

Re: Strange behaviour of linker (vc++)

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-09/1340.html>

From: Igor Tandetnik (itandetnik_at_mvps.org)

Date: 09/27/04

Date: Mon, 27 Sep 2004 15:11:43 -0400

"Philip Lawatsch" <philip@waug.at> wrote in message
news:41585a58\$1@e-post.inode.at

>> *You can force an inclusion of the symbol with /INCLUDE linker command
>> line option, but that kinda defeats the purpose.*

>

> *Well, I disagree. At least partially. I suppose one has reasons as to
> why one would initialise a static variable via a function.*

>

> *Anyway, I have tons of such things and they work fine with g++. So, is
> there anything I can do to force all "dead" code into my binary? Its
> not reasonable to specify every symbol per hand, this is hell a lot
> of work.*

That's precisely what I meant by "defeats the purpose". The idea of the idiom is to build the catalog implicitly, just by linking with corresponding libraries, but /INCLUDE trick makes you reference each class explicitly.

Another alternative is /OPT:NOREF linker option, which instructs the linker to keep everything in, whether used or not. This may result in a large executable.

Yet another is to have some kind of Init() function in your library, and have the client call it. You are back to assembling your catalog explicitly, but at least you don't need to figure out the exact mangled name of a particular variable.

>> *Needed how? As far as linker is concerned, no outside module ever
>> references this symbol, so it does not need to be included.*

>

> *Yea, but there is a function wich initialises a static in there. This
> function references outside stuff. So imo the linker should not
> discard it.*

Well, the linker decides whether to include a function not based on what the function references, but based on whether the function `_itself_` is referenced by something. Otherwise two functions calling each other

microsoft.public.vc.language: Re: Strange behaviour of linker (vc++)

would never be discarded, even though neither is being called. Once the linker decides that the function is needed, it further pulls in the stuff referenced by this function.

--

With best wishes,

Igor Tandetnik

"On two occasions, I have been asked [by members of Parliament], 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' I am not able to rightly apprehend the kind of confusion of ideas that could provoke such a question." -- Charles Babbage