

# The evolution of version resources – 32-bit version resources

 [devblogs.microsoft.com/oldnewthing/20061221-02](http://devblogs.microsoft.com/oldnewthing/20061221-02)

December 21, 2006



Raymond Chen

Last time we looked at the format of 16-bit version resources. The 32-bit version is nearly identical, except that everything is now in Unicode. Each node is stored in the following structure (in pseudo-C):

```
struct VERSIONNODE {
    WORD  cbNode;
    WORD  cbData;
    WORD  wType;
    WCHAR wszName[];
    BYTE  rgbPadding1[]; // DWORD alignment
    union {
        BYTE  rgbData[cbData];
        WCHAR wszValue[cbData / sizeof(WCHAR)];
    };
    BYTE  rgbPadding2[]; // DWORD alignment
    VERSIONNODE rgvnChildren[];
};
```

In words, each version node begins with a 16-bit value describing the size of the nodes in bytes (including its children), followed by a 16-bit value that specifies how much data (either binary or text) are associated with the node. The `wType` is zero if the node data is binary or one if the node data is a string. Other values of `wType` are reserved for future use. The rest is the same as before: A null-terminated key name, padding for `DWORD` alignment, the node data, more padding, and then the child nodes.

The only change beyond the conversion to Unicode is the introduction of the `wType` field. This field is necessary so that the system knows whether to convert the node data from Unicode to ANSI when somebody calls the `VerQueryStringA`. If it's binary data, then no conversion is done; if it's string data, then the string is converted.

