

Slay Some Earthly Demons XVI (Updates n3420)

Document: n3481 Author: Martin Uecker Date: 2025-02-15

Changes: Make it a generic constraint not attached to a syntax production.

Undefined Behavior:

- (19) A non-array lvalue with an incomplete type is used in a context that requires the value of the designated object (6.3.2.1).

Example:

```
struct f *p;
void g(void)
{
    *p;
}
```

Godbolt Link

Analysis:

This is an unused extension point.

This was part of Demons I, but it turned out that this requires more work because the proposed solution to change the definition of lvalue to require a complete type had unintended consequences. This version follows the alternative proposed in reflector message 25600 to only add a specific constraint without changing the definition of lvalue.

Wording (N3467)

6.3.3 Other operands

6.3.3.1 Lvalues, arrays, and function designator

2 Except when it is the operand of the

- sizeof operator,
- the typeof operators,
- the unary & operator,
- the ++ operator,
- the -- operator,
- the left operand of the . operator,
- or, an assignment operator,

an lvalue that does not have array type is converted to the value stored in the designated object (and is no longer an lvalue); this is called lvalue conversion. If the lvalue has qualified type, the value has the unqualified version of the type of the lvalue; additionally, if the lvalue has atomic type, the value has the non-atomic version of the type of the lvalue; otherwise, the value has the type of the lvalue. ~~If the lvalue has an incomplete type and does not have array type, the behavior is undefined.~~ If the lvalue (that does not have array type) designates an object of automatic storage duration that never had its address taken, and that object is uninitialized (not declared with an initializer and no assignment to it has been performed prior to use), the behavior is undefined. ..

6.5 Expressions

6.5.1 General

Constraints

1 During lvalue conversion, an lvalue shall not have an incomplete type.

Semantics