASS_OBJ shall be specified in a <i>generic-binding</i> that has OPERATOR (<i>defined-operator</i>) or ASSIGNMENT (=).	
Constraint: PASS_OBJ shall be specified for an overriding binding if and only if it is specified for the binding being overridden.	
Constraint: PASS_OBJ shall be specified for a <i>generic-binding</i> if and only if it is specified for all generic bindings, both inherited and declared within the type definition, with the same <i>generic-spec</i> .	
Constraint: Within the <i>specification-part</i> of a module, each <i>generic-binding</i> shall specify the same accessibility, either explicitly or implicitly, as every other <i>generic-binding</i> in the same type definition that has the same <i>generic-spec</i> .	
[Editor: Delete.]	42:45-43:2
[Editor: “-construct” ⇒ “-stmt”.]	47:12
A <i>generic-binding</i> specifies a type-bound generic interface.	47:12+
[Editor: Add a reference to the above paragraph to the index item for “generic interface”.]	Same ¶
The interface of a binding is the interface of the procedure specified by <i>procedure-name</i> or the interface of the abstract interface specified by <i>abstract-interface-name</i> .	
The scope of a <i>generic-spec</i> that is a name is the derived-type definition. The scope of a <i>generic-spec</i> that is not a name is all scopes in which the derived type or any entity of the type is accessible.	47:25+
The same <i>generic-spec</i> may be used in several <i>generic-bindings</i> within a single derived-type definition.	New ¶
[Editor: Delete.]	47:31-38
[Editor: After “deferred.” insert “A deferred binding that is not a <i>dtio-generic-spec</i> shall not be referenced by a procedure reference.”]	47:40
The overriding binding and the inherited binding shall satisfy the following conditions:	53:1-3
(1) Either both shall specify PASS_OBJ or neither shall.	
[Editor: Renumber as items 2, 3 and 4.]	53:4-7
(5) The corresponding dummy arguments shall have the same names and characteristics, except for the type of the passed-object dummy arguments.	53:8-9
[Editor: Renumber as items 6 and 7.]	53:10-12
A generic binding overrides an inherited binding if they both have the same <i>generic-spec</i> and satisfy the above conditions for overriding. A generic binding with the same <i>generic-spec</i> that does not satisfy the conditions extends the generic interface.	53:34-38
[Editor: “an interface block (” ⇒ “a generic interface (4.5.1.5,”.]	56:29
[Editor: “block (” ⇒ “(4.5.1.5,”.]	112:16
[Editor: Delete “block” thrice.]	112:18-22
[Editor: Delete unresolved issue note 294.]	114:1-5

[Editor: “an interface block (” ⇒ “a generic interface (4.5.1.5,”.]	126:35
[Editor: “an interface block (” ⇒ “a generic interface (4.5.1.5,”.]	127:5
[Editor: “interface block with ASSIGNMENT (=) specifier” ⇒ “generic interface with a generic specifier of ASSIGNMENT (=)”.]	130:2
[Editor: “an interface block (” ⇒ “a generic interface (4.5.1.5,”.]	130:16
[Editor: “An interface block (” ⇒ “A generic interface (4.5.1.5,”.]	132:26
[Editor: “interface block” ⇒ “generic interface”.]	247:27-28
If ASSIGNMENT (=) is specified in a generic specification, all of the procedures in the generic interface [NOTICE that the word “block” is intentionally deleted!]	248:28
[Make exposition of defined assignment more parallel to exposition of defined operators. See [248:14-15]. Editor: Before “Each” insert “Defined assignment may, as with generic names, apply to more than one subroutine, in which case it is generic in exact analogy to generic procedure names.”]	248:29
[Editor: Delete “As ... name”. Adequately addressed in 14.1.2.3.]	249:13-15
[Editor: Delete unresolved issue note 296.]	249:16-25
[Editor: Delete – redundant.]	249:26-31
R1214 ¹ / ₂ <i>abstract-interface-name</i> is <i>name</i>	250:43
Constraint: The <i>name</i> shall be the name of an abstract interface (12.3.2.1).	
[Editor: Edit the edit in paper 01-105r1 by inserting “nongeneric” before “ <i>binding-name</i> ”.]	253:10-12
If the procedure reference has a generic <i>binding-name</i> , then	253:12+ New ¶
(1) If the reference is consistent with one of the specific interfaces in the generic binding of the declared type of the <i>data-ref</i> whose <i>generic-name</i> is the same as <i>binding-name</i> , the corresponding binding of the dynamic type of <i>data-ref</i> is selected.	
(2) Otherwise, the reference shall be consistent with an elemental reference to one of the specific interfaces in the generic binding of the declared type of the <i>data-ref</i> whose <i>generic-name</i> is the same as <i>binding-name</i> ; the corresponding binding of the dynamic type of <i>data-ref</i> is selected.	
The selected binding shall not be deferred.	
[Editor: Delete unresolved issue note 290.]	346:24-27
deferred binding (4.5.1.5): a type-bound procedure binding that specifies the NULL() intrinsic. A deferred binding shall not be invoked.	402:14+
[Editor: “an interface block” ⇒ “a generic interface”.]	402:23
[Editor: “an interface block” ⇒ “a generic interface”.]	403:10
generic interface (4.5.1.5, 12.3.2.1): An interface specified by a generic procedure binding or a generic interface block.	
[Editor: After “type” insert “, as a defined operator, or by defined assignment”.]	409:31

[Editor: After “module” insert “, or a generic binding may be specified in a derived type 445:4 definition”.]