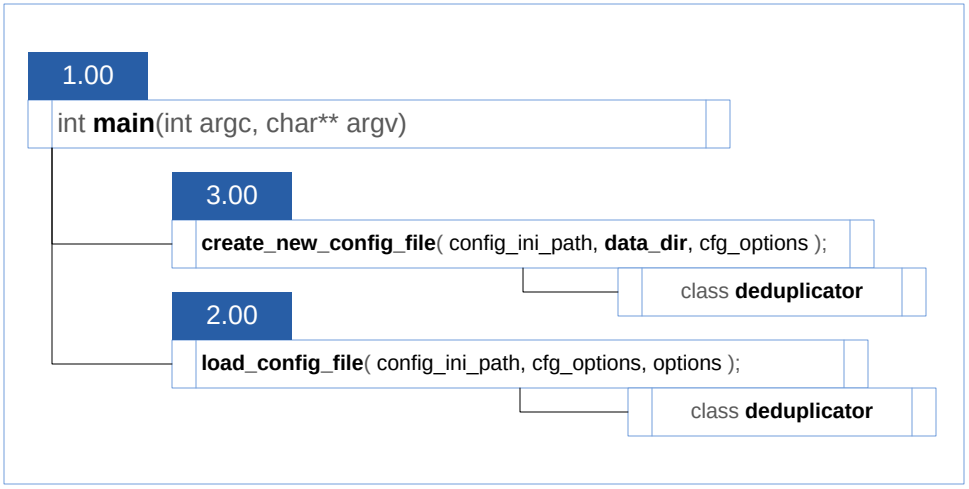
witness_node starting functions structures and the code flow



```
using namespace graphene;
```

```
namespace bpo = boost::program_options;
```

```
void write_default_logging_config_to_stream(std::ostream& out);
```

```
fc::optional<fc::logging_config> load_logging_config_from_ini_file(const fc::path& config_ini_filename);
```

class **deduplicator**

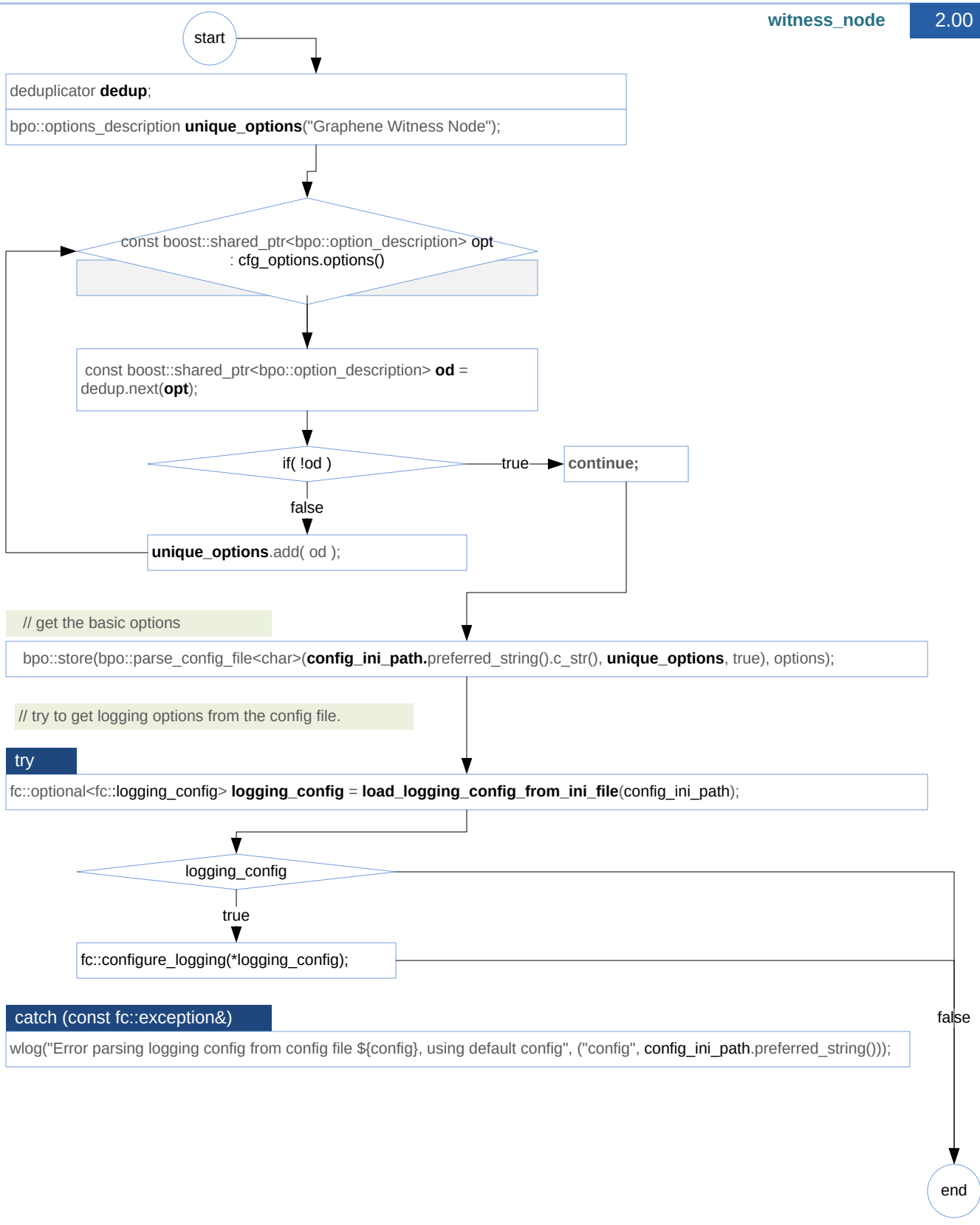
4.00

```
public:
   deduplicator() : modifier(nullptr) {}

   deduplicator(const boost::shared_ptr<bpo::option_description> (*mod_fn)(const
boost::shared_ptr<bpo::option_description>&))
         : modifier(mod_fn) {}

   const boost::shared_ptr<bpo::option_description> next(const boost::shared_ptr<bpo::option_description>& o)
   {
      const std::string name = o->long_name();
      if( seen.find( name ) != seen.end() )
         return nullptr;
      seen.insert(name);
      return modifier ? modifier(o) : o;
   }
```

```
private:
   boost::container::flat_set<std::string> seen;
   const boost::shared_ptr<bpo::option_description> (*modifier)(const boost::shared_ptr<bpo::option_description>&);
```

