Subject: evaluate command line
Posted by BetoValle on Thu, 21 Jul 2022 20:04:36 GMT
View Forum Message <> Reply to Message

I found this code below and tested it successfully. But my intention is to evaluate the contents of each line that the mysqldump program generates. How to do this? Could someone help me with an example?
Grateful!

```
CONSOLE_APP_MAIN
{
 std::stringstream ss;
 std::string pathOfCommand ="C:\\Program Files (x86)\\MariaDB 10.1\\bin\\mysqldump.exe";
 std::string params=" -u mylogin --password=mypass --verbose --extended-insert=FALSE
databaseName > ";
 std::string pathOfInputFile="C:\\TEMP\\file1.sql";

      // some code to set values for paths and solves the filename space problem
 ss << "\"";                        // command opening quote
 ss << "\"" << pathOfCommand   << "\" "; // Quoted binary (could have spaces)
 ss << " " << params << " ";    // Quoted input (could have spaces)
 ss << "\"" << pathOfInputFile << "\"";  // Quoted input (could have spaces)
 ss << "\"";                        // command closing quote

      Cout() << "seeing how the command turned out:" << ss.str() << EOL;

      int i;
      i=system( ss.str().c_str() );
      Cout() <<"backup result: " << i << EOL;

}
```

Subject: Re: evaluate command line
Posted by jjacksonRIAB on Fri, 22 Jul 2022 02:19:37 GMT
View Forum Message <> Reply to Message

Try the Sys method instead, you don't need any of that std::string stuff, just String works fine:

```
#include <Core/Core.h>

using namespace Upp;
```

```
CONSOLE_APP_MAIN {
    String output;
    Sys("ls -la", output);

    Vector<String> lines = Split(output, "\n");

    int i = 1;
    for(auto& line : lines) {
        Cout() << Format("Line #%d: %s\n", i, line);
        i++;
    }
}
```

Sys will run the command and store the results in the output variable. Split will create an array of Strings that you can loop through on newline or you can choose an alternative delimiter. If you need anything more complex than that try looking up CParser (https://www.ultimatepp.org/src$Core$CParser_en-us.html), it can help you write a hand parser. Regexps are also available (https://www.ultimatepp.org/reference$RegExp$en-us.html).

I don't have Windows installed but if you replace "ls -la" with "dir" that example should get you going.

Subject: Re: evaluate command line
Posted by BetoValle on Fri, 22 Jul 2022 13:13:29 GMT
View Forum Message <> Reply to Message

Hi, thanks but

I think there must be some additional parameter in the case of windows 10 (my case). Wouldn't you have to pass the folder? I did a quick test with only sys("dir",output) following your example and it didn't return anything!

Subject: Re: evaluate command line
Posted by BetoValle on Fri, 22 Jul 2022 16:21:55 GMT
View Forum Message <> Reply to Message

Hi, again ...

with the command "dir" in windows 10 and I really couldn't!

I tested with the parameters below and finally I was successful!

important to point out that I had to modify a parameter in the backup command: replace the ">" with "--result-file="

With ">" Sys was not interpreting correctly and displayed message charging "a table".

I understand that the Sys routine is welcome for the purpose of creating a log file. On the other hand, creating a progress bar probably won't work given the processing delay.

```
    String output;

    String fdump="C:\\Program Files (x86)\\MariaDB 10.1\\bin\\mysqldump.exe";

    String xparams="  -u mylogin --password=mypass--verbose --extended-insert=FALSE
databaseName";

    String tofile=" --result-file=C:\\TEMP\\file1.sql";

    String s;

      s << fdump;

      s << xparams;

      s << tofile;

    Sys(s,output);

    Vector<String> lines = Split(output, "\n");

    String x;

    int i = 1;

    for(auto& line : lines) {

        x << Format("Line #%d: %s\n", i, line);

        i++;

    }

    Led.SetData(x); // Led is LineEdit!
```

Thanks jjackson!

Subject: Re: evaluate command line
Posted by jjacksonRIAB on Fri, 22 Jul 2022 19:48:52 GMT
View Forum Message <> Reply to Message

Yeah dir might not work if you're in an empty directory or the delimiter is not a newline (I think it is even under Windows though). There's a function called

ChangeCurrentDirectory("whatever");

There are also several functions for reading in environment variables that could prove useful to you in certain contexts. Ex.

Cout() << GetEnv("PATH");

Will print out a PATH environment variable in a unix-like or DOS. There's also GetHomeDirectory(), GetProgramsFolder(), GetProgramsFolderX86(), GetTempDirectory() and a few others - I believe all of those call GetEnv with known environment variables.

I don't know anything about MariaDB but it may be preferable to find some kind of environment variable or registry key to locate where mysqldump.exe so your don't have to hardcode anything (knowing Windows it could be installed in one of several places or even on a different drive). Sometimes that can't be helped though... and if it's just for personal use who cares, I guess... but if, for example, someone put their Programs Folder on the D: or E: drive instead of C:, some of the functions above can help you clean up your application discovery process.

EDIT Looking at Windows, it appears to do things differently. If I use "cmd /c dir" it will print out and split everything correctly but you might want LocalProcess instead. I don't know offhand how to create this process without creating a Window because I just remembered that dir is not actually a separate program under Windows, it's a built-in shell command. There's probably some argument that be passed to hide the window.

https://www.ultimatepp.org/src$Core$AProcess$en-us.html

Subject: Re: evaluate command line
Posted by BetoValle on Fri, 22 Jul 2022 20:51:14 GMT
View Forum Message <> Reply to Message

the only way that i found to use "dir" is below
but disadvantage is that the console window opens

```
const char* cmd;

cmd="dir C:\\temp";

std::shared_ptr<FILE> pipe(popen(cmd, "r"), pclose);

char buffer[128];

std::string result = "";

while (!feof(pipe.get())) {

    if (fgets(buffer, 128, pipe.get()) != NULL)

        result += buffer;

}
```

Subject: Re: evaluate command line
Posted by jjacksonRIAB on Fri, 22 Jul 2022 23:09:48 GMT
View Forum Message <> Reply to Message

I don't get an open console window if it's a GUI app and I'm using GUI_APP_MAIN - only if it's a console app:

```
#include "TestDir.h"

TestDir::TestDir()
{
    CtrlLayout(*this, "Window title");

    String output;
    Sys("cmd.exe /c dir c:\\", output);

    Vector<String> lines = Split(output, "\n");

    String numberedOutput;

    int i = 1;
    for(auto& line : lines) {
        numberedOutput << Format("Line #%d: %s\n", i, line);
        i++;
    }
```

```
    lineEdit <<= numberedOutput;
}

GUI_APP_MAIN
{
    TestDir().Run();
}
```

Subject: Re: evaluate command line
Posted by BetoValle on Fri, 22 Jul 2022 23:51:44 GMT
View Forum Message <> Reply to Message

ok! very good! now work with command "dir".

"Sys" on windows 10, no console window opens!

```
CONSOLE_APP_MAIN
{
    String output;
    Sys("cmd.exe /c dir c:\\temp", output);
    Cout() << "aqui " << output << EOL;
    Vector<String> lines = Split(output, "\n");

    int i = 1;
    for(auto& line : lines) {
        Cout() << Format("Line #%d: %s\n", i, line);
        i++;
    }
}
```
