Class CommandLine



Propose: C++ class to handle data from the command line.

Version 2.0.0

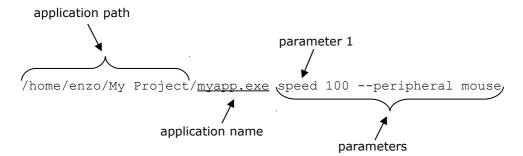
Enzo Roberto Verlato - enzover@ig.com.br https://github.com/FreeSource

Supported and tested platforms:

0.S.	Compiler	Make
WindowsXP SP2	MinGW gcc 4.6.1	gmake 3.82
Linux openSUSE 11.4 / 12.2	gcc 4.5.1 / 4.7.1	gmake 3.82
OpenIndiana 151a	gcc 3.4.3	gmake 3.81
FreeBSD 9.0	gcc 4.2.1	gmake 3.82
Solaris 9 / 10	gcc 3.3.2 / 3.4.6	gmake 3.80 / 3.81
Mac OS X 10.8.2	gcc 4.2.1	gmake 3.81

Copyright (c) 2012 Enzo Roberto Verlato. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

The standard structure of a command line:



Members:

```
string getApplicationName()
string getCurrentWorkingDirectory()

int getParametersNumber()
string getParameter( int parameterPosition )

setOptionPrefix( string optionPrefix )
setOptionPostfix( string optionPostfix )

bool hasOption( string option )
string getOptionValue( string option )
string getOptionLongValue( string option )
optionCaseSensitive()
optionCaseInsensitive()
```

string getApplicationName() 🖍

Description: Retrieves the name of the application for the current process.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main( int argc, char *argv[] ) {
 8
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
14
       try {
15
           environs::CommandLine commandLine;
           cout << commandLine.getApplicationName() << endl;</pre>
16
17
           return EXIT SUCCESS;
18
       } catch ( runtime error &error ) {
19
           cout << "Exception occurred: " << error.what() << endl;</pre>
           return EXIT FAILURE;
21
       }
23 }
24
```

Output:

```
linux:/home/enzo # ./myapp
myapp
```

Description: Retrieving the application path of the current process, not including the name of the program itself.

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
       try {
14
           environs::CommandLine commandLine;
15
           cout << commandLine.getApplicationPath() << endl;</pre>
16
           return EXIT SUCCESS;
17
18
19
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
           return EXIT FAILURE;
21
22
       }
```

```
23 }
24
```

```
linux:/home/enzo # ./myapp
/home/enzo
```

string getCurrentWorkingDirectory() \understand

Description: Retrieves the current working directory for the current process.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
14
       try {
15
           environs::CommandLine commandLine;
16
           cout << commandLine.getApplicationPath() << endl;</pre>
17
           cout << commandLine.getCurrentWorkingDirectory() << endl;</pre>
           return EXIT SUCCESS;
18
19
20
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
21
           return EXIT FAILURE;
       }
24 }
```

Output:

```
linux-hevv:/home/enzo/CommandLine/main # /home/enzo/myapp
/home/enzo/CommandLine/main
/home/enzo/CommandLine/main
```

int getParametersNumber() \understand

Description: Returns the total number of parameters on the command line for the current process, not including the name of the program itself.

```
#include <CommandLine.h>

#include <iostream>
#include <cstdlib>
#include <stdexcept>

int main() {

using std::cout;
```

```
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
14
       try {
           environs::CommandLine commandLine;
15
16
           cout << commandLine.getParametersNumber() << endl;</pre>
17
           return EXIT_SUCCESS;
18
       } catch ( runtime error &error ) {
19
           cout << "Exception occurred: " << error.what() << endl;</pre>
           return EXIT FAILURE;
21
       }
23 }
24
```

```
linux:/home/enzo # ./myapp The C++ Programming Language
4
```

string getParameter(int parameterPosition) 🛊

Description: Retrieves the parameter of the specified parameter position on the command line for the current process.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
       using std::cout;
       using std::endl;
10
       using std::string;
11
       using std::runtime error;
13
14
       try {
15
           environs::CommandLine commandLine;
16
           cout << commandLine.getParameter( 2 ) << endl;</pre>
17
           return EXIT SUCCESS;
18
19
       } catch ( runtime_error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
20
21
           return EXIT_FAILURE;
       }
23 }
24
```

Output:

```
linux:/home/enzo # ./myapp C++ evolved from C
evolved
```

setOptionPrefix(string optionPrefix) \underset

Description: Define the prefix (string added in front of the option name) used to recognize an option on the command line.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
14
      try {
15
           environs::CommandLine commandLine;
           commandLine.setOptionPrefix( "--");
16
17
           cout << commandLine.getOptionValue( "price" ) << endl;</pre>
           return EXIT SUCCESS;
18
19
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
21
           return EXIT FAILURE;
       }
24 }
25
```

Output:

