TCLAP:
Templatized C++ Command Line Parser Library

Kristi Potter
April 12, 2010

What is TCLAP?

Templated c++ command line parser

- simple interface
- templated argument list
- only header files

```
#include <tclap/CmdLine.h>
```

- automatically print out usage/help messages

```
$ ./testTCLAP --help

USAGE:
./testTCLAP  [-n <string>] [--] [--version] [-h]

Where:
  -n <string>, --name <string>  Name to print

  --, --ignore_rest  Ignores the rest of the labeled
  arguments following this flag.

  --version  Displays version information and exits.

  -h, --help Displays usage information and exits.

testTCLAP description message
```

Overview - classes

- `CmdLine:` manages the command line definition and passes parsing to appropriate arg classes
- `ValueArg, MultiArg, SwitchArg, MultiSwitchArg,UnlabeledValueArg, UnlabeledMultiArg:` templated argument classes, each argument contains its data after parsing

Example*

```
// Define the command line object
CmdLine cmd("Description message", ' ', "1.0");

// Define a value argument
ValueArg<string> nameArg("n","name","Name to print",
                         true,"Homer", "string");

// Add it to the command line object
cmd.add( nameArg );

// Parse the command line
cmd.parse( argc, argv );

// Get the value
string name = nameArg.getValue();
```

* must wrap in a try/catch

CmdLine Class

```
CmdLine ( const std::string &message,
          const char delimiter = ' ',
          const std::string &version = "none",
          bool helpAndVersion = true)
```

- `message:` displayed in usage output
- `delimiter:` separates argument flag from value
- `version:` displayed in --version
- `helpAndVersion:` create message switches

Argument Classes - Construction

```
TCLAP::Argument<T> ( const std::string & flag,
                     const std::string & name,
                      const std::string & desc,
                     bool   req,
                     T      value)
```

- `flag:` single character flag (-h)
- `name:` long argument name (--help)
- `desc:` description of arg displayed in help msg
- `req:` flag if the argument is required
- `value:` default value of the argument

Argument Classes - `ValueArg/MultiArg`

- read a value of the templated type
- returns a single value (`ValueArg`)

```
$ ./testTCLAP  -a value
```

- or a vector of values (`MultiArg`)

```
$ ./testTCLAP -a multival1 -a multival2
```

Argument Classes - `SwitchArg`

- on/off boolean switch
- doesn't parse a value
- return true/false depending on if the switch is found & the default value of the switch
- If the switch is set on the command line, then the getValue method will return the opposite of the default value for the switch.

Argument Classes - `MultiSwitchArg`

- switch that can be specified multiple times
- useful when command lines are constructed automatically from within other applications
- ex: -V means a little verbose
  -V -V -V means a lot verbose
- getValue returns the # of times the arg appears on the command line

Argument Classes - `UnlabeledValueArg`

- value arg that is not defined by a flag, but by it's position in the argv array

```
$ ./tclapCopyFiles  file1  file2
```

- the order in which these args are added to the command line object is the order in which they will be parsed

Argument Classes - `UnlabeledMultiArg`

- more than one value can be specified
- only one `UnlabeledMultiArg` per command line
- reads the values up until the end of argv
- must be the last value in the command line

Good to know...

- Combining switch args

```
$ command -a -b -c
```

or

```
$ command -abc
```

or

```
$ command -ba -c
```

Good to know...

- One arg, or the other, but not both

```
cmd.xorAdd(Arg& a, Arg& b);
```

or

```
cmd.xorAdd(vector<Arg*>xorList);
```

Good to know...

- Want flags that are longer than a single character (ie -f vs --flag)
  - force the long option (--flag) by leaving the first parameter blank

```
ValueArg<string> fileArg("","file","File name",
                         true,"val","filename");
```

Good to know...

- More options in manual...
  - constrain the values allowed for a particular argument
  - hex integers as arguments
  - ignore certain arguments
  - different output than what is provided.
  - args add themselves to the CmdLine
  - use different, non standard types like std::pair

Get It!

- http://tclap.sourceforge.net/
- licensed under the MIT License