Verbatim Literals in Managed C++? (like C#'s @"blah")

Asked 16 years ago Modified 5 years, 3 months ago Viewed 3k times

8

Is there a way to use verbatim String literals in managed C++? Similar to C#'s

```
String Docs = @"c:\documents and settings\"
```

.net c++ .net-2.0 string managed-c++

Share Improve this question Follow

edited Jun 22, 2010 at 11:03

Agnel Kurian
59.4k ● 47 ● 153 ● 225

asked Dec 9, 2008 at 16:52

Brian
5,956 ● 12 ● 61 ● 82

4 Answers

Sorted by: Highest score (default)

9

in C++11, there is raw string literal:

```
cout<<R"((\"ddd\aa)\n)"<<endl;
cout<<R"delimiter((\"ddd\aa)\n)delimiter"<<endl;
```

output is:

```
(\"ddd\aa)\n
(\"ddd\aa)\n
```

Share Improve this answer Follow

answered Mar 4, 2013 at 12:20

Wang
8,113 ● 5 ● 46 ● 77

6

This is not currently possible. Managed C++ string literals have almost the exact same rules as normal C++ strings. The managed C++ spec is in fact just an augmentation of the ANSI C++ standard.

Currently there is no support for C# style literal syntax in C++ (managed or not). You must manually escape every character.

See Section 9.1.3.3 in the C++/CLI spec for more details. (Spec Link)

Share Improve this answer Follow

answered Dec 9, 2008 at 17:12

JaredPar

753k • 151 • 1.3k • 1.5k

2 This answer was much more correct when it was given then it is now. – RichardPlunkett Dec 4, 2013 at 7:46

1 @RichardPlunkett in light of reflection of current state of things should we edit or remove this post? – John Leidegren Sep 16, 2014 at 11:59

Raw string literals can be used to achieve the desired result: String^ f = gcnew String(R"(C:\foo\bar.txt)"); [Google msdn C++ String Literals for more info] – Cameron Oct 9, 2014 at 17:52

3

While not quite as terse as the '@' C# verbatim string literal, the following does compile /Clr:pure, so you can use C++ Raw String Literals for pure MSIL and a similar result:

```
String^ f = gcnew String(R"(C:\foo\bar.txt)");
```

Raw string literals can be used in regular C++ also:

```
char *x = R"(C:\foo\bar.txt)";
```

Google "msdn C++ String Literals" for more info

Share

Improve this answer

Follow

edited Oct 9, 2014 at 18:01

answered Oct 9, 2014 at 17:55

Cameron

3,043 • 1 • 32 • 34

Oh! great!! Thank you very much – Cluster Oct 6, 2015 at 7:49

0

snip .. For .NET Programming, Visual C++ in Visual Studio 2017 supports the creation of mixed assemblies by using the /clr (Common Language Runtime Compilation) compiler option. The /clr:pure and clr:safe options are deprecated in Visual Studio 2015 and unsupported in Visual Studio 2017. If your code needs to be safe or verifiable, then we recommend that you port it to C#.

Share Improve this answer Follow

answered Sep 20, 2019 at 17:57

Romeo Macapobre
29 • 2