

## Resolution of Mantis items 2575, 2608: specifying access to class value parameters and type parameters and class local parameters.

Ballot ids: 50, 52, 55, 59

### Section 8.5

Change title of 8.5:

REPLACE:

**8.5 Object properties**

WITH:

**8.5 Object properties and object parameter data**

After : There are no restrictions on the data type of a class property.

Add:

The parameter data values of an object can also be accessed by qualifying the class value parameter or local value parameter name with an instance name. Example:

```
class vector #(parameter width = 7);  
endclass
```

```
vector #(3) v = new;  
initial $display (v.width);
```

Such an expression is not a constant expression.

### Section 8.10 This

REPLACE:

The **this** keyword is used to unambiguously refer to class properties or methods of the current instance.

WITH:

The **this** keyword is used to unambiguously refer to class properties, [value parameters](#), [local value parameters](#) or methods of the current instance.

.

### Section 8.14 Super

REPLACE:

The **super** keyword is used from within a derived class to refer to members of the base class. It is necessary to use **super** to access members of a base class when those members are overridden by the derived class.

WITH:

The **super** keyword is used from within a derived class to refer to members, [class value parameters](#) or [local value parameters](#) of the base class. It is necessary to use **super** to access members, [value parameters](#) or [local value parameters](#) of a base class when those ~~members~~ are overridden by the derived class. [An expression using super to access the value parameter or local value parameter is not a constant expression.](#)

REPLACE:

The member can be a member declared a level up or be inherited by the class one level up.

WITH:

The member, [value parameter or local value parameter](#) can be ~~a-member~~ declared a level up or be inherited by the class one level up

## Section 8.17

REPLACE:

In SystemVerilog, unqualified class properties and methods are public, available to anyone who has access to the object's name.

WITH:

In SystemVerilog, unqualified class properties and methods are public, available to anyone who has access to the object's name. [Class parameters and class local parameters are also public.](#)

## Section 8.22

In 8.22, REPLACE

Because classes and other scopes can have the same identifiers, the class scope resolution operator uniquely identifies a member of a particular class. In addition to disambiguating class scope identifiers, the :: operator also allows access to static members (class properties and methods) from outside the class, as well as access to public or protected elements of a superclass from within the derived classes.

WITH

Because classes and other scopes can have the same identifiers, the class scope resolution operator uniquely identifies a member, [a parameter or local parameter](#) of a particular class. In addition to disambiguating class scope identifiers, the :: operator also allows access to static members (class properties and methods) , [class parameters, and class local parameters](#) from outside the class, as well as access to public or protected elements of a superclass from within the derived classes. [A class parameter or local parameter is a public element of a class. A class scoped parameter or local parameter is a constant expression.](#)

REPLACE:

In SystemVerilog, the class scope resolution operator applies to all static elements of a class: static class properties, static methods, typedefs, enumerations, structures, unions, and nested class declarations.

#### WITH

In SystemVerilog, the class scope resolution operator applies to all static elements of a class: static class properties, static methods, typedefs, enumerations, [parameters](#), [local parameters](#), [constraints](#), structures, unions, and nested class declarations.

#### REPLACE:

The class scope resolution operator enables the following:

- Access to static public members (methods and class properties) from outside the class hierarchy.
- Access to public or protected class members of a superclass from within the derived classes.
- Access to type declarations and enumeration named constants declared inside the class from outside the class hierarchy or from within derived classes.

#### WITH:

The class scope resolution operator enables the following:

- Access to static public members (methods and class properties) from outside the class hierarchy.
- Access to public or protected class members of a superclass from within the derived classes.
- Access to [constraints](#), type declarations and enumeration named constants declared inside the class from outside the class hierarchy or from within derived classes.
- [Access to parameters and local parameters declared inside the class from outside the class hierarchy or from within derived classes.](#)

#### REPLACE:

Nested classes shall have the same access rights as methods do in the containing class. They have full access rights to **local** and **protected** methods and properties of the containing class. Nested classes have lexically-scoped, unqualified access to the **static** properties and methods of the containing class

#### WITH:

Nested classes shall have the same access rights as methods do in the containing class. They have full access rights to **local** and **protected** methods and properties of the containing class. Nested classes have lexically-scoped, unqualified access to the **static** properties and methods, [parameters](#), and [local parameters](#) of the containing class.

## In

### Section A.9.3 Identifiers

#### REPLACE

```
ps_parameter_identifier ::=  
[ package_scope ] parameter_identifier  
  
| { generate_block_identifier [ [ constant_expression ] ] . } parameter_identifier
```

#### WITH:

```
ps_parameter_identifier ::=  
[ package_scope+ | class_scope ] parameter_identifier
```

| { generate\_block\_identifier [ [ constant\_expression ] ] . } parameter\_identifier