LIBRARY IN A WEEK

Jeff Garland

C++Now 2015

INTRO

WHAT IS THIS SESSION?

- Workshop
  - Learn by doing
  - Different topic every year
  - “Self Organizing”
- Platform for learning from peers
  - About C++, Boost Development, etc
- Community Building
  - Create contributors
  - Create connections

WAYS TO PARTICIPATE

- Please participate!
  - Session(s) are meant to be interactive
  - Shaped/run by the participants
- Things you can do
  - Research, Write Code, Present
  - Come to morning sessions and provide input

TOPIC – C++ APPLICATION CONFIGURATION

WHY APPLICATION CONFIG? - MOTIVATION

- Almost every program beyond hello world needs configuration
- Modern applications it’s quite complex
- Current libraries have limitations
- Library that can use C++14 features
- Votes from last year’s session
  - property tree rewrite
  - program options – better
- Code (next page)!

AN EXAMPLE –

```
int main(int argc, const char **argv)
{
    std::vector<std::string> jpaths;
    jpaths.emplace_back("/usr/share/" XXXX_VERSION "/");
    jpaths.emplace_back("/usr/local/share/" XXX_VERSION "/");

    auto args = simplify_args(argc, argv);
    std::vector<std::string> remaining_args;

    for (unsigned i=0 ; i<args.size() ; ++i) {
        const std::string &arg = args[i];
        if (arg == "-h" || arg == "--help") {
```

…385 lines later args are processed…

DIMENSIONS OF THE PROBLEM

- Command line
  - myprogram –a foo –time 2015-05-12 00:00:00 foo.txt bar.txt
  - Names of parameters - short aliases
  - Positional and named parms
  - Multi-entry params
- Environment
  - Typical – path information
  - Often same as command line
- Coded default values
- Files
  - ini, json, xml formats
  - Content often needs to merge/override environment and command line

DIMENSIONS OF THE PROBLEM - 2

- Processing an option
  - Convert from text to user defined c++ type
  - Check it's valid
  - Enum example

```
enum foo { bar, baz };
vector<string> foo_allowed_values = { “bar”, “baz”};
…convert bar to enum value bar….
…if not bar or baz – error….
```

DIMENSIONS OF THE PROBLEM - ADVANCED

- Generative properties / References
- Example

```
FOO = “bar”
BAZ = ${FOO}
BAR = ${var_in_hello}
subConfig = hello.ini    //has var_in_hello = …
```

- Implications – evaluation has to be delayed
- jsonnet
  - dsl with parser
  - Turing complete language for generating config info into json (and ini)
  - http://google.github.io/jsonnet/doc/

```
// Jsonnet Example
{
   person1: {
      name: "Alice",
      welcome: "Hello " + self.name + "!",
   },
   person2: self.person1 { name: "Bob" },
}
```

```
{
   "person1": {
      "name": "Alice",
      "welcome": "Hello Alice!"
   },
   "person2": {
      "name": "Bob",
      "welcome": "Hello Bob!"
   }
}
```

PROGRAM OPTIONS

- http://www.boost.org/doc/libs/1_58_0/doc/html/program_options.html
- Does
  - Merging of Environment, command line, ini files, default values
- Doesn't
  - Handle json or xml
- Users of Program Options?
- docopt

PROPERTY TREE

- http://www.boost.org/doc/libs/1_58_0/doc/html/property_tree.html
- Does
  - Parsing/Saving of xml, json, ini files
- Doesn’t
  - Handle command line and other otpions
- Users of Program Options?

GOALS OF DESIGN

- Handle files, environment, command line – merging results
- Easy specification of error information, usage
- Easy specification of input checking
- File formats xml, json, ini
- Design of the Library
  - Simple to use
  - C++11 ++
  - Header only
  - Legacy custom source
  - Yaml
  - Not a lot of dependencies…
  - Output what the program computed for the options….
  - Round trip…
  - Composibility – build from smaller sets of options…
  - Updatte notiification if the file has changed….
  - Optiion confflicts…dependencies…positive and negative mutually exculsive, and then dependent….

APPROXIMATE PLAN

- day 1: Get Organized
  - Selection of focus sections to attack
  - Assignments and teaming
- day 2: Initial Presentations
  - Initial solutions
  - Tool selections
- day 3 & 4: More solution presentations
- day 5: Wrap up – future directions

TOOLS & RESOURCE

- Other Libraries
- C++11 is language of choice
- Github Repository
  - git clone https://github.com/JeffGarland/liaw2015.git
- Mailing List

http://mail-lists.crystalclearsoftware.com/listinfo.cgi/liaw2015-crystalclearsoftware.com

NEXT STEPS

- Meet for lunch – outside Flug
- Presentations for Tues
- Tuesday - Adam Getchell on DocOpt
- Tuesday – Boris – dsl based system + libstudxml