

Subject: Unresolved issues 17 and 18
 From: Van Snyder
 References: 00-180r2 00-233 00-251

1 Introduction

Unresolved issue 17 remarks that the terminology to access the data objects inherited from the parent type is inconsistent. It refers to the “parent component” while simultaneously denying that the “parent component” is a component.

This paper changes the model for the components of a type extension, from inheritance to embedding plus scope enlargement. It specifies that there is a single nonpointer nonallocatable scalar component of the parent type name, called the **parent component**, and that the scope of a component name includes all of the extension types of the type in which it is declared. The components of the parent type may therefore be referenced with or without qualifying by the parent type name.

Unresolved issue 18 remarks that a normative paragraph should be a constraint. By defining the parent component, the substance of that paragraph is covered by the scoping rules in section 14, which have the same strength as constraints.

2 Edits

Edits refer to 00-007r2. 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 the indicated line. Remarks for the editor are noted in the margin, or appear between [and] in the text.

[Editor: After “type” insert “, including components declared in ancestor types (4.5.3),”.]	40:25
[Editor: Add in the same paragraph: “An ancestor type is the parent type or an ancestor type of the parent type.”]	52:36
[Editor: Delete “, components, ” twice.]	52:40,42
[Editor: Delete “intrinsic ... declared”. The relation between component order and intrinsic input/output is addressed by the edit indicated in paper00-233 for [183:37-40]. The description of structure construction is broken, and is being addressed by paper 00-251.]	52:46-49
In addition to specifying the parent type, the EXTENDS <i>type-attr-spec</i> declares a nonpointer nonallocatable scalar component called the parent component . It has the name, type and type parameters of the parent type. An ancestor component is the parent component, or an ancestor component of the parent type.	53:4-6
[Editor: Insert “component, ancestor” and “component, parent” into the index.]	

NOTE 4.41_½

The scope of a component includes all of the extensions of the type in which it is declared (14.1.2.5). Therefore, an accessible component of the parent type can be referenced (6.1.2) by qualifying it with an object of the extension type, followed by zero or more accessible names of ancestor components of which it is a component, in the reverse order of the type extension. By way of example, suppose that type **A** has a public component **X**, that type **B** extends type **A**, that type **C** extends type **B**, that the ancestor components of **C** are public, and that **Y** is an object of type **C**. Then **Y%*X*** may also be accessed as **Y%B%*X***, **Y%A%*X***, or **Y%B%A%*X***.

[Editor: Delete unresolved issue note 17.]	53:7-16
[Editor: “subobject denoted by the parent type name” ⇒ “parent component”.]	53:17
[Editor: Make this paragraph a note because it is covered by the scoping rules in section 14.)]	53:20-22
[Editor: Delete unresolved issue note 18.]	53:23-25
[Editor: After “type” insert “and all of its extension types”.]	342:5
[Editor: “a type definition” ⇒ “the definition of the type in which its declaration appears, and the definition of every extension of that type”.]	346:34
[Editor: “that type” ⇒ “one of those types”.]	346:35
[Editor: “the type” ⇒ “one of those types”.]	346:36
ancestor component (4.5.3.1): The <i>parent component</i> or an <i>ancestor component</i> of the <i>parent component</i> .	397:37+
ancestor type (4.5.3): The <i>parent type</i> or an <i>ancestor type</i> of the <i>parent type</i> .	
[Editor: After “type” insert “, including <i>components</i> declared in <i>ancestor types</i> ,”.]	400:37
parent component (4.5.3.1): A <i>scalar component</i> of an <i>extension type</i> that has the same <i>name</i> , <i>type</i> and <i>type parameters</i> as the <i>parent type</i> , and includes the <i>components</i> of the <i>parent type</i> .	404:25+