

**Proposal for C23
WG14 N2849**

Title: Type annex tgmth narrowing macros with integer args
Author, affiliation: C FP group
Date: 2021-10-08
Proposal category: Technical
Reference: N2601, N2596

This proposal refers to X.13 in N2601, the annex for IECC 60559 interchange and extended types.

In [SC22WG14.20399] Joseph Myers reported a case that is not properly covered by the `<tgmth.h>` rules for functions that round to narrower type, namely the case where all arguments are of integer type and the macro prefix is `dN` indicating a decimal floating type.

The problem exists because the first step of the type determination rules applies the usual arithmetic conversions which handle integer-type arguments based solely on argument types. If all arguments are of integer type, these rules determine type `double`. For `dN`-prefixed macros that round to narrower type there is no corresponding function with `double` parameters, and the behavior is undefined. The intended type for this case was `_Decimal64`.

With the suggested change below, the rules first treat the case where all arguments are of integer type first and base the determined type on the prefix, and then apply the usual arithmetic conversions to all other cases.

Suggested changes:

Change the first bullet in X.13 #5:

— First-apply the rules (for determining the corresponding real type of the generic parameters) in 7.25 for macros that do not round result to narrower type, using the usual arithmetic conversion rules in X.4.2, to obtain ~~a preliminary type *P* for the generic parameters.~~

to:

— First, obtain a preliminary type *P* for the generic parameters: if all arguments are of integer type, then *P* is `double` if the macro prefix is `f`, `d`, or `fN` and *P* is `_Decimal64` if the macro prefix is `dN`; otherwise (if some argument is not of integer type), apply the rules (for determining the corresponding real type of the generic parameters) in 7.25 for macros that

do not round result to narrower type, using the usual arithmetic conversion rules in X.4.2, to obtain *P*.