The GETDISPINFO notifications tell you what information they want

devblogs.microsoft.com/oldnewthing/20070308-00

March 8, 2007

Raymond Chen

The `XXN_GETDISPINFO` notifications used by the common controls are used when the control asks its parent to generate information that had been marked as delay-rendered, either explicitly via values such as `LPSTR_TEXTCALLBACK` or implicitly by being an owner-data control, for example. In fact the control is really just the middle man between the code that requested information about an item (via a message like `LVM_GETITEM`) and the code that generates it (your `LVN_GETDISPINFO` handler). In other words, the code flow goes like this: Somebody interested in retrieving data from a list view creates a `LVITEM` structure and initializes the `LVITEM.mask` and other fields as necessary, based on the mask. (For example, if the `LVIF_TEXT` flag is set, then `LVITEM.pszText` and `LVITEM.cchTextMax` must also be set to the buffer and its size.) it then sends a `LVM_GETITEM` message to the list view control. The list view control looks at the `LVITEM.mask` to see what information needs to be filled in. Some of the information the list view can provide on its own. Other parts of the information require help from the list view control's parent. For example, if the `LVITEM.mask` has the `LVIF_TEXT` flag set, and the item has its text set to `LPSTR_TEXTCALLBACK`, then the list view needs to consult its parent to get the text. The list view control sends the `LVN_GETDISPINFO` message to its parent, saying, "Hey, somebody is looking for information; please provide the information that is requested in the `LVITEM.mask` member." After the parent handles the message, the results are returned back to the original caller. There's a little bonus step that occurs just before the results are returned: If the parent set the `LVIF_DI_SETITEM` flag in the `LVITEM.mask`, then the returned values are also saved into the list view control as if you had sent a `LVM_SETITEM` message. For example, if you set the `LVIF_DI_SETITEM` flag in response to a request for `LVIF_TEXT`, then the text you return will be saved into the list view item, overwriting the previous value of `LPSTR_TEXTCALLBACK`. This is handy if you only want to compute the result once and let the list view cache the result from the on. Notice that throughout this process, the `LVITEM.mask` controls what information is being requested by the original caller of the list view as well as what is being requested by the list view of its parent. If you make the mistake of changing the value of `LVITEM.mask` (aside from setting the `LVIF_DI_SETITEM` flag, as noted in the "bonus step"), then you interfere with this game of "pass the buck". After the parent handles the message, the results are returned back to the original caller. But if you have modified the `LVITEM.mask`, then the results being returned back to the caller aren't the same as the ones the caller requested! For example, if the list view sees the `LVIF_TEXT` flag set, then it will copy the string provided by the parent back into the caller's buffer. But wait a second, if the parent is the one who set the `LVIF_TEXT` flag, that means that the original caller didn't ask for the text. There is no buffer to copy the results back into. The list view copies the string to an unintialized pointer, and all sorts of memory corruption occurs as a result.

Moral of the story: When responding to a `XXN_GETDISPINFO` notification, respect the `mask`. It's the bookkeeping that specifies what information you're being asked to provide (and therefore what information will be copied back to the original caller). If you change this bookkeeping, the original caller is in for a big surprise. It's like being the cook in a restaurant modifying the customer's order. "Oh, you didn't want the salad; let me give you the veal instead."

Raymond Chen

Follow