The evolution of dialog templates – 32-bit Classic Templates

devblogs.microsoft.com/oldnewthing/20040621-00

June 21, 2004

Raymond Chen

Okay, last time we talked about the 16-bit classic DIALOG template. This time, we're going to talk about the 32-bit classic DIALOG template.

There really isn't much going on. Some 8-bit fields got expanded to 16-bit fields, some 16-bit fields got expanded to 32-bit fields, extended styles were added, and all strings got changed from ANSI to Unicode.

The template starts like this:

```
DWORD dwStyle;   // dialog style
DWORD dwExStyle; // extended dialog style
WORD  cItems;    // number of controls in this dialog
WORD  x;         // x-coordinate
WORD  y;         // y-coordinate
WORD  cx;        // width
WORD  cy;        // height
```

This is basically the same as the 16-bit dialog template, except that there's a new `dwExStyle` field, and the `cItems` went from a BYTE to a WORD. Consequently, the maximum number of controls per 32-bit dialog is 65535. That should be enough for a while.

After this header come a series of strings, just like in 16-bit dialog templates. But this time, the strings are Unicode. For example, if you wanted to store the string "Hello", you would write out the twelve bytes

```
48 00 65 00 6C 00 6C 00 6F 00 00 00 ; "Hello"
```

As with the 16-bit case, in the 32-bit dialog template, you can often specify an ordinal instead of a string. Here, it's done by writing the bytes FF 00 followed by the 16-bit ordinal (in little-endian format). For example, if you wanted to specify the ordinal 42, you would write out the four bytes

```
FF 00 2A 00        ; 00FF followed by WORD (little-endian)
```

The three strings are the same as last time:

- The menu name, which can be a string or an ordinal.
- The class, which must be a string (no ordinals allowed).
- The dialog title, which must be a string (no ordinals allowed).

If the DS_SETFONT style is set, then what follows next is a WORD indicating the point size and a string specifying the font name. Otherwise, there is no font information. Same as in the 16-bit dialog template.

So far, everything has been WORD-aligned.

After the header comes a series of dialog item templates. Each item template begins on a DWORD boundary. (Insert padding if necessary to achieve this.)

```
DWORD dwStyle;   // window style
DWORD dwExStyle; // window extended style
WORD  x;         // x-coordinate (DLUs)
WORD  y;         // y-coordinate (DLUs)
WORD  cx;        // width (DLUs)
WORD  cy;        // height (DLUs)
WORD  wID;       // control ID
```

As before, the dialog coordinates are recorded in dialog units (DLUs).

Next comes the class name, either as a null-terminated Unicode string or as an ordinal. The ordinal codes for the six "standard" window classes are the same as for 16-bit dialog templates:

- 0x0080 = "button"
- 0x0081 = "edit"
- 0x0082 = "static"
- 0x0083 = "listbox"
- 0x0084 = "scrollbar"
- 0x0085 = "combobox"

After the class name comes the control text, either as a null-terminated string or as an ordinal, following the same rules as for the 16-bit template. **Extra weirdness**: To specify an ordinal here, use FFFF instead of 00FF as the ordinal marker. I don't know why.

After the control text comes up to 65535 bytes of "extra data" in the form of a 16-bit count, followed by the actual data. If there is no "extra data", then use a count of zero.

