Flags-Macro

> ***name: csl***

> ***E-Mail: 3079625093@qq.com***

```
   _|_|  _|
 _|     _|    _|_|_|    _|_|_|    _|_|_|     _|_|_|  _|_|     _|_|_|    _|_|_|  _|  _|_|    _|_|
_|_|_|_|  _|  _|    _|  _|    _|  _|_|         _|    _|    _|  _|    _|  _|       _|_|     _|    _|
 _|     _|  _|    _|  _|    _|     _|_|     _|    _|    _|  _|    _|  _|       _|       _|    _|
 _|     _|    _|_|_|    _|_|_|  _|_|_|    _|  _|    _|    _|_|_|    _|_|_|  _|         _|_|
                          _|
                        _|_|
```

OverRide

this is a simple 'program-command-line-parameter-parsing' library using cpp-macro.

Usage

```cpp
#include "flags.h"

/**
 * @brief define params
 */
DEFINE_STRING(name, "the name of the person", "");
DEFINE_DOUBLE(age, "the age of the person", 0.0);
DEFINE_BOOL(sex, "the sex of the person[female: 0, male:1]", 1);
DEFINE_STRING_VEC(likes, "the likes of the person", "");

int main(int argc, char const *argv[]) {
  /**
   * @brief you should use 'try-catch' struct
   */
  try {
    // set the version
    ns_flags::setup_version("2.0");
    // setup the flags
    ns_flags::setup_flags(argc, argv);

    // output the values
    INFO("here is the information of a person:");
```

```
    TEXT("name: ", ns_flags::name);
    TEXT("age: ", ns_flags::age);
    TEXT("sex: ", ns_flags::sex);
    TEXT("likes: ", ns_flags::likes, '\n');

    // output the setup string values
    INFO("the init param setup values:");
    ns_log::setFirSedName("option", "initVal");
    TEXT(ns_flags::get_param_setup_value(), '\n');

    // output the describes of the params
    INFO("the describes of params:");
    ns_log::setFirSedName("option", "desc");
    ns_log::setSplitor(",\n");
    TEXT(ns_flags::get_param_desc());

  } catch (const std::exception &e) {
    std::cerr << e.what() << '\n';
  }

  return 0;
}
```

output

right command line

```
./flags --name csl --age 21.0 --sex 1 --likes coding sleep play
```

```
[ info  ] [1644319759] here is the information of a person:
name: csl
age: 21
sex: 1
likes: [coding, sleep, play]

[ info  ] [1644319759] the init param setup values:
[{'option': likes, 'initVal': [coding, sleep, play]}, {'option': sex, 'initVal': [1]}, {'option':
age, 'initVal': [21.0]}, {'option': name, 'initVal': [csl]}]

[ info  ] [1644319759] the describes of params:
[{'option': likes, 'desc': the likes of the person},
{'option': sex, 'desc': the sex of the person[female: 0, male:1]},
{'option': name, 'desc': the name of the person},
{'option': age, 'desc': the age of the person},
{'option': help, 'desc': display the help docs[default: false]},
{'option': version, 'desc': the version of this program[default: 1.0]}]
```

if the command is wrong, the ***error*** info will outputed

```
./flags --nema csl
```

```
some error(s) happened in the command line:
[ error ] the option named '--nema' is invalid, use '--help' option for help.
```

Macroes

```
/**
 * @brief support param types
 * [int, bool, std::string, double]
 * std::vectror<[int, bool, std::string, double]>
 */
```

- ***DEFINE_INT(parName, desc, default)***
- ***DEFINE_BOOL(parName, desc, default)***
- ***DEFINE_STRING(parName, desc, default)***
- ***DEFINE_DOUBLE(parName, desc, default)***

```
/**
 * @brief define an '[]-type' param
 *
 * @param parName the name of the param
 * @param desc the describe of this param, used in the help docs
 * @param default the default value of the param
 */
```

- ***DEFINE_INT_VEC(vecName, desc, ...)***
- ***DEFINE_BOOL_VEC(vecName, desc, ...)***
- ***DEFINE_STRING_VEC(vecName, desc, ...)***
- ***DEFINE_DOUBLE_VEC(vecName, desc, ...)***

```
/**
 * @brief define a 'std::vector<[]>-type' param
 *
 * @param vecName the name of the vector
 * @param desc the describe of this vector, used in the help docs
 * @param __VA_ARGS__ the default elems of the vector
 */
```

Methods

- ***void setup_help(const std::string& helpStr)***

```
/**
 * @brief Set the string of the help docs
 * @attention if this method called, then the help docs will not generate
 * automatically
 * @param helpStr the help string
 */
```

- ***setup_version(const std::string& version)***

```
/**
 * @brief Set the version of this program
 *
 * @param version the version string [default: 1.0]
 */
```

- *get_param_setup_value()*

```
/**
 * @brief Get the param setup string value at the begining of the main function
 */
```

- *get_param_desc()*

```
/**
 * @brief Get the param describe that user defined
 */
```

- *setup_flags(int argc, char const* argv[])*

```
/**
 * @brief set up the flags according the params user defined
 *
 * @param argc the count of the arguments
 * @param argv the value of the arguments
 */
```