The dialog manager, part 1: Warm-ups

devblogs.microsoft.com/oldnewthing/20050329-00

March 29, 2005

Raymond Chen

I think a lot of confusion about the dialog manager stems from not really understanding how it works. It’s really not that bad. I’ll start by describing how dialog boxes are created over the next few articles, then move on to the dialog message loop, and wrap up with some topics regarding navigation. There will be nine parts in all.

The first major chunk of the dialog manager has to do with reading the dialog template and creating a dialog box based on it.

All of the CreateDialogXxx functions are just front-ends to the real work that happens in CreateDialogIndirectParam. Some of them are already visible in the macros: CreateDialog is just a wrapper around CreateDialogParam, with a parameter of zero. Similarly, CreateDialogIndirect is just a wrapper around CreateDialogIndirectParam with a zero parameter.

Here’s a slightly less trivial wrapper:

```
HWND WINAPI CreateDialogParam(HINSTANCE hinst,
    LPCTSTR pszTemplate, HWND hwndParent,
    DLGPROC lpDlgProc, LPARAM dwInitParam)
{
  HWND hdlg = NULL;
  HRSRC hrsrc = FindResource(hinst, pszTemplate,
                             RT_DIALOG);
  if (hrsrc) {
    HGLOBAL hglob = LoadResource(hinst, hrsrc);
    if (hglob) {
      LPVOID pTemplate = LockResource(hglob); // fixed 1pm
      if (pTemplate) {
        hdlg = CreateDialogIndirectParam(hinst,
                 pTemplate, hwndParent, lpDlgProc,
                 dwInitParam);
      }
      FreeResource(hglob);
    }
  }
  return hdlg;
}
```

All CreateDialogParam does is use the hinst/pszTemplate to locate the lpTemplate, then use that template in CreateDialogIndirectParam.

Okay, this was easy. Tomorrow, we’re going to create the dialog from the template.

Raymond Chen

Follow