And that's all there is. As with last time, I'll present an annotated dialog template.

```
0000  C4 20 C8 80 00 00 00 00-0B 00 24 00 2C 00 E6 00  . ........$.,...
0010  5E 00 00 00 00 00 52 00-65 00 70 00 6C 00 61 00  ^.....R.e.p.l.a.
0020  63 00 65 00 00 00 08 00-4D 00 53 00 20 00 53 00  c.e.....M.S. .S.
0030  68 00 65 00 6C 00 6C 00-20 00 44 00 6C 00 67 00  h.e.l.l. .D.l.g.
0040  00 00 00 00 00 00 02 50-00 00 00 00 04 00 09 00  .......P........
0050  30 00 08 00 FF FF FF FF-82 00 46 00 69 00 26 00  0.........F.i.&.
0060  6E 00 64 00 20 00 77 00-68 00 61 00 74 00 3A 00  n.d. .w.h.a.t.:.
0070  00 00 00 00 80 00 83 50-00 00 00 00 36 00 07 00  .......P....6...
0080  72 00 0C 00 80 04 FF FF-81 00 00 00 00 00 00 00  r...............
0090  00 00 02 50 00 00 00 00-04 00 1A 00 30 00 08 00  ...P........0...
00A0  FF FF FF FF 82 00 52 00-65 00 26 00 70 00 6C 00  ......R.e.&.p.l.
00B0  61 00 63 00 65 00 20 00-77 00 69 00 74 00 68 00  a.c.e. .w.i.t.h.
00C0  3A 00 00 00 00 00 00 00-80 00 83 50 00 00 00 00  :..........P....
00D0  36 00 18 00 72 00 0C 00-81 04 FF FF 81 00 00 00  6...r...........
00E0  00 00 00 00 03 00 03 50-00 00 00 00 05 00 2E 00  .......P........
00F0  68 00 0C 00 10 04 FF FF-80 00 4D 00 61 00 74 00  h.........M.a.t.
0100  63 00 68 00 20 00 26 00-77 00 68 00 6F 00 6C 00  c.h. .&.w.h.o.l.
0110  65 00 20 00 77 00 6F 00-72 00 64 00 20 00 6F 00  e. .w.o.r.d. .o.
0120  6E 00 6C 00 79 00 00 00-00 00 00 00 03 00 01 50  n.l.y..........P
0130  00 00 00 00 05 00 3E 00-3B 00 0C 00 11 04 FF FF  ......>.;.......
0140  80 00 4D 00 61 00 74 00-63 00 68 00 20 00 26 00  ..M.a.t.c.h. .&.
0150  63 00 61 00 73 00 65 00-00 00 00 00 01 00 03 50  c.a.s.e........P
0160  00 00 00 00 AE 00 04 00-32 00 0E 00 01 00 FF FF  ........2.......
0170  80 00 26 00 46 00 69 00-6E 00 64 00 20 00 4E 00  ..&.F.i.n.d. .N.
0180  65 00 78 00 74 00 00 00-00 00 00 00 00 00 01 50  e.x.t..........P
0190  00 00 00 00 AE 00 15 00-32 00 0E 00 00 04 FF FF  ........2.......
01A0  80 00 26 00 52 00 65 00-70 00 6C 00 61 00 63 00  ..&.R.e.p.l.a.c.
01B0  65 00 00 00 00 00 00 00-00 00 01 50 00 00 00 00  e..........P....
01C0  AE 00 26 00 32 00 0E 00-01 04 FF FF 80 00 52 00  ..&.2.........R.
01D0  65 00 70 00 6C 00 61 00-63 00 65 00 20 00 26 00  e.p.l.a.c.e. .&.
01E0  41 00 6C 00 6C 00 00 00-00 00 00 00 00 00 01 50  A.l.l..........P
01F0  00 00 00 00 AE 00 37 00-32 00 0E 00 02 00 FF FF  ......7.2.......
0200  80 00 43 00 61 00 6E 00-63 00 65 00 6C 00 00 00  ..C.a.n.c.e.l...
0210  00 00 00 00 00 00 01 50-00 00 00 00 AE 00 4B 00  .......P......K.
0220  32 00 0E 00 0E 04 FF FF-80 00 26 00 48 00 65 00  2.........&.H.e.
0230  6C 00 70 00 00 00 00 00                          l.p.....
```

As before, we start with the header.

```
0000  C4 20 C8 80  // dwStyle
0004  00 00 00 00  // dwExStyle
0008  0B 00        // cItems
000A  24 00 2C 00  // x, y
000E  E6 00 5E 00  // cx, cy
```

In other words, the header says

dwStyle	= 0x80C820C4	= WS_POPUP | WS_CAPTION | WS_SYSMENU | DS_CONTEXTHELP | DS_SETFONT | DS_MODALFRAME | DS_3DLOOK

dwExStyle	= 0x00000000	
cItems	= 0x0B	= 11
x	= 0x0024	= 36
y	= 0x002C	= 44
cx	= 0x00E6	= 230
cy	= 0x005E	= 94

After the header come the menu name, class name, and dialog title:

```
0012  00 00         // no menu
0014  00 00         // default dialog class
0016  52 00 65 00 70 00 6C 00 61 00 63 00
      65 00 00 00   // "Replace"
```

Again, since the DS_SETFONT bit is set in the style, the next section describes the font to be used by the dialog:

```
0026  08 00         // wSize = 8
0028  4D 00 53 00 20 00 53 00 68 00 65 00 6C 00
      6C 00 20 00 44 00 6C 00 67 00 00 00
                    // "MS Shell Dlg"
```

This dialog box uses 8pt “MS Shell Dlg” as its dialog font.

