

## Integer Expressions in Wave Assignment Statements

When compiling a wave assignment statement, the right-hand expression is compiled as 64-bit integer if the compiler knows that the destination wave is 64-bit integer. Otherwise, the expression is compiled as double. For example:

```
Function ExpressionCompiledAsInt64()
    Make/O/N=1/L w64
    w64 = 0x7FFFFFFFFFFFFFFF
    Print w64[0]
End
```

0x7FFFFFFFFFFFFFFF is the largest signed 64-bit integer value expressed in hexadecimal. It can not be precisely represented in double precision.

Here the /L flag tells Igor that the wave is signed 64-bit integer. This causes the compiler to compile the right-hand expression using signed 64-bit signed operations. The correct value is assigned to the wave and the correct value is printed.

In the next example, the compiler does not know that the wave is signed 64-bit integer because the /L flag was not used. Consequently the right-hand expression is compiled using double-precision operations. Because 0x7FFFFFFFFFFFFFFF can not be precisely represented in double precision, the value assigned to the wave is incorrect, as is the printed value:

```
Function ExpressionCompiledAsDouble()
    Make/O/N=1/Y=(0x80) w64      // Wave is int64 but Igor does not know it
    w64 = 0x7FFFFFFFFFFFFFFF    // Expression compiled as double
    Print w64[0]
End
```

You can use the /L flag in a wave declaration. For example:

```
Function ExpressionCompiledAsInt64()
    Make/O/N=1/Y=(0x80) w64      // Wave is int64 but Igor does not know it
    Wave/L w = w64              // Igor knows that w refers to int64 wave
    w = 0x7FFFFFFFFFFFFFFF      // Expression compiled as int64
    Print w[0]
End
```

This tells Igor that the wave reference by w is signed 64-bit integer, causing the compiler to use signed 64-bit integer to compile the right-hand expression.

If the wave was unsigned 64-bit, we would need to use Wave/L/U.

In summary, to assign values to a 64-bit integer wave, make sure to use the correct flags so that Igor will compile the right-hand expression using 64-bit integer operations.

To view the contents of an integer wave, especially a 64-bit integer wave, display it in a table using an integer or hexadecimal column format or print individual elements on the command line. Print on an entire wave is not yet integer-aware and uses doubles.

## Complex Integer Expressions

In Igor Pro 7, you could write wave assignment statements for complex integer waves up to 32 bits. For example:

```
Make/O/I/C complex32BitWave = cmplx(p,p)
```

In Igor Pro 8 or later, you can also write wave assignment statements for 64-bit complex integer waves. For example:

```
Make/O/L/C complex64BitWave = cmplx(p,p) // This was not supported in Igor7
```