

Re: Array of char pointers.

Re: Array of char pointers.

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2008-04/msg00606.html>

- *From:* Barry Schwarz <schwarzb@xxxxxxxx>
 - *Date:* Mon, 14 Apr 2008 19:00:50 -0700
-

On Mon, 14 Apr 2008 09:31:13 -0700 (PDT), goodTweeTieBird <goodTweeTieBird@xxxxxxxx> wrote:

On Apr 13, 9:57 pm, Barry Schwarz <schwa...@xxxxxxxx> wrote:

On Sun, 13 Apr 2008 09:25:11 -0700 (PDT), goodTweeTieBird

<goodTweeTieB...@xxxxxxxx> wrote:

If I had to say what "arr" is, I would say it is an array of pointers to chars. If so why is it intialized with strings instead of addresses?

```
char *arr[] =
{
    "a", "b", "c", "dd", "eee"
};
```

It is not initialized with strings. It is initialized with the address of strings.

I misspoke, I should have said why does it appear to be initialized with strings as I am providing strings and not addresses. But you figured out what I meant.

Instead of

```
char *arr[] =
```

should I write

```
char const *arr[] =
```

Re: Array of char pointers.

If the elements of arr will always point to sting literals, then the const modifier is appropriate. String literals are not modifiable and the const modifier compels the compiler to issue a diagnostic if you try to change any part of them.

However, the const modifier is completely unrelated to your question. It only appears you are initializing the array with strings. Each of the string literals is actually an array. In this context, an expression of array type is converted to the address of the first array element with type pointer to element. So the "a" is actually an array of two char ('a' and '\0') which is evaluated as the address of the 'a' with type char*. This is exactly the type that should be assigned to arr[0].

Remove del for email

.