Actually, FlagsAttribute can't do more; that's why it's an attribute

devblogs.microsoft.com/oldnewthing/20090811-00

August 11, 2009

Raymond Chen

A few years ago, Abhinaba wondered why FlagsAttribute didn't also alter the way enumeration values are auto-assigned. Because attributes don't change the language. They are instructions to the runtime environment or (in rarer cases) to the compiler. An attribute can instruct the runtime environment to treat the function or class in a particular way. For example, you can use an attribute to tell the runtime environment that you want the program entry point to run in a single-threaded apartment, to tell the runtime environment how to look up your p/invoke function, or to tell the compiler to suppress a particular class of warnings. But changing how values for enumerations are assigned, well that actually changes the language. An attribute can't change the operator precedence tables. An attribute can't change the way overloaded functions are resolved. An attribute can't change the statement block tokens from curly braces to square braces. An attribute can't change the IL that gets generated. The code still compiles to the same IL; the attribute just controls the execution environment, such as how the JIT compiler chooses to lay out a structure in memory.

Attribute or not, enumerations follow the same rule for automatic assignment: An enumeration symbol receives the value one greater than the previous enumeration symbol.

Raymond Chen

Follow