

How do I access the magic IEEE floating point values like NaN in code?

 devblogs.microsoft.com/oldnewthing/20100305-00

March 5, 2010



Raymond Chen

There are functions like `_isnan`, `_isnanf`, `_finite`, and `_fpclass` for detecting that a floating point value is one of the special values like NaN, but how do you actually generate one of these values?

You can access these values from the `std::numeric_limits` template.

```
std::numeric_limits<float>::infinity(); // positive infinity
std::numeric_limits<float>::quiet_NaN(); // non-signalling NaN
```

Wait, where's negative infinity? The compiler folks provided these handy little definitions for when you need to generate a special value (as opposed to merely detecting one), and for which the `numeric_limits` template comes up short.

```
DECLSPEC_SELECTANY extern const float FLOAT_POSITIVE_INFINITY = ((float)(1e308 *
10));
DECLSPEC_SELECTANY extern const float FLOAT_NEGATIVE_INFINITY = ((float)(-1e308 *
10));
DECLSPEC_SELECTANY extern const float FLOAT_NaN = ((float)((1e308 * 10)*0.));
```

Disclaimer: Applies to Microsoft Visual Studio. Your mileage may vary. Use the template when available.

Bonus chatter: Note that you must use functions like `_isnan` to detect special values, because floating point special values behave very strangely in comparisons. (For example, NaN does not compare equal to itself!)

[Raymond Chen](#)

Follow

