

ost::CommandOption(3)

ost::CommandOption(3)

NAME

ost::CommandOption –

CommandOption is the base class for all command line options.**SYNOPSIS**

#include <cmdoptns.h>

Inherited by **ost::CommandOptionNoArg**, and **ost::CommandOptionWithArg**.**Public Types**enum **OptionType** { **hasArg**, **noArg**, **trailing**, **collect** }*OptionType is for denoting what type of option this is, with an arg, without an arg or the trailing args.***Public Member Functions**virtual ~**CommandOption** ()*A virtual destructor just in case.***CommandOption** (const char *inOptionName, const char *inOptionLetter, const char *inDescription, **OptionType** inOptionType, bool inRequired=false, **CommandOption** **ppNext=&**defaultCommandOptionList**)*CommandOption contructor.*virtual void **foundOption** (**CommandOptionParse** *cop, const char *value=0)*foundOption is called by the **CommandOptionParse** object during the parsing of the command line options.*virtual void **foundOption** (**CommandOptionParse** *cop, const char **value, int num)*foundOption is called by the **CommandOptionParse** object during the parsing of the command line options.*virtual void **parseDone** (**CommandOptionParse** *cop)*Once parsing of command line options is complete, this method is called.*virtual void **performTask** (**CommandOptionParse** *cop)*Once **CommandOption** objects have completed parsing and there are no errors they may have some specific tasks to perform.*virtual bool **hasValue** ()*For fields with the required flag set, this method is used to determine if the Option has satisfied it's required status.***Public Attributes**const char * **optionName***Long option name, these will be preceded with '--' on the command line.*const char * **optionLetter***option letter, these will be preceded with '-' on the command line.*const char * **description***A short description of the option for Usage messages.***OptionType** **optionType***This command option's OptionType.*bool **required***True if this parameter is required.***CommandOption** * **next***This next **CommandOption** in this list of options or nil if no more options exist.***Detailed Description****CommandOption** is the base class for all command line options.

Command line options can be defined statically and used when constructing a command line parser object using makeCommandOptionParse. This serves only as a base class to

CommandOptionWithArg, **CommandOptionRest** or **CommandOptionNoArg** which can also be used to derive more complex classes or even entire applications.**Author:**

Gianni Mariani <gianni AT mariani DOT ws>

Examples:

ost::CommandOption(3)

ost::CommandOption(3)

cmdlineopt.cpp.**Member Enumeration Documentation****enum ost::CommandOption::OptionType**

OptionType is for denoting what type of option this is, with an arg, without an arg or the trailing args.
Option type

Enumerator:

- hasArg* This option is associated with a value.
- noArg* This option is a flag only.
- trailing* Remaining of the command line arguments.
- collect* Collect values that are not a value to an option.

Constructor & Destructor Documentation**virtual ost::CommandOption::~CommandOption () [virtual]**

A virtual destructor just in case.

ost::CommandOption::CommandOption (const char * inOptionName, const char * inOptionLetter, const char * inDescription, OptionType inOptionType, bool inRequired = false, CommandOption ** ppNext = &DefaultCommandOptionList)

CommandOption contructor. Note the default values for required and ppNext.

Parameters:

- inOptionName* long option name
- inOptionLetter* short letter name
- inDescription* short description of the option
- inOptionType* the type of this option
- inRequired* true if option is required
- ppNext* the linked list header

Member Function Documentation**virtual void ost::CommandOption::foundOption (CommandOptionParse * cop, const char ** value, int num) [virtual]**

foundOption is called by the **CommandOptionParse** object during the parsing of the command line options. **Parameters:**

- cop* pointer to the command option parser
- value* an array of values of this option
- num* number of values in the array

Reimplemented in **ost::CommandOptionWithArg**.

virtual void ost::CommandOption::foundOption (CommandOptionParse * cop, const char * value = 0) [virtual]

foundOption is called by the **CommandOptionParse** object during the parsing of the command line options. **Parameters:**

- cop* pointer to the command option parser
- value* the value of this option

Reimplemented in **ost::CommandOptionWithArg**, and **ost::CommandOptionNoArg**.

virtual bool ost::CommandOption::hasValue () [virtual]

For fields with the required flag set, this method is used to determine if the Option has satisfied it's required status. The default methods simply returns true if any values have been found. This could be specialized to return true based on some other criteria.

Reimplemented in **ost::CommandOptionWithArg**.

virtual void ost::CommandOption::parseDone (CommandOptionParse * cop) [virtual]

Once parsing of command line options is complete, this method is called. This can be used to perform last minute checks on the options collected.

Parameters:

- cop* pointer to the command option parser

Examples:

ost::CommandOption(3)

ost::CommandOption(3)

cmdlineopt.cpp.**virtual void ost::CommandOption::performTask (CommandOptionParse * cop) [virtual]**

Once **CommandOption** objects have completed parsing and there are no errors they may have some specific tasks to perform. PerformTask must return.

Parameters:

cop pointer to the command option parser

Examples:**cmdlineopt.cpp.****Member Data Documentation****const char* ost::CommandOption::description**

A short description of the option for Usage messages. e.g. Usage: mycommand : blah -f, --file <DESCRIPTION here>

CommandOption* ost::CommandOption::next

This next **CommandOption** in this list of options or nil if no more options exist.

const char* ost::CommandOption::optionLetter

option letter, these will be preceded with '-' on the command line. e.g. -f foo.x

const char* ost::CommandOption::optionName

Long option name, these will be preceded with '--' on the command line. e.g. --file foo.x

OptionType ost::CommandOption::optionType

This command option's OptionType.

bool ost::CommandOption::required

True if this parameter is required. If the parameter is not supplied and required is true, an error will be flagged in the option processor.

Author

Generated automatically by Doxygen for GNU CommonC++ from the source code.