We illustrate the 32-bit resource format by looking at the resources for the 32-bit `shell32.dll`.

```

0000 98 03 34 00 00 00 56 00-53 00 5F 00 56 00 45 00 ..4...V.S_.V.E.
0010 52 00 53 00 49 00 4F 00-4E 00 5F 00 49 00 4E 00 R.S.I.O.N_.I.N.
0020 46 00 4F 00 00 00 00 00-BD 04 EF FE 00 00 01 00 F.O.....
0030 00 00 06 00 35 0B 54 0B-00 00 06 00 35 0B 54 0b ....5.T....5.T.
0040 3F 00 00 00 00 00 00 00-04 00 04 00 02 00 00 00 ?.....
0050 00 00 00 00 00 00 00 00-00 00 00 00 F6 02 00 00 .....
0060 01 00 53 00 74 00 72 00-69 00 6E 00 67 00 46 00 ..S.t.r.i.n.g.F.
0070 69 00 6C 00 65 00 49 00-6E 00 66 00 6F 00 00 00 i.l.l.e.I.n.f.o...
0080 D2 02 00 00 01 00 30 00-34 00 30 00 39 00 30 00 .....0.4.0.9.0.
0090 34 00 42 00 30 00 00 00-4C 00 2C 00 01 00 43 00 4.B.0...L....C.
00A0 6F 00 6D 00 70 00 61 00-6E 00 79 00 4E 00 61 00 o.m.p.a.n.y.N.a.
00B0 6D 00 65 00 00 00 00 00-4D 00 69 00 63 00 72 00 m.e....M.i.c.r.
00C0 6F 00 73 00 6F 00 66 00-74 00 20 00 43 00 6F 00 o.s.o.f.t. .C.o.
00D0 72 00 70 00 6F 00 72 00-61 00 74 00 69 00 6F 00 r.p.o.r.a.t.i.o.
00E0 6E 00 00 00 5A 00 32 00-01 00 46 00 69 00 6C 00 n...Z....F.i.l.
00F0 65 00 44 00 65 00 73 00-63 00 72 00 69 00 70 00 e.D.e.s.c.r.i.p.
0100 74 00 69 00 6F 00 6E 00-00 00 00 00 57 00 69 00 t.i.o.n....W.i.
0110 6E 00 64 00 6F 00 77 00-73 00 20 00 53 00 68 00 n.d.o.w.s. .S.h.
0120 65 00 6C 00 6C 00 20 00-43 00 6F 00 6D 00 6D 00 e.l.l. .C.o.m.m.
0130 6F 00 6E 00 20 00 44 00-6C 00 6C 00 00 00 00 00 o.n. .D.l.l....
0140 74 00 54 00 01 00 46 00-69 00 6C 00 65 00 56 00 t.*...F.i.l.l.e.V.
0150 65 00 72 00 73 00 69 00-6F 00 6E 00 00 00 00 00 e.r.s.i.o.n....
0160 36 00 2E 00 30 00 30 00-2E 00 32 00 39 00 30 00 6...0.0...2.9.0.
0170 30 00 2E 00 32 00 38 00-36 00 39 00 20 00 28 00 0...2.8.6.9. .(
0180 78 00 70 00 73 00 70 00-5F 00 73 00 70 00 32 00 x.p.s.p._s.p.2.
0190 5F 00 67 00 64 00 72 00-2E 00 30 00 36 00 30 00 _g.d.r...0.6.0.
01A0 33 00 31 00 36 00 2D 00-31 00 35 00 31 00 32 00 3.1.6.-.1.5.1.2.
01B0 29 00 00 00 30 00 10 00-01 00 49 00 6E 00 74 00 )...0....I.n.t.
01C0 65 00 72 00 6E 00 61 00-6C 00 4E 00 61 00 6D 00 e.r.n.a.l.N.a.m.
01D0 65 00 00 00 53 00 48 00-45 00 4C 00 4C 00 33 00 e...S.H.E.L.L.3.
01E0 32 00 00 00 80 00 5C 00-01 00 4C 00 65 00 67 00 2.....L.e.g.
01F0 61 00 6C 00 43 00 6F 00-70 00 79 00 72 00 69 00 a.l.C.o.p.y.r.i.
0200 67 00 68 00 74 00 00 00-A9 00 20 00 4D 00 69 00 g.h.t.... .M.i.
0210 63 00 72 00 6F 00 73 00-6F 00 66 00 74 00 20 00 c.r.o.s.o.f.t. .
0220 43 00 6F 00 72 00 70 00-6F 00 72 00 61 00 74 00 C.o.r.p.o.r.a.t.
0230 69 00 6F 00 6E 00 2E 00-20 00 41 00 6C 00 6C 00 i.o.n... .A.l.l.
0240 20 00 72 00 69 00 67 00-68 00 74 00 73 00 20 00 .r.i.g.h.t.s. .
0250 72 00 65 00 73 00 65 00-72 00 76 00 65 00 64 00 r.e.s.e.r.v.e.d.
0260 2E 00 00 00 40 00 18 00-01 00 4F 00 72 00 69 00 ....@....O.r.i.
0270 67 00 69 00 6E 00 61 00-6C 00 46 00 69 00 6C 00 g.i.n.a.l.F.i.l.
0280 65 00 6E 00 61 00 6D 00-65 00 00 00 53 00 48 00 e.n.a.m.e...S.H.
0290 45 00 4C 00 4C 00 33 00-32 00 2E 00 44 00 4C 00 E.L.L.3.2...D.L.
02A0 4C 00 00 00 6A 00 25 00-01 00 50 00 72 00 6F 00 L...j.%...P.ro.
02B0 64 00 75 00 63 00 74 00-4E 00 61 00 6D 00 65 00 d.u.c.t.N.a.m.e.
02C0 00 00 00 00 4D 00 69 00-63 00 72 00 6F 00 73 00 ....M.i.c.r.o.s.
02D0 6F 00 66 00 74 00 AE 00-20 00 57 00 69 00 6E 00 o.f.t... .W.i.n.
02E0 64 00 6F 00 77 00 73 00-AE 00 20 00 4F 00 70 00 d.o.w.s... .O.p.
02F0 65 00 72 00 61 00 74 00-69 00 6E 00 67 00 20 00 e.r.a.t.i.n.g. .
0300 53 00 79 00 73 00 74 00-65 00 6D 00 00 00 00 00 S.y.s.t.e.m....
0310 42 00 1E 00 01 00 50 00-72 00 6F 00 64 00 75 00 B....P.r.o.d.u.
0320 63 00 74 00 56 00 65 00-72 00 73 00 69 00 6F 00 c.t.V.e.r.s.i.o.
0330 6E 00 00 00 36 00 2E 00-30 00 30 00 2E 00 32 00 n...6...0.0...2.

