

Pidls and monikers do roughly the same thing, just backwards

devblogs.microsoft.com/oldnewthing/20060712-17

July 12, 2006



Raymond Chen

When operating with the Windows shell, you will almost certainly find yourself at some point working with a pointer to an item ID list, known also as a “pidl” (rhymes with “middle”). On the other hand, when working with OLE you may find yourself having to deal with monikers. In a sense, they both do the same thing. They let you refer to some sort of object inside a namespace (to which you can bind or otherwise operate on), they have a hierarchical structure, you can persist them, and so on. Why, then, did the Windows shell team invent pidls when monikers do the same thing? The fundamental difference between pidls and monikers is not in what you can do with them, but rather in their “bias”. A moniker is a COM object. This means that its primary existence is in the form of a COM interface (`IMoniker`), which means that the code behind the object is loaded and ready. You can convert it to a persistence format, but that’s not how monikers really spend their lives. If you have an array of monikers, all the code behind those objects has been loaded and initialized. A pidl, on the other hand, spends most of its life in its persistence format. Only when you bind to it does a live COM object come out. Consider, for example, the case where you enumerated the contents of a shell folder. This produces a pidl for each item in the folder, but producing and retaining that pidl doesn’t require that the code for each item be loaded and initialized. A folder with a thousand items produces a thousand little chunks of data, not a thousand COM objects. Consider, for example, a folder that contains a dozen Excel spreadsheets. If you enumerate the contents of the folder, you don’t want Excel to fire up and load each of those spreadsheets to give you a dozen live, running spreadsheet objects. At this stage, you are just talking **about** that spreadsheet file. It’s not until the user double-clicks on it or otherwise tries to activate that spreadsheet that you want Excel to start up.

Since the shell spends nearly all its time talking **about** things and comparatively rarely talks **to** them, an object that spends most of its time “dead” was more appropriate. You might say that the difference between pidls and monikers is a matter of life and death.

[Raymond Chen](#)

Follow

