# Earthly Demon: Conversion of Pointers to Integers of Insufficient Size

Author: Martin Uecker
Number: N3712
Date: 2025-09-21

**Charter Principle:**

Avoid ambiguities: *Undefined behaviors*, *unspecified behaviors*, *implementation-defined behaviors*, and other *portability issues* enumerated in Annex J of the Standard should be eliminated or reduced. These issues might lead to application vulnerabilities.

**Annex J.2 (C23):**

(23) Conversion of a pointer to an integer type produces a value outside the range that can be represented (6.3.2.3).

**Example:**

```
int foo(int *p)
{
      return (char)p;
}
```

https://godbolt.org/z/vbGGvKxYP

Many compilers already warn for this by default.

**Recommendation:** Make this a constraint violation.

**Proposed Wording Change (relative to N3550)**

6.3 Conversions
6.3.3 Other operands
6.3.3.3 Pointers

6 Any pointer type can be converted to **bool or to** an integer type **that is able to represent all values of the pointer type.** Except as previously specified **for converstions to bool,** the result is implementation-defined. ~~**If the result cannot be represented in the integer type, the behavior is undefined. The result is not required to be in the range of values of any integer type.**~~ **It is not required that there exists an integer type that can represent all values of the pointer type.**

6.5.5 Cast operators

Constraints

4 A pointer type shall be converted only to void, ~~**an integer type, or**~~ a pointer type, **or an integer type that is able to represent all values of the pointer type.** Only a pointer, integer, or nullptr_t type shall be converted to a pointer type. The type nullptr_t shall not be converted to any type other than void, bool or a pointer type. If the target type is nullptr_t, the cast expression shall be a null pointer constant or have type nullptr_t.