

Document Number: WG14 N465/X3J11 95-066

C9X Revision Proposal

=====

Title: Data Representation Extensions (REP)
Author: Frank Farance, William Rugolsky, Jr.
Author Affiliation: Farance Inc.
Postal Address: 555 Main Street, New York, NY, 10044-0150, USA
E-mail Address: frank@farance.com, rugolsky@farance.com
Telephone Number: +1 212 486 4700
Fax Number: +1 212 759 1605
Sponsor: X3J11
Date: 1995-08-25
Proposal Category:

- Editorial change/non-normative contribution
- Correction
- ☒ New feature
- Addition to obsolescent feature list
- Addition to Future Directions
- Other (please specify) _____

Area of Standard Affected:

- Environment
- ☒ Language
- Preprocessor
- Library
- Macro/typedef/tag name
- Function
- Header

Prior Art: IEEE 1596.5 standard

Target Audience: C programmers, network and database applications

Related Documents (if any): IEEE 1596.5, SBEIR proposal, OAX proposal

Proposal Attached: ☒ Yes ☐ No, but what's your interest?

Abstract:

This proposal enhances the features of the SBEIR (Specification-Based Extended Integer Range) proposal and the OAX (ordering and alignment extensions) proposal by adding storage qualifiers for data representation. This proposal (and OAX) should be incorporated into the SBEIR proposal. The proposal is presented separately because the feature is conceptually, semantically, and syntactically separate. The problem this proposal solves is defining the representation of integral values to facilitate data interchange across heterogeneous systems.

Like the OAX proposal, a common misconception is that these features can be provided by the programmer: the programmer manually packs and unpacks the bits to create, say, a two complement bit pattern of a number. While the programmer can do this manually, the code becomes error-prone and non-obvious.

There are several common data representations (character, integer, floating). This proposal only addresses integers.

Integers are usually represented in twos complement (most common), ones complement (some implementations), or signed magnitude (few implementations). While it is possible to standardize all three representations, only twos complement notation is proposed because it is by far the most common used for data representation.

The storage qualifier "twoscomp" is used to specify this data representation. When reading or writing values, the compiler generates the appropriate conversion code.

If the twos complement representation of an "int" differs in range from the native "int", the value is promoted to "long" according to the value preserving rules.

When demoting a value to a twos complement representation, the truncation is performed via modulo arithmetic (like unsigned values). There are no overflow or exceptions possible with twos complement representation: all arithmetic is performed via modulo arithmetic.

The twos complement representation must use all bits in the container it occupies, i.e., "CHAR_BIT*(sizeof(x))" bits.

The following are examples of IEEE 1596.5 types as bound to C with SBEIR, OAX, and REP features.

```
/* big endian, signed 16-bit value, aligned to two bytes */
typedef twoscomp bigend align:2 signed int exact:16
    AlignedBigSignedDoublet;
```

```
/* little endian, unsigned 32-bit value, aligned to one byte */
typedef twoscomp littleend align:1 unsigned int exact:32
    UnalignedLittleUnsignedQuadlet;
```