Next come the eleven dialog item templates. Not remember that each template must be DWORD-aligned, so we need some padding here to get up to a four-byte boundary.

```
0042 00 00          // Padding for alignment
```

Now that we are once again DWORD-aligned, we can read the first dialog item template.

```
0044  00 00 02 50   // dwStyle
0048  00 00 00 00   // dwExStyle
004C  04 00 09 00   // x, y
0050  30 00 08 00   // cx, cy
0054  FF FF         // wID
0056  FF FF 82 00   // "static"
005A  46 00 69 00 26 00
0060  6E 00 64 00 20 00 77 00-68 00 61 00 74 00 3A 00
0070  00 00         // "Fi&nd what:"
0072  00 00         // no extra data
```

Notice here that the “static” class was encoded as an ordinal. The template for this item is therefore

dwStyle	= 0x50020000	= WS_CHILD | WS_VISIBLE | WS_GROUP | SS_LEFT
dwExStyle	= 0x00000000	
x	= 0x0004	= 4
y	= 0x0009	= 9
cx	= 0x0030	= 48
cy	= 0x0008	= 8
wID	= 0xFFFF	= -1
szClass	= ordinal 0x0082	= “static”
szText	= “Fi&nd what:”	

The other controls are similarly unexciting.

```
// Second control
0074  80 00 83 50   // dwStyle
0078  00 00 00 00   // dwExStyle
007C  36 00 07 00   // x, y
0080  72 00 0C 00   // cx, cy
0084  80 04         // wID
0086  FF FF 81 00   // "edit"
008A  00 00         // ""
008C  00 00         // no extra data
008E  00 00         // padding to achieve DWORD alignment
// Third control
0090  00 00 02 50   // dwStyle
0094  00 00 00 00   // dwExStyle
0098  04 00 1A 00   // x, y
009C  30 00 08 00   // cx, cy
00A0  FF FF         // wID
00A2  FF FF 82 00   // "static"
00A6  52 00 65 00 26 00 70 00 6C 00
00B0  61 00 63 00 65 00 20 00 77 00 69 00 74 00 68 00
00C0  3A 00 00 00   // "Re&place with:"
00C4  00 00         // no extra data
00C6  00 00         // padding to achieve DWORD alignment
// Fourth control
00C8  80 00 83 50   // dwStyle
00CC  00 00 00 00   // dwExStyle
00D0  36 00 18 00   // x, y
00D4  72 00 0C 00   // cx, cy
00D8  81 04         // wID
00DA  FF FF 81 00   // "edit"
00DE  00 00         // ""
00E0  00 00         // no extra data
00E2  00 00         // padding to achieve DWORD alignment
// Fifth control
00E4  03 00 03 50   // dwStyle
00E8  00 00 00 00   // dwExStyle
00EC  05 00 2E 00   // x, y
00F0  68 00 0C 00   // cx, cy
00F4  10 04         // wID
00F6  FF FF 80 00   // "button"
00FA  4D 00 61 00 74 00
0100  63 00 68 00 20 00 26 00 77 00 68 00 6F 00 6C 00
0110  65 00 20 00 77 00 6F 00 72 00 64 00 20 00 6F 00
0120  6E 00 6C 00 79 00 00 00
                    // "Match &whole word only"
0128  00 00         // no extra data
012A  00 00         // padding to achieve DWORD alignment
// Sixth control
012C  03 00 01 50   // dwStyle
0130  00 00 00 00   // dwExStyle
0134  05 00 3E 00   // x, y
0138  3B 00 0C 00   // cx, cy
013C  11 04         // wID
```

```
013E  FF FF 80 00   // "button"
0142  4D 00 61 00 74 00 63 00 68 00 20 00 26 00
0150  63 00 61 00 73 00 65 00 00 00
                    // "Match &case"
015A  00 00         // no extra data
// Seventh control
015C  01 00 03 50   // dwStyle
0160  00 00 00 00   // dwExStyle
0164  AE 00 04 00   // x, y
0168  32 00 0E 00   // cx, cy
016C  01 00         // wID
016E  FF FF 80 00   // "button"
0172  26 00 46 00 69 00 6E 00 64 00 20 00 4E 00
0180  65 00 78 00 74 00 00 00
                    // "&Find Next"
0188  00 00         // no extra data
018A  00 00         // padding to achieve DWORD alignment
// Eighth control
018C  00 00 01 50   // dwStyle
0190  00 00 00 00   // dwExStyle
0194  AE 00 15 00   // x, y
0198  32 00 0E 00   // cx, cy
019C  00 04         // wID
019E  FF FF 80 00   // "button"
01A2  26 00 52 00 65 00-70 00 6C 00 61 00 63 00
01B0  65 00 00 00   // "&Replace"
01B4  00 00         // no extra data
01B6  00 00         // padding to achieve DWORD alignment
// Ninth control
01B8  00 00 01 50   // dwStyle
01BC  00 00 00 00   // dwExStyle
01C0  AE 00 26 00   // x, y
01C4  32 00 0E 00   // cx, cy
01C8  01 04         // wID
01CA  FF FF 80 00   // "button"
01CE  52 00
01D0  65 00 70 00 6C 00 61 00 63 00 65 00 20 00 26 00
01E0  41 00 6C 00 6C 00 00 00
                    // "Replace &All"
01E8  00 00         // no extra data
01EA  00 00         // padding to achieve DWORD alignment
// Tenth control
01EC  00 00 01 50   // dwStyle
01F0  00 00 00 00   // dwExStyle
01F4  AE 00 37 00   // x, y
01F8  32 00 0E 00   // cx, cy
01FC  02 00         // wID
01FE  FF FF 80 00   // "button"
0202  43 00 61 00 6E 00 63 00 65 00 6C 00 00 00
                    // "Cancel"
0210  00 00         // no extra data
0212  00 00         // padding to achieve DWORD alignment
```

```
// Eleventh control
0214  00 00 01 50   // dwStyle
0218  00 00 00 00   // dwExStyle
021C  AE 00 4B 00   // x, y
0220  32 00 0E 00   // cx, cy
0224  0E 04         // wID
0226  FF FF 80 00   // "button"
022A  26 00 48 00 65 00 6C 00 70 00 00 00
                    // "&Help"
0236  00 00         // no extra data
```

