

# My initial confusion when encountering functions whose names begin with Co

[devblogs.microsoft.com/oldnewthing/20190715-00](https://devblogs.microsoft.com/oldnewthing/20190715-00)

July 15, 2019



Raymond Chen

My formal education is in mathematics. In this highly abstract world, you spend a lot of time working with various types of duality and dual spaces. In general terms, a dual of a concept  $X$  is a special type of function (a *homomorphism*) that takes an  $X$  and produces a scalar. The convention in mathematics for dual spaces is that the *co-* prefix denotes the dual.

For example, in linear algebra, the dual of a *vector* is a *covector*: A *covector* is a special kind of function (*linear operator*) that operates on vectors. (If you're an engineer, then a vector is a column of numbers, and a covector is a row of numbers, with function application being the usual matrix product.)

In Windows, COM functions generally begin with the prefix `Co-` : `CoInitialize` , `CoCreateInstance` , `CoRegisterClassObject` . The `Co-` prefix is just a namespace disambiguator, just like how shell functions begin with `SH-` and native system services begin with `Zw-` (which doesn't stand for anything).

The first time I encountered these COM functions, it was in a meeting, when somebody said something like, “You need to call `CoCreateInstance` with...” I came from a world where the *co-* prefix represented mathematical duality, so I assumed that this mysterious function somehow created a dual object. You took this dual object, applied it to another object, and out came a number! I then spent some time trying to figure out what this dual object could look like, and how it could be useful.

But of course it wasn't any such thing. It was just “create an instance of an object.”

Today is the first day of the 2019 International Mathematical Olympiad.

[Raymond Chen](#)

**Follow**

