

Consequences of the Explorer view model: If you create a view, then you succeeded, even if you'd rather fail

 devblogs.microsoft.com/oldnewthing/20081124-00

November 24, 2008



Raymond Chen

Commenter Anonymous asked [why navigating to a drive with no media displays a dialog instead of showing the error message in the view.](#)

This is an unfortunate consequence of Explorer's browser/view model. The shell browser binds to the `IShellFolder` and asks for the view by calling `IShellFolder::CreateViewWindow`. The view window calls `IShellFolder::EnumObjects` to figure out what to show in the view—and here is where the error dialog appears asking you to insert a disc into the drive.

The problem is that `IShellFolder::EnumObjects` has to return an enumerator or an error code. There is no return value that says “Um, yeah, could you display this text instead?” In a narrow sense, there's no way to return it since there is no way to return a string from `IShellFolder::EnumObjects`, but it's also not possible in a broader sense, since there is no rule that says only shell views can call `IShellFolder::EnumObjects`. Anybody can bind to a shell folder and enumerate its contents. And most of them don't have any place to display a text message instead of the enumerated objects. For example, the folder tree uses `IShellFolder::EnumObjects` to fill in children of a node. If you expand a node for an empty floppy drive, where is the “Sorry” message supposed to appear?

Now, you might say, “Well, make a special case for Explorer,” and maybe that's the right thing to do, but designing in a special case to a general interface just for one program tends to create resentment for others: “How come Explorer can do this but my program can't?”

[Raymond Chen](#)

Follow

