Difference between <cstring> and <string>

Earlier today (actually yesterday due to my time-zone) I was attempting a programming interview using *Visual Studio 2012* for C++ on Interview Street (which uses *g++*).

To be brief, I came across several compilation errors[1] when I was using

```
#include <cstring>
```

which was provided by the skeleton code in one of the question, and after turning to

```
#include <string>
```

all compilation errors magically disappeared.

However, upon submission to Interview Street, I had to add `c` back; otherwise I got compilation errors.

It was the first time I was bitten by non-standardization....

My question is: what inside `<string>` and `<cstring>` took me (precious) more than half an hour?

[1] For anyone who is curious:

One error by Visual Studio 2012 if `using <cstring>` is:

> error C2338: The C++ Standard doesn't provide a hash for this type.

in

> c:\program files (x86)\microsoft visual studio 11.0\vc\include\xstddef

possibly for `string` as key in `unordered_map`

One error by g++ if `using <string>` is:

> 'strlen' was not declared in this scope

c++ visual-studio g++

edited Oct 11 '12 at 6:09

asked Oct 10 '12 at 16:59
Dante is not a Geek
6,021 5 26 60

3 Answers

The `cstring` header provides functions for dealing with C-style strings — null-terminated arrays of characters. This includes functions like `strlen` and `strcpy`. It's the C++ version of the classic `string.h` header from C.

The `string` header provides the `std::string` class and related functions and operators.

The headers have similar names, but they're not really related beyond that. They cover separate tasks.

answered Oct 10 '12 at 17:03

Rob Kennedy
131k 13 193 359

1 The puzzle is: g++ does not seem to need `<string>` for `string` , whereas Visual Studio does not seem to need `<cstring>` for c stuff. Why? – Dante is not a Geek Oct 10 '12 at 17:08

5 The C++ standard allows standard headers to include other standard headers, so you could get the contents of `string` from any of the other headers you've already included, such as `iostream` . Likewise for `cstring` . Always include *all* the headers you need. Don't rely on your specific environment to implicitly include some of them for you; you'll run into problems if you need your code to be portable, or if you change compiler versions and the new version has different implicit header dependencies. – Rob Kennedy Oct 10 '12 at 17:09

6 @DanteisnotaGeek: Implementations are allowed to include other headers in their headers. Most probably VS adds `#include <cstring>` in some of the other headers you include, and gcc adds `#include <string>` in some header you included. You *should* include **both** since your code depends on both headers. – David Rodríguez - dribeas Oct 10 '12 at 17:12

In C++, you wouldn't use `#include <somefile.h>` , but instead `#include <somefile>` . Now C++ has its string classes in `<string>` , but the c-string functions are also available, which would be in `<string.h>` . C++ uses for 'traditional' c- include files. Therefore, `<cstring>` and `<string>`

http://www.cplusplus.com/reference/clibrary/cstring/

edited Oct 10 '12 at 17:13

answered Oct 10 '12 at 17:03

Rudolf Mühlbauer
1,960 9 14

The puzzle is: g++ does not seem to need `<string>` for `string` , whereas Visual Studio does not seem to need `<cstring>` for c stuff. Why? – Dante is not a Geek Oct 10 '12 at 17:09

1 @DanteisnotaGeek good question. MSVC always was a bit strange. – Rudolf Mühlbauer Oct 10 '12 at 17:13

`<cstring>` has the C string code from the C header string.h. `C++` has a convention where `c` headers have the same base name, except for a leading `c` and no trailing `.h` . All the contents are available under the `std::` namespace.

`<string>` has the standard library `std::string` and related functions

edited Oct 10 '12 at 17:17

answered Oct 10 '12 at 17:04

juanchopanza
165k 15 222 346

1 The puzzle is: g++ does not seem to need `<string>` for `string` , whereas Visual Studio does not seem to need `<cstring>` for c stuff. Why? – Dante is not a Geek Oct 10 '12 at 17:08

@DanteisnotaGeek then `<string>` must be included in a different header. You should always include `<string>` if you need it. I don't think there is any standard header that is guaranteed to include it. – juanchopanza Oct 10 '12 at 17:11