

The evolution of dialog templates – 16-bit Extended Templates

 devblogs.microsoft.com/oldnewthing/20040622-00

June 22, 2004



Raymond Chen

The next step in the evolution of dialog templates is the Extended Dialog or DIALOGEX. First, let's look at the 16-bit version.

The 16-bit extended dialog template is purely historical. The only operating systems to support it were the Windows 95/98/Me series. It is interesting only as a missing link in the evolution towards the 32-bit extended dialog template.

The basic rules are the same as for the non-extended template. The extended dialog template starts off with a different header:

```
WORD  wDlgVer;      // version number - always 1
WORD  wSignature;  // always 0xFFFF
DWORD dwHelpID;    // help ID
DWORD dwExStyle;   // window extended style
DWORD dwStyle;     // dialog style
BYTE  cItems;      // number of controls in this dialog
WORD  x;           // x-coordinate
WORD  y;           // y-coordinate
WORD  cx;          // width
WORD  cy;          // height
```

The first two fields specify a version number (so far, only version 1 extended dialogs have been defined), and a signature value 0xFFFF which allows this template to be distinguished from a non-extended dialog template.

Next come two new fields. The help identifier is an arbitrary 32-bit value that you can retrieve from the dialog later with the [GetWindowContextHelpId](#) function. The extended dialog style you've seen before.

As before, after the header come the strings. First comes the menu, then the class, then dialog title, all encoded the same way as with the non-extended template.

If the DS_SETFONT style is set, then a custom font exists in the template. The format of the font information is slightly different for extended templates. In classic templates, all you get is a WORD of point size and a font name. But in the extended template, the font information is a little richer:

```
WORD wPoint;          // point size
WORD wWeight;        // font weight
BYTE bItalic;        // 1 if italic, 0 if not
BYTE bCharSet;       // character set (see CreateFont)
CHAR szFontName[];   // variable-length
```

New fields are the weight, character set, and whether the font is italic.

After the header come the dialog item templates, each of which looks like this:

```
DWORD dwHelpID;      // help identifier
DWORD dwExStyle;     // window extended style
DWORD dwStyle;       // window style
WORD x;              // x-coordinate (DLUs)
WORD y;              // y-coordinate (DLUs)
WORD cx;             // width (DLUs)
WORD cy;             // height (DLUs)
DWORD wID;           // control ID
CHAR szClassName[]; // variable-length (possibly ordinal)
CHAR szText[];       // variable-length (possibly ordinal)
WORD cbExtra;        // amount of extra data
BYTE rgbExtra[cbExtra]; // extra data follows (usually none)
```

This takes the classic item template and adds the following:

- New dwHelpID and dwExStyle fields.
- dwStyle field moved.
- control ID expanded to DWORD.
- cbExtra expanded to WORD.

Not that expanding the control ID to a 32-bit value accomplishes much in 16-bit Windows, but it's there nonetheless.

And that's all.

