

What can I do if I don't want my file version number to be a sequence of four integers?

 devblogs.microsoft.com/oldnewthing/20230503-51

May 3, 2023



Raymond Chen

In the Windows file version resource, the file version number is given as a sequence of four 16-bit integers. Here's the prototype file version resource given [on docs.microsoft.com](https://docs.microsoft.com):

```

#define VER_FILEVERSION          3,10,349,0
#define VER_FILEVERSION_STR     "3.10.349.0\0"

#define VER_PRODUCTVERSION      3,10,0,0
#define VER_PRODUCTVERSION_STR  "3.10\0"

#ifndef DEBUG
#define VER_DEBUG                0
#else
#define VER_DEBUG                VS_FF_DEBUG
#endif

VS_VERSION_INFO VERSIONINFO
FILEVERSION      VER_FILEVERSION
PRODUCTVERSION  VER_PRODUCTVERSION
FILEFLAGSMASK   VS_FFI_FILEFLAGSMASK
FILEFLAGS       (VER_PRIVATEBUILD|VER_PRERELEASE|VER_DEBUG)
FILEOS          VOS__WINDOWS32
FILETYPE        VFT_DLL
FILESUBTYPE     VFT2_UNKNOWN
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904E4"
        BEGIN
            VALUE "CompanyName",      VER_COMPANYNAME_STR
            VALUE "FileDescription",  VER_FILEDESCRIPTION_STR
            VALUE "FileVersion",      VER_FILEVERSION_STR
            VALUE "InternalName",     VER_INTERNALNAME_STR
            VALUE "LegalCopyright",   VER_LEGALCOPYRIGHT_STR
            VALUE "LegalTrademarks1", VER_LEGALTRADEMARKS1_STR
            VALUE "LegalTrademarks2", VER_LEGALTRADEMARKS2_STR
            VALUE "OriginalFilename", VER_ORIGINALFILENAME_STR
            VALUE "ProductName",      VER_PRODUCTNAME_STR
            VALUE "ProductVersion",   VER_PRODUCTVERSION_STR
        END
    END
END

BLOCK "VarFileInfo"
BEGIN
    /* The following line should only be modified for localized versions. */
    /* It consists of any number of WORD,WORD pairs, with each pair */
    /* describing a language,codepage combination supported by the file. */
    /* */
    /* For example, a file might have values "0x409,1252" indicating that it */
    /* supports English language (0x409) in the Windows ANSI codepage (1252). */

    VALUE "Translation", 0x409, 1252
END
END

```

But what if your file versioning scheme isn't major.minor.build.revision? For example, maybe your file versioning scheme is major.minor.patch? Can you just leave out the fourth value?

```
// Does this work?
#define VER_FILEVERSION          3,10,349
#define VER_FILEVERSION_STR     "3.10.349\0"
```

If you try this, you'll find that many tools show the version as 3.1.349.0. Where is the extra .0 coming from?

The file version information is recorded in two formats in the version resource. There is a binary machine-parseable version, and there is a string human-readable version. The tools are probably getting the machine-parseable version, which takes the form of a VS_FIXED-FILEINFO structure.

```
typedef struct tagVS_FIXEDFILEINFO {
    DWORD dwSignature;
    DWORD dwStrucVersion;
    DWORD dwFileVersionMS;
    DWORD dwFileVersionLS;
    DWORD dwProductVersionMS;
    DWORD dwProductVersionLS;
    DWORD dwFileFlagsMask;
    DWORD dwFileFlags;
    DWORD dwFileOS;
    DWORD dwFileType;
    DWORD dwFileSubtype;
    DWORD dwFileDateMS;
    DWORD dwFileDateLS;
} VS_FIXEDFILEINFO;
```

The file version number is represented in the form of two 32-bit integers `dwFileVersionMS` and `dwFileVersionLS`, and the traditional interpretation of these values is to break it down as follows:

dwFileVersionMS		dwFileVersionLS	
1	0		
5			
major	minor	build	revision

Technically, it's just a 64-bit number, and you could try to get the word out to the entire industry that, "For my app, please don't break it down into four 16-bit integers. Instead, break it down into three 16-bit integers, and ignore the last one. Thanks. Love ya!"

You may even succeed at convincing some tools vendors to go along with your special rule. But you're unlikely to convince all of them.

You can put whatever 64-bit value you like, but you have to accept that tools are going to parse it as if it were four 16-bit integers. You can tell your customers, "If you try to view the file version information, many tools will show it in four parts rather than three. You can ignore the last part, the one with the trailing `.0`."

There's a bit of an escape hatch: The `VER_FILEVERSION_STR`. You can set this string to anything you like. By convention, it's a human-readable version of the binary file information, but there is no enforcement.

```
// Don't show the final .0 to the user. We don't use it.  
#define VER_FILEVERSION_STR          "3.10.349\0"
```

You can even put extra bonus information in there if you like.

```
#define VER_FILEVERSION_STR          "3.10.349 (prerelease)\0"
```

Most tools will also show this string to the user, although it carries no formal meaning.