

When I select multiple files in the File Open dialog, why does the last item come first?

devblogs.microsoft.com/oldnewthing/20190409-00

April 9, 2019



Raymond Chen

If you have a File Open dialog that supports multiple selections, and you select multiple files, you may notice that the file you selected last is listed first. This is annoying when the order of the files in the selection is significant. Why does the last item selected become the first item listed?

This is a case of a leaky abstraction.

The File.Open dialog shows the list of selected items in the order they are provided by the namespace extension's `IShellView::GetItemObject()` method:

```
IDataObject *pdto;  
shellView->GetItemObject(SVGIO_SELECTION, IID_PPV_ARGS(&pdto));
```

The system-provided filesystem namespace extension returns the selected items in the order they appear in the view, with the exception that the current focused item always comes first.

Okay, so why does the focused item come first?

In order to get a better context menu.

When you ask for the context menu of a group of files, the shell needs to decide which context menu handler will be asked to provide the verbs. One idea would be to ask each individual file for its context menu, and then combine all the context menus somehow. Maybe you show the intersection of all of the context menus. Maybe you show the union of all the context menus, and if the user picks an action that is not available on all items, it just ignores the items that can't do it.

But all of that would have been a lot of work, especially since matching up verbs on different context menu is a bit of guesswork. If two context menus have a verb called "Make Current", does that mean that they are equivalent? What if one of them is a verb on a database, and "Add Table" adds a table to the database; whereas the other one is a verb on a 3D model, and "Add Table" adds a piece of furniture to the model.

Rather than creating a large number of context menus and trying to merge them in a not-completely-nonsensical way, the shell just picks one of the items and says, “Okay, you are the one in control.” And the verbs that come from that file are the verbs for the entire selection. (Don’t forget, we had 4MB of memory to work with.)

Okay, we’re getting closer.

If you select a bunch of files, and then right-click the selection, the shell has to pick somebody to be the one that sets the context menu. And the algorithm it uses is “I’ll pick the item that has focus.” And the way the view conveys this is by setting the focus item as the first item.

Okay, so now we can start unwinding. The focus item is the first item in the selection. And the File Open dialog lists the items in its edit box in the order they appear in the selection. The result is that the focus item appears first in the edit box.

Sometime later (I don’t know exactly when), the flag `SVGIO_FLAG_VIEWORDER` was added to the `IShellView::GetItemObject` method, which tells the view, “Don’t put the focus item first. Just give me the items in the order they appear in the view.”

When this flag was added, nobody went back and retrofitted the File Open dialog to use it. It still asks for the selected items the old-fashioned way, and the old-fashioned way is to return the focus item first.

But wait, you’re not entirely helpless here. If you are so moved, you could [submit feedback](#) requesting that the File Open dialog use the `SVGIO_FLAG_VIEWORDER` flag so that the selected items show up in view order. Maybe if it gets enough votes, something will happen. Hey, it worked for the Persian calendar.

Raymond Chen

Follow

