

# Displaying the dictionary, part 1: Naive version

---

 [devblogs.microsoft.com/oldnewthing/20050613-16](http://devblogs.microsoft.com/oldnewthing/20050613-16)

June 13, 2005



Raymond Chen

We return briefly to the ongoing Chinese/English dictionary series and write some code to display all the definitions we had worked so hard to collect. (I figure you're anxious to see something on the screen, so I am going to handle the Traditional Chinese/Simplified Chinese issue later. For now, the "Simplified" column will be blank.)

Take the dictionary program we've been developing so far and paste it into our new scratch program. (Delete the `main` function, of course.) First, search/replace and change `m_hwndChild` to `m_hwndLV` since our child window is a listview, and it's just nicer to say what it is up front since we're going to be talking about it a lot. Next, make the following additional changes:

```

class RootWindow : public Window
{
public:
    virtual LPCTSTR ClassName() { return TEXT("Scratch"); }
    static RootWindow *Create();
protected:
    LRESULT HandleMessage(UINT uMsg, WPARAM wParam, LPARAM lParam);
    LRESULT OnCreate();
    const DictionaryEntry& Item(int i) { return m_dict.Item(i); }
    int Length() { return m_dict.Length(); }
private:
    enum {
        IDC_LIST = 1,
    };
    enum {
        COL_TRAD,
        COL_SIMP,
        COL_PINYIN,
        COL_ENGLISH,
    };
private:
    HWND m_hwndLV;
    Dictionary m_dict;
};
LRESULT RootWindow::OnCreate()
{
    m_hwndLV = CreateWindow(WC_LISTVIEW, NULL,
        WS_VISIBLE | WS_CHILD | WS_TABSTOP |
        LVS_NOSORTHEADER |
        LVS_SINGLESEL | LVS_REPORT,
        0, 0, 0, 0,
        m_hwnd,
        (HMENU)IDC_LIST,
        g_hinst,
        NULL);
    if (!m_hwndLV) return -1;
    ListView_SetExtendedListViewStyleEx(m_hwndLV,
        LVS_EX_FULLROWSELECT,
        LVS_EX_FULLROWSELECT);

    LVCOLUMN lvc;
    lvc.mask = LVCF_TEXT | LVCF_WIDTH;
    lvc.cx = 200;
    lvc.pszText = TEXT("Traditional");
    ListView_InsertColumn(m_hwndLV, COL_TRAD, &lvc);
    lvc.mask = LVCF_TEXT | LVCF_WIDTH;
    lvc.cx = 200;
    lvc.pszText = TEXT("Simplified");
    ListView_InsertColumn(m_hwndLV, COL_SIMP, &lvc);
    lvc.mask = LVCF_TEXT | LVCF_WIDTH;
    lvc.cx = 200;
    lvc.pszText = TEXT("PinYin");
    ListView_InsertColumn(m_hwndLV, COL_PINYIN, &lvc);
}

```

```

lvc.mask = LVCF_TEXT | LVCF_WIDTH;
lvc.cx = 800;
lvc.pszText = TEXT("English");
ListView_InsertColumn(m_hwndLV, COL_ENGLISH, &lvc);
ListView_SetItemCount(m_hwndLV, Length());
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;
}

```

After creating the listview control, we set it into full row select mode and create our columns. Before inserting the words into the dictionary, we use `ListView_SetItemCount` to tell the listview the number of items we're about to put into the listview. (This is optional; it allows the listview to pre-allocate some structures.) I'm not using an STL iterator because this code is going to be deleted soon. You'll find out why if you can't figured it out already.

Compile and run this program. Notice that it takes a ridiculously long time to start up. That's because our loop is inserting 20,000 dictionary entries into the listview, and that can't be fast.

Next time, we'll work on speeding that up.

Raymond Chen

**Follow**

