

The New C Standard (Excerpted material)

An Economic and Cultural Commentary

Derek M. Jones

derek@knosof.co.uk

3.17.3

unspecified value

unspecified value

valid value of the relevant type where this International Standard imposes no requirements on which value is chosen in any instance

76

unspecified behavior

Commentary

Like unspecified behavior, unspecified values can be created by strictly conforming programs. Making use of such a value is by definition dependent on unspecified behavior.

Coding Guidelines

unspecified behavior

In itself the generation of an unspecified value is usually harmless. However, a coding guideline's issue occurs if program output changes when different unspecified values, chosen from the set of values possible in a given implementation, are generated. In practice it can be difficult to calculate the affect that possible unspecified values have on program output. Simplifications include showing that program output does not change when different unspecified values are generated, or a guideline recommendation that the construct generating an unspecified value not be used. A subexpression that generates an unspecified value having no affect on program output is dead code.

dead code

Example

```

1  extern int ex_f(void);
2
3  void f(void)
4  {
5  int loc;
6  /*
7   * If a call to the function ex_f returns a different value each
8   * time it is invoked, then the evaluation of the following can
9   * yield a number of different possible results.
10  */
11  loc = ex_f() - ex_f();
12  }
```

NOTE An unspecified value cannot be a trap representation.

77

correct program

Commentary

Unspecified values can occur for correct program constructs and correct data. A trap representation is likely to raise an exception and change the behavior of a correct program.

References