

How do I control the order of the pages in property sheets from my shell extension?

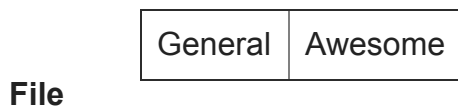
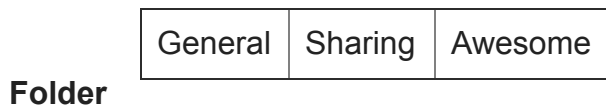
devblogs.microsoft.com/oldnewthing/20130815-00

August 15, 2013



Raymond Chen

A customer wanted to know whether a shell extension can control the order of the property sheet pages in a property sheet. The `IShellPropSheetExt` interface lets you add pages and replace pages, but nothing about rearranging them. Naturally a shell extension can control the relative order of its own pages (by changing in the order in which it calls `IShellPropSheetExt::AddPages`) but how can it affect the order of pages from other shell extensions? Imagine if that were possible. Every shell extension would set itself to be first! The customer was kind enough to explain what they were doing. “We were more concerned about consistency, because our tab appears in different positions depending on whether you are viewing a file or folder. Nothing critical. It just looked nicer if our extension always appeared in the same location.” Well, sure, it looks nicer *for you* if your extension always appears in the same relative position. But consider:



If you imposed a consistent position for your extension, it would have to go into position 2, but then that means that Sharing is no longer in the number 2 position when available. Maybe users like it when Sharing was the second page when available? Even if you manage to remain in the same position, it might not be in the same position due to changes in text length.

| | | | |
|---------|---------|---------|-------------------|
| General | Sharing | Awesome | Previous Versions |
|---------|---------|---------|-------------------|

Folder

| | | |
|---------|-------------------|---------|
| General | Previous Versions | Awesome |
|---------|-------------------|---------|

File

Sure, your awesome extension is always in third position, but since the length of the string *Sharing* is not the same as the length of the string *Previous Versions*, its position is not visually consistent. Now, sure, maybe Explorer could have a flag *ConsistentPosition* that a shell extension could specify to indicate that it wants a consistent position, and let Explorer figure out how to arrange the tabs in order to achieve that. In the second example above, you would get

| | | | |
|---------|---------|---------|-------------------|
| General | Awesome | Sharing | Previous Versions |
|---------|---------|---------|-------------------|

Folder

| | | |
|---------|---------|-------------------|
| General | Awesome | Previous Versions |
|---------|---------|-------------------|

File

but that's easy because you have only two cases to reconcile and because you have only one person who specified *ConsistentPosition*. Let's say that there are two shell extensions which specify *ConsistentPosition*. The *Awesome* extension applies to files whose extensions contain the letter A, and the *Brillant* extension applies to files whose extensions contain the letter B. You now have the following cases:

| | |
|---------|---------|
| General | Awesome |
|---------|---------|

.A

| | |
|---------|----------|
| General | Brillant |
|---------|----------|

.B

| | | | |
|---------|---------|---|----------|
| General | Awesome | ↔ | Brillant |
|---------|---------|---|----------|

.AB

Now there is no placement of property sheet pages such that *Awesome* and *Brilliant* can both have a consistent position. And I'm not even counting the cases where property sheet extensions hide their pages conditionally at runtime, so this sort of static analysis becomes impossible.

So no, you cannot force your property sheet page to appear at any particular position.

Raymond Chen

Follow

