

# What is the opposite of LVM\_SORTITEMS?

---

 [devblogs.microsoft.com/oldnewthing/20230518-00](https://devblogs.microsoft.com/oldnewthing/20230518-00)

May 18, 2023



Raymond Chen

A customer was using the `LVM_SORTITEMS` message to sort the items in a classic Win32 ListView control in response to the user checking a box called “Show sorted by date”.

That part worked great.

They ran into trouble when the user unchecked the box: What is the message for telling the ListView control to return to what it looked like before it was sorted?

There is no such message.

Sorting a ListView is a destructive operation. The items are sorted according to the new criteria, and the previous item order is lost. You’ll have to restore the order yourself. For example, if the ListView had been sorted by name before the user chose to sort by date, you can return to the previous order by re-sorting by name. On the other hand, if there was no particular order to the items, you’ll have to save the original order and reapply it when the box is unchecked.

If you think about it, there’s no way the ListView could confidently restore the previous order: Suppose that after sorting by date, the user creates a new item, and then unchecks the box. Even if the ListView had remembered the previous item order, it wouldn’t know where to put that new item. (Maybe the customer expected the “undo sort” to be a roll-back and also delete the newly-added item?)

If you want things back the way they were, you’ll have to remember where they were.

**Bonus chatter:** In Windows 2000, you used to be allowed to pass `nullptr` as the sort callback function for `LVM_SORTITEMS`, and it would default to sorting alphabetically by the first column. If you tried that in Windows XP, however, you got an “invalid parameter” error. This was a regression introduced by the [Windows security push](#) and some overzealous parameter validation. Surprisingly, nobody noticed this regression for several years, by which point the new stricter validation had become accepted as the new standard behavior. It was a bug with seniority.

