

Re: byte alignment/dynamic memory from heap x64

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2007-09/msg00257.L>

- *From:* Tim Roberts <timr@xxxxxxxx>
 - *Date:* Sun, 16 Sep 2007 19:14:56 -0700
-

Mike L <MikeL@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

My IM driver uses SLIST_ENTRY and SLIST_HEADER(s) and they are declared in a struct and when my driver is loaded I dynamically allocate memory using these structs.

for example(not my real struct...)

```
struct X {  
  
    int y;  
    PNDIS_BUFFER pBuf;  
    ....  
    SLIST_HEADER ListHeader;  
    SLIST_ENTRY ListEntry;  
    ...  
};
```

The documentation for x64 states that SLIST_ENTRY and SLIST_HEADER must be 16 byte aligned.

How do I force this to happen ?

You don't have to. A few minutes browsing through the include files will show that SLIST_HEADER and SLIST_ENTRY are already declared with 16-byte alignment. As long as your structure starts on a 16-byte boundary, the header and entry structures will be so aligned.

Any advice is appreciated, I might have a bug in my code for one of the above methods, so I really just need someone to tell me how this is normally done and I can figure out the details. I saw another function `_align_memory`, but did not want to link an NDIS driver to the c runtime for memory management, but maybe that is another alternative.

Re: byte alignment/dynamic memory from heap x64

You're sure you are compiling for x64? In the 32-bit build environments, they aren't 16-byte aligned because they don't need to be.

--

Tim Roberts, timr@xxxxxxxxx
Providenza & Boekelheide, Inc.

.