```

```

0340 39 00 30 00 30 00 2E 00-32 00 38 00 36 00 39 00 9.0.0...2.8.6.9.
0350 00 00 00 00 44 00 00 00-01 00 56 00 61 00 72 00 ....D....V.a.r.
0360 46 00 69 00 6C 00 65 00-49 00 6E 00 66 00 6F 00 F.i.l.e.I.n.f.o.
0370 00 00 00 00 24 00 04 00-00 00 54 00 72 00 61 00 ....$.....T.r.a.
0380 6E 00 73 00 6C 00 61 00-74 00 69 00 6F 00 6E 00 n.s.l.a.t.i.o.n.
0390 00 00 00 00 09 04 B0 04 .....

```

As always, we start with the root node.

```

0000 98 03          // cbNode (node ends at 0x0000 + 0x0398 = 0x0398)
0002 34 00          // cbData = sizeof(VS_FIXEDFILEINFO)
0004 00 00          // wType = 0 (binary data)
0006 56 00 53 00 5F 00 56 00 45 00 52 00 53 00 49 00
      4F 00 4E 00 5F 00 49 00 4E 00 46 00 4F 00 00 00
      // L"VS_VERSION_INFO" + null terminator
0026 00 00          // padding to restore alignment
0028 BD 04 EF FE    // dwSignature
002C 00 00 01 00    // dwStrucVersion
0030 00 00 06 00    // dwFileVersionMS = 6.0
0034 35 0B 54 0B    // dwFileVersionLS = 2900.2869
0038 00 00 06 00    // dwProductVersionMS = 6.0
003C 35 0B 54 0B    // dwProductVersionLS = 2900.2869
0040 3F 00 00 00    // dwFileFlagsMask
0044 00 00 00 00    // dwFileFlags
0048 04 00 04 00    // dwFileOS = VOS_NT_WINDOWS32
004C 02 00 00 00    // dwFileType = VFT_DLL
0050 00 00 00 00    // dwFileSubtype
0054 00 00 00 00    // dwFileDateMS
0058 00 00 00 00    // dwFileDateLS
005C          // no padding needed

```

As with the 16-bit version resource, the root node is always a binary node consisting of a `VS_FIXEDFILEINFO`. After the root node come its children.

```

005C F6 02          // cbNode (node ends at 0x005C + 0x02F6 = 0x0352)
005E 00 00          // cbData (no data)
0060 01 00          // wType = 1 (string data)
0062 53 00 74 00 72 00 69 00 6E 00 67 00 46 00 69 00
      6C 00 65 00 49 00 6E 00 66 00 6F 00 00 00
      // L"StringFileInfo" + null terminator
007E 00 00          // padding to restore alignment

```

The `StringFileInfo` contains no data, so the fact that it's string data is irrelevant. As before, the children of the `StringFileInfo` are language nodes.

```

0080 D2 02          // cbNode (node ends at 0x0080 + 0x02D2 = 0x0352)
0082 00 00          // cbData (no data)
0084 01 00          // wType = 1 (string data)
0086 30 00 34 00 30 00 39 00 30 00 34 00 42 00 30 00
      00 00          // L"040904B0" + null terminator
0098          // no padding needed

```

The children of the language node are the strings that make up the bulk of the version information.

```
0098 4C 00          // cbNode (node ends at 0x0088 + 0x004C = 0x00D40)
009A 2C 00          // cbData
009C 01 00          // wType = 1 (string data)
009E 43 00 6F 00 6D 00 70 00 61 00 6E 00 79 00 4E 00
      61 00 6D 00 65 00 00 00
      // L"CompanyName" + null terminator
00B6 00 00          // padding to restore alignment
00B8 4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00
      74 00 20 00 43 00 6F 00 72 00 70 00 6F 00 72 00
      61 00 74 00 69 00 6F 00 6E 00 00 00
      // L"Microsoft Corporation" + null terminator
00E4          // no padding needed
```