Whew. Tedious and entirely unexciting. Here's the original resource compiler source code that we reverse-engineered:

```
DIALOG 36, 44, 230, 94
STYLE WS_POPUP | WS_CAPTION | WS_SYSMENU | DS_MODALFRAME | DS_3DLOOK | NOT WS_VISIBLE
CAPTION "Replace"
FONT 8, "MS Shell Dlg"
BEGIN
    CONTROL "Fi&nd What:", -1, "static", WS_GROUP | SS_LEFT,
            4, 9, 48, 8
    CONTROL "", 0x0480, "edit",
            WS_BORDER | WS_GROUP | WS_TABSTOP | ES_AUTOHSCROLL,
            54, 7, 114, 12
    CONTROL "Re&place with:", -1, "static", WS_GROUP | SS_LEFT,
            4, 26, 48, 8
    CONTROL "", 0x0481, "edit",
            WS_BORDER | WS_GROUP | WS_TABSTOP | ES_AUTOHSCROLL,
            54, 24, 114, 12
    CONTROL "Match &whole word only", 0x0410, "button",
            WS_GROUP | WS_TABSTOP | BS_AUTOCHECKBOX,
            5, 46, 104, 12
    CONTROL "Match &case", 0x0411, "button",
            WS_TABSTOP | BS_AUTOCHECKBOX,
            5, 62, 59, 12
    CONTROL "&Find Next", IDOK, "button",
            WS_GROUP | WS_TABSTOP | BS_DEFPUSHBUTTON,
            174, 4, 50, 14
    CONTROL "&Replace", 0x0400, "button",
            WS_TABSTOP | BS_PUSHBUTTON,
            174, 21, 50, 14
    CONTROL "Replace &All", 0x0401, "button",
            WS_TABSTOP | BS_PUSHBUTTON,
            174, 38, 50, 14
    CONTROL "Cancel", IDCANCEL, "button",
            WS_TABSTOP | BS_PUSHBUTTON,
            174, 55, 50, 14
    CONTROL "Cancel", 0x040E, "button",
            WS_TABSTOP | BS_PUSHBUTTON,
            174, 75, 50, 14
END
```

As before, we didn't explicitly say "DS_SETFONT" in the dialog's STYLE directive since that is implied by the "FONT" directive, and we took advantage of the fact that WS_VISIBLE is on by default.

And you probably recognize this dialog from yesterday. It's the replace dialog from findtext.dlg. (Though it's not exactly the same since the findtext.dlg template uses some shorthand directives like DEFPUSHBUTTON instead of manually writing out the details of the button control as a CONTROL.)

Next time: The 16-bit extended dialog template, also known as DIALOGEX.

Raymond Chen

Follow