

LONG FILE NAMES (LFN) STRUCTURE

How Windows 95 Stores Long Filenames

Jeff Prosis

Copyright notice: Taken from PC Magazine, June 25, 1996 by Jeff Prosis

"Windows 95 stores short filenames the same way DOS and 16-bit windows so. Every file on every disk is accompanied by a 32-byte directory entry that records the name of the file as well as the file's attributes, a date and time stamp, and other information."

The format of the short directory entry is as follows:

Offset	Description	Size
0	Filename	8 bytes (ASCII)
8	Filename extension	3 bytes (ASCII)
11	File attributes	1 byte (encoded)
12	reserved	10 bytes
22	Time stamp	2 bytes (encoded)
24	Date stamp	2 bytes (encoded)
26	Starting cluster	2 bytes
28	File size	4 bytes

File attributes byte

7: reserved	3: Volume label
6: reserved	2: System
5: archive	1: Hidden
4: subdirectory	0: Read-only

Time stamp byte

11-15: Hours (0-23)
5-10: Minutes (0-59)
0- 4: Seconds divided by 2 (0-29)

Date stamp byte

11-15: Year (relative to 1980)
5- 8: Month (1-12)
0- 4: Day of month (0-31)

"Because of compatibility issues, adding long filename support to an operating system that uses 8.3 filenames isn't as expanding directory entries to hold more than 11 characters. ...

Windows 95's designers devised a clever solution to the problem of supporting long filenames while preserving compatibility with previous versions of DOS and Windows applications. ... Through testing, Microsoft found that if a directory entry is marked with an "impossible" combination of read-only, hidden, system, and volume label attribute bits - that is, if the directory entry's attribute byte holds the value 0Fh - the enumeration functions built inot all existing versions of DOS and pre-95 versions of Windows will skip over that directory entry as if it weren't there.

The solution for Windows 95, then, was to store two names for every file and subdirectory: a short name that's visible to all applications and a long name that's visible only to Windows 95 applications...Short filenames are stored in 8.3 format in conventional 32-byte directory entries. Windows creates a short filename from a long one by truncating it to six uppercase characters and adding "~1" to the end of the base filename. If there's

LONG FILE NAMES (LFN) STRUCTURE

How Windows 95 Stores Long Filenames

Jeff Prosis

already another filename with the same first six characters, the number is incremented. The extension is kept the same, and any character that was illegal in earlier versions of Windows and DOS is replaced with an underscore.

Long filenames are stored in specially formatted 32-byte long filename (LFN) directory entries marked with attribute bytes set to 0Fh. For a given file or subdirectory, a group of one or more LFN directory entries immediately precedes the single 8.3 directory entry on the disk. Each LFN directory entry contains up to 13 characters of the long filename, and the OS strings together as many as needed to comprise an entire long filename.

Filenames are stored in Unicode format, which requires 2 bytes per character as opposed to ASCII's 1 byte. Filename characters are spread among three separate fields: the first 10 bytes (five characters) in length, the second 12 bytes (6 characters), and the third 4 bytes (two characters). The lowest five bits of the directory entry's first byte hold a sequence number that identifies the directory entry's position relative to other LFN directory entries associated with the same file. If a long filename requires three LFN directory entries, for example, the sequence number of the first will be 1, that of the second will be 2, and the sequence of the third will be 3. Bit 6 of the third entry's first byte is set to 1 to indicate that it's the last entry in the sequence.

The attribute field appears at the same location in LFN directory entries as in 8.3 directory entries. ... The starting cluster number field also appears at the same location, but in LFN directory entries its value is always 0. The type indicator field also holds 0 in every long filename I've examined, but Adrian King's *Inside Windows 95* (Microsoft Press, 1994) says it can also hold a nonzero value indicating that the directory entry contains "class information" for the corresponding file. ... The LFN directory entry's checksum byte holds an eight-bit checksum value computed by adding certain fields of the 8.3 directory entry and performing a modulo 256 operation on the result. Windows 95 uses this checksum to detect orphaned or corrupted LFN directory entries.

Long filename directory entry

OFFSET	DESCRIPTION	Size
0	Sequence byte	1 byte
1	First five characters of LFN	10 bytes
11	File attributes	1 byte
12	Type indicator	1 byte (always 0??)
13	Checksum	1 byte
14	Next six characters of LFN	12 bytes
26	Starting cluster number	2 bytes (always 0)
28	Next two characters of LFN	4 bytes

*NOTE: The above structure may span up to 31 entries. The last entry will be a standard 8.3 filename directory structure.

Sequence byte

7: apparently unused (always 0)

6: 1=final component of this LFN

LONG FILE NAMES (LFN) STRUCTURE

How Windows 95 Stores Long Filenames

Jeff Prosis

5: apparently unused (always 0)
0-4: sequence number (1-31)

```
-----  
Robert Vandervelde + ...that what we have learned and  
RVand@snowhill.com + truly understood, we discovered  
Enterprise, AL     + ourselves.  
The Wiregrass      +           - Richard C. Dorf  
-----
```