```
linux:/home/enzo # ./myapp --price 0.99
0.99
```

setOptionPostfix(string optionPostfix) \underset

Description: Define the postfix (string added to the end of the option name) used to recognize an option on the command line.

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
       using std::cout;
10
      using std::endl;
11
       using std::string;
       using std::runtime error;
13
14
           environs::CommandLine commandLine;
15
           commandLine.setOptionPostfix( "=" );
16
           cout << commandLine.getOptionValue( "price" ) << endl;</pre>
17
           return EXIT SUCCESS;
18
19
       } catch ( runtime_error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
21
```

```
linux:/home/enzo # ./myapp price=0.99
0.99
```

bool hasOption(string option) \spadesuit

Description: Checks if the specified option exists.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
12
       using std::runtime error;
13
14
       try {
15
           environs::CommandLine commandLine;
           commandLine.setOptionPostfix( ":" );
16
17
           if( commandLine.hasOption( "ISBN-10" ) ) {
               cout << "yes" << endl;
18
           } else {
19
                cout << "no" << endl;</pre>
           }
           return EXIT SUCCESS;
23
       } catch ( runtime error &error ) {
24
           cout << "Exception occurred: " << error.what() << endl;</pre>
25
           return EXIT FAILURE;
26
       }
27
28 }
29
```

Output:

```
linux:/home/enzo # ./myapp Paperback: 208 pages Publisher: O'Reilly
Media; 1 edition (August 19, 2011) Language: English ISBN-10: 1449397670
Weight: 14.4 ounces
yes
```

string getOptionValue(string option) 🏚

Description: Retrieves the value of the specified option on the command line for the current process.

```
#include <CommandLine.h>
#include <iostream>
```

```
4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
      using std::cout;
10
      using std::endl;
11
      using std::string;
12
      using std::runtime error;
13
14
      try {
           environs::CommandLine commandLine;
15
           commandLine.setOptionPostfix( ":" );
16
           cout << commandLine.getOptionValue( "Language" ) << endl;</pre>
17
18
           return EXIT_SUCCESS;
19
20
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
21
           return EXIT FAILURE;
       }
23
24 }
```

```
linux:/home/enzo # ./myapp Paperback: 208 pages Publisher: O'Reilly
Media; 1 edition (August 19, 2011) Language: English ISBN-10: 1449397670
Weight: 14.4 ounces
English
```

string getOptionLongValue(string option) \(\bar{\lambda}\)

Description: Retrieves the long value of the specified option (a range of parameters delimited by the next option if it exists) on the command line for the current process.

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 7 int main() {
 8
 9
      using std::cout;
10
      using std::endl;
      using std::string;
12
      using std::runtime error;
13
14
           environs::CommandLine commandLine;
15
           commandLine.setOptionPrefix( "--" );
16
17
           cout << commandLine.getOptionLongValue( "peripheral" );</pre>
18
           cout << endl;</pre>
           return EXIT SUCCESS;
19
21
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
23
           return EXIT FAILURE;
       }
24
25 }
```

```
linux:/home/enzo # ./myapp --speed 100 --peripheral mouse display
keyboard --price 1000
mouse display keyboard
```

optionCaseSensitive() •

Description: Differ use of uppercase and lowercase letters on the option parameter for the other functions. Option parameter is case sensitive by default.

Example:

```
1 #include <CommandLine.h>
 3 #include <iostream>
 4 #include <cstdlib>
 5 #include <stdexcept>
 6
 7 int main() {
 8
 9
       using std::cout;
10
       using std::endl;
11
       using std::string;
       using std::runtime error;
13
14
       try {
15
           environs::CommandLine commandLine;
           commandLine.optionCaseSensitive();
16
           commandLine.setOptionPrefix( "--" );
17
           cout << commandLine.getOptionLongValue( "PERIPHERAL" );</pre>
18
19
           cout << endl;</pre>
           return EXIT SUCCESS;
       } catch ( runtime error &error ) {
           cout << "Exception occurred: " << error.what() << endl;</pre>
           return EXIT FAILURE;
24
       }
26 }
27
```

Output:

```
linux:/home/enzo # ./myapp --peripheral mouse
```

optionCaseInsensitive() •

Description: No differ use of uppercase and lowercase letters on the option parameter for the other functions.

```
1 #include <CommandLine.h>
3 #include <iostream>
4 #include <cstdlib>
5 #include <stdexcept>
```

```
7 int main() {
9
      using std::cout;
10
      using std::endl;
11
      using std::string;
      using std::runtime_error;
13
14
      try {
15
          environs::CommandLine commandLine;
          commandLine.optionCaseInsensitive();
16
          commandLine.setOptionPrefix( "--");
17
          cout << commandLine.getOptionLongValue( "PERIPHERAL" );</pre>
18
           cout << endl;</pre>
19
          return EXIT SUCCESS;
21
22
       } catch ( runtime_error &error ) {
23
          cout << "Exception occurred: " << error.what() << endl;</pre>
           return EXIT_FAILURE;
24
       }
25
26 }
27
```

```
linux:/home/enzo # ./myapp --peripheral mouse
mouse
```