

Re: How do I create a function in my library for passing user callback function

Re: How do I create a function in my library for passing user callback function

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2008-04/msg02374.html

- *From:* Ben Bacarisse <ben.usenet@xxxxxxxxxx>
 - *Date:* Tue, 15 Apr 2008 01:07:34 +0100
-

"Bill Reid" <hormelfree@xxxxxxxxxxxxxxxxxxxx> writes:

Ben Bacarisse <ben.usenet@xxxxxxxxxx> wrote in message
news:87bq4cci3r.fsf@xxxxxxxxxxxx

"Bill Reid" <hormelfree@xxxxxxxxxxxxxxxxxxxx> writes:
<snip lots>

<snip>

```
extern void my_library_function (int (*)(const char*));

library_c.c

#include "library_h.h"

void my_library_function(int my_callback_function(const
char*)) {
}
```

You must be aware that is this not the normal syntax.

Sure it is. I use it all the time, so it's normal to me.

That's fine, but I would not suggest it as a "first example" or in "how to" code just because it needs more explanation.

Usually, the function definition is similar to the declaration, except the ';' is replaced by the function body.

Re: How do I create a function in my library for passing user callback function

Right. Let me clue you in on something about me: I try to conserve keystrokes where they AREN'T needed, and use them where they help me understand my own code.

Again, that is up to you, but in my editor copying the prototype is fewer keystrokes than writing the two forms you used.

<snip>

Function pointer parameters usually look like this:

```
void my_library_function(int (*my_callback_function)(const char*)) { }
```

Sure. But then I have to type "("" (three whole extra characters!) where I don't need to type any more characters, because the two ways of defining the parameters ARE TOTALLY FUNCTIONALLY IDENTICAL.

which is, of course, how you wrote it in the function's prototype.

There, I'm forced to write it that way, unless I again want to type unwanted characters...WHICH I DON'T.

This I don't get. You can save two keystrokes if you use the same style in the prototype (but let me guess: a parameter name in a prototype is "unwanted characters").

<snip>

The answer is that there is a special dispensation:

A declaration of a parameter as "function returning type" shall be adjusted to "pointer to function returning type"

Great, which of course completely conforms to the basic idea above that a function name is actually a function pointer by the simple necessity of computer science...I'm assuming this is "standard" language, so despite you calling it a "special dispensation" IT IS TOTALLY CONFORMING "STANDARD" SYNTAX.

Yes, but that does not make it good syntax. With your example, a beginner would wonder why a variable declared to store this callback

Re: How do I create a function in my library for passing user callback function

Re: How do I create a function in my library for passing user callback function

pointer looked so different to the parameter that receives it. C has enough special cases already.

I don't want to have a style war about how you write your programs, but in "how to" postings I would suggest being consistent and avoiding special cases. At least some of the "fuzz" here may have come from your use of two different styles.

--

Ben.

.