Now the obligatory annotated hex dump.

```

0000 01 00 FF FF 00 00 00 00-00 00 00 00 C4 00 C8 80 .....
0010 0B 24 00 2C 00 E6 00 5E-00 00 00 52 65 70 6C 61 .$. ,...^...Repla
0020 63 65 00 08 00 90 01 00-00 4D 53 20 53 68 65 6C ce.....MS Shel
0030 20 44 6C 67 00 00 00 00-00 00 00 00 00 00 00 02 Dlg.....
0040 50 04 00 09 00 30 00 08-00 FF FF FF FF 82 46 69 P....0.....Fi
0050 26 6E 64 20 77 68 61 74-3A 00 00 00 00 00 00 00 &nd what:.....
0060 00 00 00 00 80 00 83 50-36 00 07 00 72 00 0C 00 .....P6...r...
0070 80 04 00 00 81 00 00 00-00 00 00 00 00 00 00 00 .....
0080 00 00 02 50 04 00 1A 00-30 00 08 00 FF FF FF FF ...P....0.....
0090 82 52 65 26 70 6C 61 63-65 20 77 69 74 68 3A 00 .Re&place with:.
00A0 00 00 00 00 00 00 00 00-00 00 80 00 83 50 36 00 .....P6.
00B0 18 00 72 00 0C 00 81 04-00 00 81 00 00 00 00 00 ..r.....
00C0 00 00 00 00 00 00 03 00-03 50 05 00 2E 00 68 00 .....P....h.
00D0 0C 00 10 04 00 00 80 4D-61 74 63 68 20 26 77 68 .....Match &wh
00E0 6F 6C 65 20 77 6F 72 64-20 6F 6E 6C 79 00 00 00 ole word only...
00F0 00 00 00 00 00 00 00 00-03 00 01 50 05 00 3E 00 .....P..>.
0100 3B 00 0C 00 11 04 00 00-80 4D 61 74 63 68 20 26 ;.....Match &
0110 63 61 73 65 00 00 00 00-00 00 00 00 00 00 00 01 case.....
0120 00 03 50 AE 00 04 00 32-00 0E 00 01 00 00 00 80 ..P....2.....
0130 26 46 69 6E 64 20 4E 65-78 74 00 00 00 00 00 00 &Find Next.....
0140 00 00 00 00 00 00 00 03-50 AE 00 15 00 32 00 0E .....P....2..
0150 00 00 04 00 00 80 26 52-65 70 6C 61 63 65 00 00 .....&Replace..
0160 00 00 00 00 00 00 00 00-00 00 00 03 50 AE 00 26 .....P..&
0170 00 32 00 0E 00 01 04 00-00 80 52 65 70 6C 61 63 .2.....Replac
0180 65 20 26 41 6C 6C 00 00-00 00 00 00 00 00 00 00 e &All.....
0190 00 00 00 03 50 AE 00 37-00 32 00 0E 00 02 00 00 ....P..7.2.....
01A0 00 80 43 61 6E 63 65 6C-00 00 00 00 00 00 00 ..Cancel.....
01B0 00 00 00 00 00 03 50 AE-00 4B 00 32 00 0E 00 0E .....P..K.2....
01C0 04 00 00 80 26 48 65 6C-70 00 00 00 .....&Help...

```

Once again, we start with the header.

```

0000 01 00 // wVersion
0002 FF FF // wSignature
0004 00 00 00 00 // dwHelpID
0008 00 00 00 00 // dwExStyle
000C C4 00 C8 80 // dwStyle
0010 0B // cItems
0011 24 00 2C 00 // x, y
0015 E6 00 5E 00 // cx, cy

```

The header breaks down as follows:

wVersion	= 0x0001	= 1
<hr/>		
wSignature	= 0xFFFF	
<hr/>		
dwHelpID	=	= 0
	0x00000000	
<hr/>		

dwExStyle	=		= 0
		0x00000000	
dwStyle	=		= WS_POPUP WS_CAPTION WS_SYSMENU
		0x80C800C4	DS_SETFONT DS_MODALFRAME DS_3DLOOK
cltems	=	0x0B	= 11
x	=	0x0024	= 36
y	=	0x002C	= 44
cx	=	0x00E6	= 230
cy	=	0x005E	= 94

Next come the menu name, class name, and dialog title:

```
0019 00 // no menu
001A 00 // default dialog class
001B 52 65 70 6C 61 63 65 00 // "Replace"
```

Same as the 16-bit classic template.

The presence of DS_SETFONT means that there's font information ahead. This looks slightly different.

```
0023 08 00 // wSize = 8
0025 90 01 // wWeight = 0x02BC = 700 = FW_NORMAL
0027 00 // Italic
0028 00 // Character set = 0x00 = ANSI_CHARSET
0029 4D 53 20 53 68 65 6C 20 44 6C 67 00
// "MS Shell Dlg"
```

