

Re: Different C runtime libraries

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2005-07/msg00480.html>

- *From:* "Eugene Gershnik" <gershnik@xxxxxxxxxxx>
 - *Date:* Mon, 11 Jul 2005 14:18:51 -0700
-

Igor Tandetnik wrote:

```
> "Eugene Gershnik" <gershnik@xxxxxxxxxxx> wrote in message
> <news:uhlG4NlhFHA.3652@xxxxxxxxxxxxxxxxxxxxxxxx>
>>> I really see no reason at all. For example the debug and release CRT
>>> version should have exactly the same interface. Only be the
>>> implementation should differ, not the interface .h files! If my .LIB
>>> references "printf", the printf function to use should be the
>>> version of the CRT at link time. .LIB should be only seen as a
>>> bunch of .OBJ files put together. Why can't the world be so easy?
>>
>> And if my lib creates a FILE * then passes it to your lib and your
>> lib passes it to fprintf which fprintf should be called?
>
> Whichever the final executable is linked against. Remember that
> neither LIB actually contains code for fprintf nor fopen, just
> references to them. These references are resolved only when all OBJs
> and LIBs are brought together and the executable is ultimately built.
```

I am not sure what you mean by the above. I do remember where the code is located. Which one should the exe pick the one my lib expects or yours? This is precisely what is discussed below.

```
>> If it is yours it
>> cannot handle a FILE that has different members from what it expects.
>> And my multithreaded debug FILE will be very different from your
>> singlethreaded release one.
>
> The idea is that the executable, rather than individual LIBs, chooses
> which CRT to link everything against. Then, you cannot possibly end up
> with mismatched versions of fopen and fprintf that don't agree on the
> layout of FILE.
```

Have you heard of macros or inline functions? If I have for example (not a real code)

```
#define fputc(c, f) (f)->somefield++ = (c)
```

I bet there will be problems. For better or worse my FILE may be different from your FILE and they expect different fprintfs. Stdio `_almost_` solves the

Re: Different C runtime libraries

problem by using an "abstarct" FILE * and fopen factory but this is not enough. This is even worse for other parts of library.

- > But attacking straw men does
- > not help much to resolve this question.

I am not sure where you see the strawmen.

--

Eugene

<http://www.gershnik.com>

-
- *Follow-Ups:*
 - ◆ *Re: Different C runtime libraries*
 - ◇ *From:* Igor Tandetnik
 - *References:*
 - ◆ *Different C runtime libraries*
 - ◇ *From:* SerGioGio
 - ◆ *Re: Different C runtime libraries*
 - ◇ *From:* Eugene Gershnik
 - ◆ *Re: Different C runtime libraries*
 - ◇ *From:* Igor Tandetnik
 - Prev by Date: *Re: Different C runtime libraries*
 - Next by Date: *Re: Different C runtime libraries*
 - Previous by thread: *Re: Different C runtime libraries*
 - Next by thread: *Re: Different C runtime libraries*
 - Index(es):
 - ◆ *Date*
 - ◆ *Thread*