One of my favorite error codes: Optional parameter missing

devblogs.microsoft.com/oldnewthing/20140129-00

January 29, 2014

Raymond Chen

The error *Optional parameter missing* sounds awfully paradoxical, doesn't it. I mean, if the parameter is optional, why are you complaining that it's missing? This KB article explains why, specifically, the part that says, "If a parameter is omitted, the calling program must…". For those who don't want to click through, here's the deal: Methods described in a type library can declare parameters as optional. Optional parameters must come at the end of the parameter list, of course. The catch is how you programmatically invoke a method that contains optional parameters. If you want to call a method that has optional parameters, you still have to pass the parameters. But you have to pass them a special way so that the method knows that they were omitted. (Specifically, you have to pass them as type `VT_ERROR`, code `DISP_E_PARAMNOTFOUND`.) In other words, an optional parameter is optional from the high-level language, but it isn't optional at the low-level interface. Rather the `optional` keyword instructs the scripting engine to deploy syntactic sugar and allow the script to omit the parameters. The scripting engine then must follow specific rules in providing values for the parameters that were omitted by the script author.

In the lingo of C# and C++, a parameter declared `optional` in the TLB is really a mandatory parameter whose default value is `DISP_E_PARAMNOTFOUND`.

Raymond Chen

Follow