

# Displaying the dictionary, part 3: Using an owner-data listview

---

 [devblogs.microsoft.com/oldnewthing/20050615-30](http://devblogs.microsoft.com/oldnewthing/20050615-30)

June 15, 2005



Raymond Chen

Owner-data listviews let you take over data management from the listview. This is useful in our case since we have over twenty thousand dictionary entries, and creating even that many *blank* listview items takes an unacceptably long amount of time.

Let's convert our listview to an owner-data listview. Believe it or not, this is quite easy to do once we have the [text callback technique from last time](#). Make the following changes:

```

LRESULT RootWindow::OnCreate()
{
    m_hwndLV = CreateWindow(WC_LISTVIEW, NULL,
        WS_VISIBLE | WS_CHILD | WS_TABSTOP |
        LVS NOSORTHEADER | LVS_OWNERDATA |
        LVS_SINGLESEL | LVS_REPORT,
        0, 0, 0, 0,
        m_hwnd,
        (HMENU)IDC_LIST,
        g_hinst,
        NULL);

    ...
    // for (int i = 0; i < Length(); i++) {
    // const DictionaryEntry& de = Item(i);
    // LVITEM item;
    // item.mask = LVIF_TEXT;
    // item.iItem = i;
    // item.iSubItem = COL_TRAD;
    // item.pszText = const_cast<LPWSTR>(de.m_pszTrad);
    // item.iItem = ListView_InsertItem(m_hwndLV, &item);
    // if (item.iItem >= 0) {
    // item.iSubItem = COL_PINYIN;
    // item.pszText = const_cast<LPWSTR>(de.m_pszPinyin);
    // ListView_SetItem(m_hwndLV, &item);
    // item.iSubItem = COL_ENGLISH;
    // item.pszText = const_cast<LPWSTR>(de.m_pszEnglish);
    // ListView_SetItem(m_hwndLV, &item);
    // }
    // }
    return 0;
}

```

That's right, we made things better by deleting code. Isn't that satisfying?

Owner-data is like the text callback mechanism in the extreme: The listview doesn't record *any* information about the contents of your items. Whenever it needs something, it always asks. To create twenty thousand items, we just call `ListView_SetItemCount` and tell it that there are twenty thousand items. (There is also a `ListView_SetItemCountEx` macro which lets you pass flags, none of which are relevant here.)

In many owner-data cases, the data comes from an external source, in which case the `LVN_ODCACHEHINT` notification can be helpful. The listview sends this notification to say, "I'm going to be asking a lot of questions about items in this range. You might want to go work on them." Note that the listview might ask questions about items outside the range, too. The notification is just a hint that most of the questions are likely to be in that range. In our case, we have all the data ahead of time, so we have no need for the hint.

Notice that with this change to an owner-data listview, the program starts up almost instantly. Remember also the way we arranged the data in our string pool: All the strings for an item are adjacent, and strings for consecutive items follow one another. This means that all the data for one screenful of information resides in contiguous memory. Result: Better locality, fewer page faults. We'll see more benefits of the string pool later.

That's all for this month. Next month, we'll come back to filling in the second column of data: the simplified Chinese characters.

Raymond Chen

**Follow**