Now follow the extended dialog item templates. This should all be old hat by now, so I won't go into detail.

```

// First control
0035 00 00 00 00 // dwHelpID
0039 00 00 00 00 // dwExStyle
003D 00 00 02 50 // dwStyle
0041 04 00 09 00 // x, y
0045 30 00 08 00 // cx, cy
0049 FF FF FF FF // dwID
004D 82 // szClass = ordinal 0x82 = "static"
004E 46 69 26 6E 64 20 77 68 61 74 3A 00
// "Fi&nd what:"
005A 00 00 // no extra data
// Second control
005C 00 00 00 00 // dwHelpID
0060 00 00 00 00 // dwExStyle
0064 80 00 83 50 // dwStyle
0068 36 00 07 00 // x, y
006C 72 00 0C 00 // cx, cy
0070 80 04 00 00 // dwID
0074 81 // "edit"
0075 00 // ""
0076 00 00 // no extra data
// Third control
0078 00 00 00 00 // dwHelpID
007C 00 00 00 00 // dwExStyle
0080 00 00 02 50 // dwStyle
0084 04 00 1A 00 // x, y
0088 30 00 08 00 // cx, cy
008C FF FF FF FF // dwID
0090 82 // "static"
0091 52 65 26 70 6C 61 63 65 20 77 69 74 68 3A 00
// "Re&place with:"
00A0 00 00 // no extra data
// Fourth control
00A2 00 00 00 00 // dwHelpID
00A6 00 00 00 00 // dwExStyle
00AA 80 00 83 50 // dwStyle
00AE 36 00 18 00 // x, y
00B2 72 00 0C 00 // cx, cy
00B6 81 04 00 00 // dwID
00BA 81 // "edit"
00BB 00 // ""
00BC 00 00 // no extra data
// Fifth control
00BE 00 00 00 00 // dwHelpID
00C2 00 00 00 00 // dwExStyle
00C6 03 00 03 50 // dwStyle
00CA 05 00 2E 00 // x, y
00CE 68 00 0C 00 // cx, cy
00D2 10 04 00 00 // dwID
00D6 80 // "button"
00D7 4D 61 74 63 68 20 26 77 68 6F 6C 65 20 77
6F 72 64 20 6F 6E 6C 79 00

```

```

// "Match &whole word only"
00EE 00 00 // no extra data
// Sixth control
00F0 00 00 00 00 // dwHelpID
00F4 00 00 00 00 // dwExStyle
00F8 03 00 01 50 // dwStyle
00FC 05 00 3E 00 // x, y
0100 3B 00 0C 00 // cx, cy
0104 11 04 00 00 // dwID
0108 80 // "button"
0109 4D 61 74 63 68 20 26 63 61 73 65 00
// "Match &case"
0115 00 00 // no extra data
// Seventh control
0117 00 00 00 00 // dwHelpID
011B 00 00 00 00 // dwExStyle
011F 01 00 03 50 // dwStyle
0123 AE 00 04 00 // x, y
0127 32 00 0E 00 // cx, cy
012B 01 00 00 00 // dwID
012F 80 // "button"
0130 26 46 69 6E 64 20 4E 65 78 74 00
// "&Find Next"
013B 00 00 // no extra data
// Eighth control
013D 00 00 00 00 // dwHelpID
0141 00 00 00 00 // dwExStyle
0145 00 00 03 50 // dwStyle
0149 AE 00 15 00 // x, y
014D 32 00 0E 00 // cx, cy
0151 00 04 00 00 // dwID
0155 80 // "button"
0156 26 52 65 70 6C 61 63 65 00
// "&Replace"
015F 00 00 // no extra data
// Ninth control
0161 00 00 00 00 // dwHelpID
0165 00 00 00 00 // dwExStyle
0169 00 00 03 50 // dwStyle
016D AE 00 26 00 // x, y
0171 32 00 0E 00 // cx, cy
0175 01 04 00 00 // dwID
0179 80 // "button"
017A 52 65 70 6C 61 63 65 20 26 41 6C 6C 00
// "Replace &All"
0187 00 00 // no extra data
// Tenth control
0189 00 00 00 00 // dwHelpID
018D 00 00 00 00 // dwExStyle
0191 00 00 03 50 // dwStyle
0195 AE 00 37 00 // x, y
0199 32 00 0E 00 // cx, cy

```

```
019D 02 00 00 00 // dwID
01A1 80 // "button"
01A2 43 61 6E 63 65 6C 00
// "Cancel"
01A9 00 00 // no extra data
// Eleventh control
01AB 00 00 00 00 // dwHelpID
01AF 00 00 00 00 // dwExStyle
01B3 00 00 03 50 // dwStyle
01B7 AE 00 4B 00 // x, y
01BB 32 00 0E 00 // cx, cy
01BF 0E 04 00 00 // dwID
01C3 80 // "button"
01C4 26 48 65 6C 70 00
// "&Help"
01CA 00 00 // no extra data
```

The original dialog template is the one you've seen twice already, with only one change: The DIALOG keyword has been changed to DIALOGEX.

```
DIALOGEX 36, 44, 230, 94
```

```
...
```

Tomorrow, we reach the modern era with the 32-bit DIALOGEX template.

Raymond Chen

Follow