Notice that for string types, the `cbData` includes the null terminator.

```

00E4 5A 00          // cbNode (node ends at 0x00E4 + 0x005A = 0x013E)
00E6 32 00          // cbData
00E8 01 00          // wTypes = 1 (string data)
00EA 46 00 69 00 6C 00 65 00 44 00 65 00 73 00 63 00
      72 00 69 00 70 00 74 00 69 00 6F 00 6E 00 00 00
      // L"FileDescription" + null terminator
010A 00 00          // padding to restore alignment
010C 57 00 69 00 6E 00 64 00 6F 00 77 00 73 00 20 00
      53 00 68 00 65 00 6C 00 6C 00 20 00 43 00 6F 00
      6D 00 6D 00 6F 00 6E 00 20 00 44 00 6C 00 6C 00
      00 00
      // L"Windows Shell Common Dll" + null terminator
013E 00 00          // padding to restore alignment
0140 74 00          // cbNode (node ends at 0x0140 + 0x0074 = 0x01B4)
0142 54 00          // cbData
0144 01 00          // wType = 1 (string data)
0146 46 00 69 00 6C 00 65 00 56 00 65 00 72 00 73 00
      69 00 6F 00 6E 00 00 00
      // L"FileVersion" + null terminator
015E 00 00          // padding to restore alignment
0160 36 00 2E 00 30 00 30 00 2E 00 32 00 39 00 30 00
      30 00 2E 00 32 00 38 00 36 00 39 00 20 00 28 00
      78 00 70 00 73 00 70 00 5F 00 73 00 70 00 32 00
      5F 00 67 00 64 00 72 00 2E 00 30 00 36 00 30 00
      33 00 31 00 36 00 2D 00 31 00 35 00 31 00 32 00
      29 00 00 00
      // L"6.00.2900.2869 (xpsp_sp2_gdr.060316-1512)"
      // + null terminator
01B4          // no padding needed
01B4 30 00          // cbNode (node ends at 0x01B4 + 0x0030 = 0x01E4)
01B6 10 00          // cbData
01B8 01 00          // wType = 1 (string data)
01BA 49 00 6E 00 74 00 65 00 72 00 6E 00 61 00 6C 00
      4E 00 61 00 6D 00 65 00 00 00
      // L"InternalName" + null terminator
01D4          // no padding needed
01D4 53 00 48 00 45 00 4C 00 4C 00 33 00 32 00 00 00
      // L"SHELL32" + null terminator
01E4          // no padding needed
01E4 80 00          // cbNode (node ends at 0x01E4 + 0x0080 = 0x0264)
01E6 5C 00          // cbData
01E8 01 00          // wType = 1 (string data)
01EA 4C 00 65 00 67 00 61 00 6C 00 43 00 6F 00 70 00
      79 00 72 00 69 00 67 00 68 00 74 00 00 00
      // L"LegalCopyright" + null terminator
0208          // no padding needed
0208 A9 00 20 00 4D 00 69 00 63 00 72 00 6F 00 73 00
      6F 00 66 00 74 00 20 00 43 00 6F 00 72 00 70 00
      6F 00 72 00 61 00 74 00 69 00 6F 00 6E 00 2E 00
      20 00 41 00 6C 00 6C 00 20 00 72 00 69 00 67 00
      68 00 74 00 73 00 20 00 72 00 65 00 73 00 65 00
      72 00 76 00 65 00 64 00 2E 00 00 00

