If you're going to specify the LVS_SORTASCENDING or LVS_SORTDESCENDING style, you had better be telling the truth

September 26, 2024

A customer found that if they created a Win32 ListView control, and then used `LVM_SORTITEMS` to sort the items in descending order (presumably in response to a "reverse sort order" button somewhere), then keyboard searching stopped working. Why does sorting a ListView break keyboard searching?

Closer investigation uncovered that they had created the ListView with the `LVS_SORTASCENDING` style. When this style is enabled, the ListView control assumes that all the items are sorted ascending in alphabetical order by their label text. When you insert a new item, the ListView uses a binary search to find the insertion point for the new item. And when the user starts typing, it uses a binary search to find the item to select for the incremental search.

If you use the `LVM_SORTITEMS` or `LVM_SORTITEMSEX` message to change the order of the items, and the result doesn't match the ListView's declared sort order, then the ListView's binary search algorithm fails (because binary search assumes a sorted list), and incremental searching breaks down.

When the code changes the order of the items, the result needs to match the styles specified on the ListView: If the `LVS_SORTASCENDING` style is set, then the items must be sorted ascending. If the `LVS_SORTDESCENDING` style is set, then the items must be sorted descending. If neither style is set, then there is no requirement on the order of the items.

In this case, after sorting the items in descending order, the code definitely needs to remove the `LVS_SORTASCENDING` style, since the items are no longer sorted ascending. And it may as well add the `LVS_SORTDESCENDING` style so that the ListView can take advantage of the new sort order.[1]

[1] ListView cannot update the styles automatically because it doesn't know what sort criteria your custom sort function uses. I guess it could check whether the result of the sort is ascending or descending and auto-update the styles, but it doesn't do that. People who specify `LVS_SORTASCENDING` or `LVS_SORTDESCENDING` typically just set it and forget it, allowing the ListView to maintain the sorted order when new items are inserted. They don't try to override the style by doing a manual sort operation to mess up the ListView's carefully groomed item order.