
Subject: From std::string to String
Posted by [Giorgio](#) on Wed, 14 Feb 2018 13:35:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi there,
in my application I read a text file with an Ansi codepage (Windows1252) and put part of the text in a vector of std::string. Later, I use those std::string to perform a simple SQL query:

```
std::vector<std::string> fields;  
  
Insert(My_Table)(My_Field1, (String)fields[0])  
(My_Field2, (String)fields[1])  
(My_Field3, (String)fields[2]);
```

I have to cast from std::string to String (that is UTF8, if I remember correctly): I have an error compiling if I use std::string directly. The problem is that with "strange" characters (e.g. Ö) the

convert from std::string to String?
Thanks,
Gio

Subject: Re: From std::string to String
Posted by [Klugier](#) on Wed, 14 Feb 2018 20:20:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Try to create Upp::String basing on std::string like this:

```
String(fields[0]);
```

IMO, you should use Core library to deal with text files. The easiest way is to use LoadFile(path):

```
Upp::String fileContent = Upp::LoadFile("PathToTextFile.txt");  
  
// now you could use some powerfull functions from Core like Split;  
Upp::Vector<String> fileContentLines = Upp::Split(fileContent, "\n");  
for (const String& line : fileContentLines) {  
    // Loop over the line...  
}
```

I will start from rewriting you code from standard library into Upp approach. This could save you

problems with file encoding conversion.

Sincerely,
Klugier

Subject: Re: From std::string to String
Posted by [Giorgio](#) on Thu, 15 Feb 2018 08:47:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

Klugier wrote on Wed, 14 February 2018 21:20Hello,

Try to create Upp::String basing on std::string like this:

```
String(fields[0]);
```

Actually, I tried that already: it compiles, but I still have gibberish in the db instead of the correct characters

Klugier wrote on Wed, 14 February 2018 21:20

IMO, you should use Core library to deal with text files. The easiest way is to use LoadFile(path):

```
Upp::String fileContent = Upp::LoadFile("PathToTextFile.txt");
```

```
// now you could use some powerfull functions from Core like Split;  
Upp::Vector<String> fileContentLines = Upp::Split(fileContent, "\n");  
for (const String& line : fileContentLines) {  
    // Loop over the line...  
}
```

I will start from rewriting you code from standard library into Upp approach. This could save you problems with file encoding conversion.

Do you mean Upp can read data from an Ansi encoded file and convert it to utf-8 automatically?
Currently I use stream from standard c++ library to read data from file.

Subject: Re: From std::string to String
Posted by [omari](#) on Thu, 15 Feb 2018 09:10:34 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quote:

Do you mean Upp can read data from an Ansi encoded file and convert it to utf-8 automatically?
Yes, for your case :

```
String filecontent = LoadFileBOM(filepath, CHARSET_WIN1252);
```

...

Subject: Re: From std::string to String

Posted by [Giorgio](#) on Thu, 15 Feb 2018 10:40:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Ok, I found a quick & dirty solution: I convert formats using these functions, need to check if using Upp functions to load data solve the problem in a more elegant way.

Subject: Re: From std::string to String

Posted by [Giorgio](#) on Thu, 15 Feb 2018 14:35:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

I am in the process to use Core functionalities to rewrite my application, I have a couple of questions.

I use LoadFile() and noticed that it has no .close() method or something like that (it replaces a ifstream in c++ standard library and with it I have to open and close the file): my understanding is that I have not to open/close the stream as with ifstream; is my understanding correct?

There are a couple of functions that receive and return Upp::Vector<Upp::String> used as follows:

(MyApp.h)

```
Upp::Vector<Upp::String> SanitizeData(Upp::Vector<Upp::String> fields);
```

(MyApp.cpp)

```
Upp::Vector<Upp::String> fields;
```

(Here the Vector is filled in using data read from a file)

...

```
Upp::Vector<Upp::String> sanitized = SanitizeData(fields);
```

Previously, I used std::vector<std::string> and the application compiled, but now I get "error C2280: attempting to reference a deleted function".

Thanks,
Gio

Subject: Re: From std::string to String
Posted by [omari](#) on Thu, 15 Feb 2018 14:55:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

1 - LoadFile is a function that open a stream and read its content and close it, returning the content.

std::ifstream is a class, the Upp similare is Upp::FileIn, that is used by LoadFile

2 -

Quote:

(MyApp.h)

```
Upp::Vector<Upp::String> SanitizeData(Upp::Vector<Upp::String> fields);
```

you have to pass "fields" by reference :

```
Upp::Vector<Upp::String> SanitizeData(Upp::Vector<Upp::String>& fields);
```

and if it is readonly, it is recommended to use "const"

```
Upp::Vector<Upp::String> SanitizeData(const Upp::Vector<Upp::String>& fields);
```

Subject: Re: From std::string to String
Posted by [Oblivion](#) on Thu, 15 Feb 2018 15:03:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Giorgio

[It seems Omari has already replied.]

Quote:

I use LoadFile() and noticed that is has no .close() method or something like that (it replaces a ifstream in c++ standard library and with it I have to open and close the file): my understanding is that I have not to open/close the stream as with ifstream; is my understanding correct?

No, you don't. LoadFile() is a function, it does not have any method. It reads a file into a string, and closes the file/stream automatically.

If you need control over the stream, you can use FileIn, FileOut, FileAppend classes. (In fact, LoadFile() function uses an instance of FileIn)

They'll be closed automatically when they get out of their scope (when they are destroyed). In between you can explicitly call Close() if you need to.

Previously, I used `std::vector<std::string>` and the application compiled, but now I get "error C2280: attempting to reference a deleted function".

Probably because of move/copy semantics. Something is destroyed in the process, before it can be referenced. check if your data can be copied or moved.

As far as I can see, you are passing a copy of vector, just put an ampersand (&)

Best regards,
Oblivion

Subject: Re: From `std::string` to String
Posted by [Giorgio](#) on Thu, 15 Feb 2018 16:10:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you Oblivion and omari, your input was very helpful!
