

CommandLineParser Reference Manual

1.0

Generated by Doxygen 1.3.2

Mon Aug 30 10:50:58 2004

Contents

1	CommandLineParser Compound Index	1
1.1	CommandLineParser Compound List	1
2	CommandLineParser Class Documentation	3
2.1	CommandLineParser Class Reference	3

Chapter 1

CommandLineParser Compound Index

1.1 CommandLineParser Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

CommandLineParser (This class provides an easy-to-use command line argument parser)	3
---	---

Chapter 2

CommandLineParser Class Documentation

2.1 CommandLineParser Class Reference

This class provides an easy-to-use command line argument parser.

```
#include <CommandLineParser.h>
```

Public Member Functions

- **CommandLineParser** (string description_text, int width_tokenfield=30)\
- virtual **~CommandLineParser** ()
- void **addParameter** (string token, string helptext, string defaultvalue=string())
method for adding a parameter.
- void **parse** (int argc, char **argv) throw (string)
method for parsing the command line arguments. Throws a help string if the -h switch has been activated.
- string **getArgument** (string parameter) throw (string)
method for retrieving a string argument. Throws a string exception if a non-optional parameter has not being set.
- bool **isSet** (string parameter)
- long int **getLongIntArgument** (string parameter, int base=0) throw (string)
method for retrieving an integer argument. Throws a string exception if a non-optional parameter has not been set or when the conversion to a number value fails.
- double **getDoubleArgument** (string parameter) throw (string)
method for retrieving a floating-point argument. Throws a string exception if a non-optional parameter has not been set or when the conversion to a number value fails.

2.1.1 Detailed Description

This class provides an easy-to-use command line argument parser.

Example program:

```
#include <iostream>
#include <new>
#include <string>
#include "CommandLineParser.h"

using namespace std;

int main(int argc, char **argv)
{
    try{
        CommandLineParser parser("CommandLineParser class example program.");
        parser.addParameter("-name","optional string parameter","MyName");
        parser.addParameter("I","required integer parameter");
        parser.addParameter("--OptionalInteger","optional integer parameter","1");
        parser.addParameter("D","required double parameter");
        parser.addParameter("S","command line switch");
        parser.parse(argc,argv);
        cout << "--name argument: " << parser.getArgument("-name") << endl
              << "-I (integer) : " << parser.getLongIntArgument("I") << endl
              << "--OptionalInteger : "
              << parser.getLongIntArgument("--OptionalInteger") << endl
              << "-D (double) : " << parser.getDoubleArgument("D") << endl
              << "switch S "
              << ((parser.isSet("S")) ? "has" : "hasn't")
              << " been set." << endl;
    }
    catch(string msg)
    {
        cerr << msg << endl;
        return -1;
    }
    catch(bad_alloc e)
    {
        cerr << "Memory allocation failed!" << endl
              << e.what() << endl;
        return -2;
    }
    catch(...)
    {
        cerr << "An unknown exception occurred!" << endl;
        return -3;
    }
    return 0;
}
```

When calling the program with the -h switch it prints out:

```
CommandLineParser class example program.
Command line switches:
-h                - help
--name            - optional string parameter (default value: MyName )
-I               - required integer parameter
--OptionalInteger - optional integer parameter (default value: 1.0 )
-D               - required double parameter
-S               - command line switch
```

Calling it with


```
./CommandLineParser -I 2 -D 2.1 -S
--name argument: MyName
-I (integer) : 2
--OptionalInteger : 1
-D (double) : 2.1
switch S has been set.
```

Author:

Ruediger Knoerig

2.1.2 Constructor & Destructor Documentation

2.1.2.1 CommandLineParser::CommandLineParser (string *description_text*, int *width_tokenfield* = 30) [inline]

default constructor

Parameters:

description_text text describing the function of the program

width_tokenfield width of the token field in the help text

2.1.2.2 virtual CommandLineParser::~~CommandLineParser () [inline, virtual]

destructor

2.1.3 Member Function Documentation

2.1.3.1 void CommandLineParser::addParameter (string *token*, string *helptext*, string *defaultvalue* = string())

method for adding a parameter.

Parameters:

token the parameter has the command line switch -token

helptext helptext describing the function of the switch

default default value for the parameter. Omit for required parameters; for optional parameters enter at least some nonsense

2.1.3.2 string CommandLineParser::getArgument (string *parameter*) throw (string)

method for retrieving a string argument. Throws a string exception if a non-optional parameter has not being set.

Parameters:

parameter token of the parameter to retrieve

Returns:

assigned argument for the parameter

2.1.3.3 double CommandLineParser::getDoubleArgument (string *parameter*) throw (string)

method for retrieving a floating-point argument. Throws a string exception if a non-optional parameter has not been set or when the conversion to a number value fails.

Parameters:

parameter token of the parameter to retrieve

Returns:

assigned argument for the parameter

2.1.3.4 long int CommandLineParser::getLongIntArgument (string *parameter*, int *base* = 0) throw (string)

method for retrieving an integer argument. Throws a string exception if a non-optional parameter has not been set or when the conversion to a number value fails.

Parameters:

parameter token of the parameter to retrieve

base base of the number system (must be between 2 and 36, for default: 10)

Returns:

assigned argument for the parameter

2.1.3.5 bool CommandLineParser::isSet (string *parameter*) [inline]

method returning true if an parameter has been set

Parameters:

parameter token of the parameter to retrieve

Returns:

true if the command line switch was in use

2.1.3.6 void CommandLineParser::parse (int *argc*, char ** *argv*) throw (string)

method for parsing the command line arguments. Throws a help string if the -h switch has been activated.

Parameters:

argc number of strings in the array argv

argv Array of C-strings with the command line parameters

The documentation for this class was generated from the following files:

- CommandLineParser.h
- CommandLineParser.cpp

Index

- ~CommandLineParser
 - CommandLineParser, 5
- addParameter
 - CommandLineParser, 5
- CommandLineParser, 3
 - CommandLineParser, 5
- CommandLineParser
 - ~CommandLineParser, 5
 - addParameter, 5
 - CommandLineParser, 5
 - getArgument, 5
 - getDoubleArgument, 5
 - getLongIntArgument, 6
 - isSet, 6
 - parse, 6
- getArgument
 - CommandLineParser, 5
- getDoubleArgument
 - CommandLineParser, 5
- getLongIntArgument
 - CommandLineParser, 6
- isSet
 - CommandLineParser, 6
- parse
 - CommandLineParser, 6