```

```

        // L"© Microsoft Corporation. "
        // L"All rights reserved." + null terminator
0264      // no padding needed
0264 40 00      // cbNode (node ends at 0x0264 + 0x0040 = 0x02A4)
0266 18 00      // cbData
0268 01 00      // wType = 1 (string data)
026A 4F 00 72 00 69 00 67 00 69 00 6E 00 61 00 6C 00
        46 00 69 00 6C 00 65 00 6E 00 61 00 6D 00 65 00
        00 00      // L"OriginalFilename" + null terminator
028C      // no padding needed
028C 53 00 48 00 45 00 4C 00 4C 00 33 00 32 00 2E 00
        44 00 4C 00 4C 00 00 00
        // L"SHELL32.DLL" + null terminator
02A4      // no padding needed
02A4 6A 00      // cbNode (node ends at 0x02A4 + 0x006A = 0x030E)
02A6 25 00 01 00 50 00 72 00 6F 00 64 00 75 00 63 00
        74 00 4E 00 61 00 6D 00 65 00 00 00
        // L"ProductName" + null terminator
02C2 00 00      // padding to restore alignment
02C4 4D 00 69 00 63 00 72 00 6F 00 73 00 6F 00 66 00
        74 00 AE 00 20 00 57 00 69 00 6E 00 64 00 6F 00
        77 00 73 00 AE 00 20 00 4F 00 70 00 65 00 72 00
        61 00 74 00 69 00 6E 00 67 00 20 00 53 00 79 00
        73 00 74 00 65 00 6D 00 00 00
        // L"Microsoft® Windows® "
        // L"Operating System" + null terminator
030E 00 00      // padding to restore alignment
0310 42 00      // cbNode (node ends at 0x0310 + 0x0042 = 0x0352)
0312 1E 00      // cbData
0314 01 00      // wType = 1 (string data)
0316 50 00 72 00 6F 00 64 00 75 00 63 00 74 00 56 00
        65 00 72 00 73 00 69 00 6F 00 6E 00 00 00
        // L"ProductVersion" + null terminator
0334      // no padding needed
0334 36 00 2E 00 30 00 30 00 2E 00 32 00 39 00 30 00
        30 00 2E 00 32 00 38 00 36 00 39 00 00 00
        // L"6.00.2900.2869" + null terminator
0352 00 00      // padding to restore alignment

```

At offset `0x0352` , we've reached the end of the `ProductVersion` node, the language node, and the `StringFileInfo` node. The next node is therefore a sibling to `StringFileInfo` .

```

0354 44 00      // cbNode (node ends at 0x0354 + 0x0044 = 0x0394)
0356 00 00      // cbData
0358 01 00      // wType = 1 (string data)
035A 56 00 61 00 72 00 46 00 69 00 6C 00 65 00 49 00
        6E 00 66 00 6F 00 00 00
        // L"VarFileInfo" + null terminator
0372 00 00      // padding to restore alignment

```

Since we haven't exhausted the `VarFileInfo` node, the next node is a child.

```

0374 24 00          // cbNode (node ends at 0x0374 + 0x0024 = 0x0394)
0376 04 00          // cbData
0378 00 00          // wType = 0 (binary data)
037A 54 00 72 00 61 00 6E 00 73 00 6C 00 61 00 74 00
      69 00 6F 00 6E 00 00 00
      // L"Translation" + null terminator
0392 00 00          // padding to restore alignment
0394 09 04 B0 04    // 0x0409 = US English
      // 0x04B0 = 1200 = Unicode UCS-2 little-endian
0398                // no padding needed

```

We have reached the end of the `Translation` node, the `VarFileInfo` node, and the root node. Our parsing of the version data is complete and resulted in the following version resource:

```

FILEVERSION      3,0,2900,2869
PRODUCTVERSION  3,0,2900,2869
FILEFLAGSMASK   VS_FFI_FILEFLAGSMASK
FILEFLAGS       0
FILEOS          VOS_NT_WINDOWS32
FILETYPE        VFT_DLL
FILESUBTYPE     VFT_UNKNOWN
BEGIN
  BLOCK "StringFileInfo"
  BEGIN
    BLOCK "040904B0"
    BEGIN
      VALUE "CompanyName", "Microsoft Corporation"
      VALUE "FileDescription", "Windows Shell Common Dll"
      VALUE "FileVersion", "6.00.2900.2869 (xpsp_sp2_gdr.060316-1512)"
      VALUE "InternalName", "SHELL32"
      VALUE "LegalCopyright", "\251 Microsoft Corporation. All rights reserved."
      VALUE "OriginalFilename", "SHELL32.DLL"
      VALUE "ProductName", "Microsoft\256 Windows\256 Operating System"
      VALUE "ProductVersion", "6.00.2900.2869"
    END
  END
  BLOCK "VarFileInfo"
  BEGIN
    VALUE "Translation", 0x0409, 0x04B0
  END
END

```

There you have it, the format of 32-bit version resources. Future versions of Windows may extend this format, but that's what it looks like up through Windows Vista.

Now, after all that byte-mashing, I have to confess that I've been lying to you. What you saw was not the actual version resource in `shell32.dll`. We'll discuss what's really going on next time.

Raymond Chen

**Follow**

