Behavior of long and ulong in JScript .NET

October 3, 2002

There is a discontinuity in behavior between (u)long and the other types because long and ulong values may loose precision when coerced to Number. This document describes the behavior of the existing JScript .NET implementation.

Numeric Literals
The type of numeric literals that contain a decimal or an exponent is double.
The type of all other numeric literals is the first of [int, long, ulong, double] that can fit the value. In this case, if the type must be double, the value is guaranteed to be the same as if the literal contained a decimal.
Numeric literals are always scanned as positive numbers. Negative numbers are handled by the parser as a unary minus expression.

Coercions
Implicit conversions from longs and ulong to floating point always succeed.
Implicit coercions from floating point to long or ulong or between long and ulong succeeded iff the value can be exactly represented. Otherwise a type mismatch exception is thrown. e.g. a floating point number must not have a fraction part to be coerced to a long.
If the coercion is detected at compile time, the compiler emits a warning that the conversion could fail at runtime.
Explicit coercions always succeed and truncate as needed.

Operators
Unary minus uses the following algorithm.
        if op is 0 then return -double(0)
        else If -op fits in a long, return long(-op)
        else return -double(op).

Unary minus always returns double for the early bound ulong case.

Binary arithmetic operations (+, -, *, /, %), one of three overloads for the operation is chosen.
- long: long X long → Object
- ulong: ulong X ulong → Object
- double: double X double → Object

The following algorithm is used to decide which overload.
    if either operand is a double then the double overload is chosen
    else if one operand is ulong then
        if the other operand is negative then the double overload is chosen
        else the ulong overload is chosen
    else the long overload is chosen.

Both operands are coerced as appropriate for the given overload.

The double overloads behave as specified in E3. The long overload will usually return the result as a long if the result fits in a long. The ulong overload will usually return the result as a ulong if the result fits in a ulong. In both cases, if the result doesn't fit, the result of the double overload is returned. The exceptions to this behavior are as follows:
- The division operation always returns the result of the double overload.
- If the result of an multiplication or modulo operation is 0, then the result of the double overload is returned.
- If the right operand of modulo is 0, NaN is returned.

Bitwise shift operators coerce always coerce both operands to to double, and then execute as defined in E3.

The logical operators operate on the full 64bits found in long and ulong values.

The comparison operators will coerce both operands to double if either operand is a double. For the long, ulong comparisons, the result compares as expected on the numerical value.