

Subject: : AmigaOS4

Topic: : DevilutionX - Diablo 1

Re: DevilutionX - Diablo 1

Author: : Hans

Date: : 2019/2/3 1:54:14

URL:

@NoCache

Does the code you're compiling work on big-endian CPUs?

If the code reads the file in as one large buffer and the file was generated for little-endian machines, then the code snippet you posted will read values round the wrong way.

If endianness is the problem, then you'll need to:

1. Have a file with endianness conversion macros (e.g., LE32\_TO\_CPU())
2. Go through the code and insert the macros wherever data from the file is being read (e.g., LE64\_TO\_CPU(pHeader->HetTablePos64)) (that's assuming that HetTablePos64 is 64-bit

HINTS:

- #include <machine/endian.h> defines BYTE\_ORDER, so you can use #if (BYTE\_ORDER == BIG\_ENDIAN) to define the correct versions of the macros in both big and little-endian platforms
- GCC 4.4.0 and newer have built in \_\_builtin\_bswap16(x), \_\_builtin\_bswap32(x), \_\_builtin\_bswap64(x)

Hans