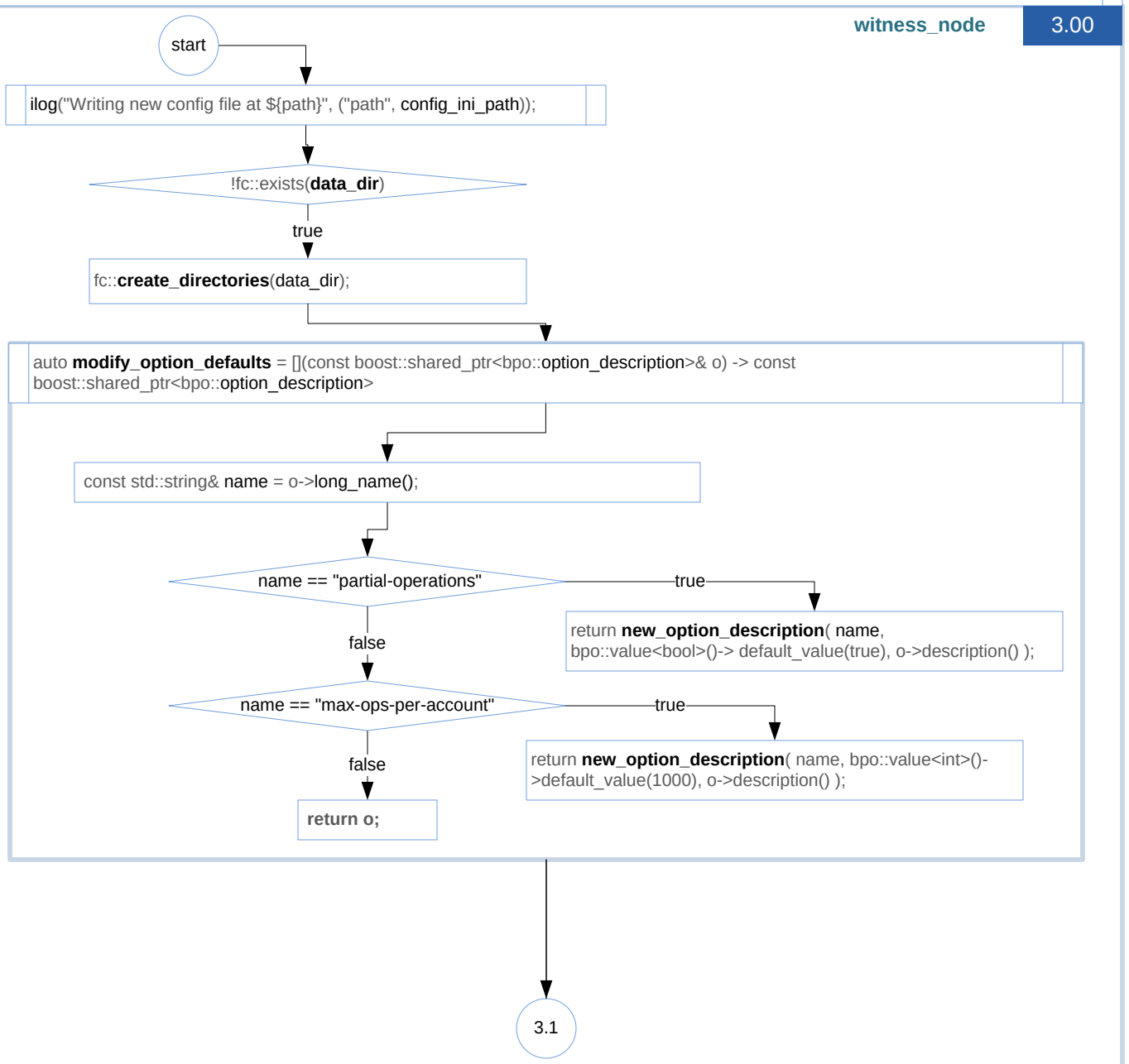
static void **load_config_file**(const fc::path& **config_ini_path**, const bpo::options_description& cfg_options, bpo::variables_map& options)

witness_node 2.00



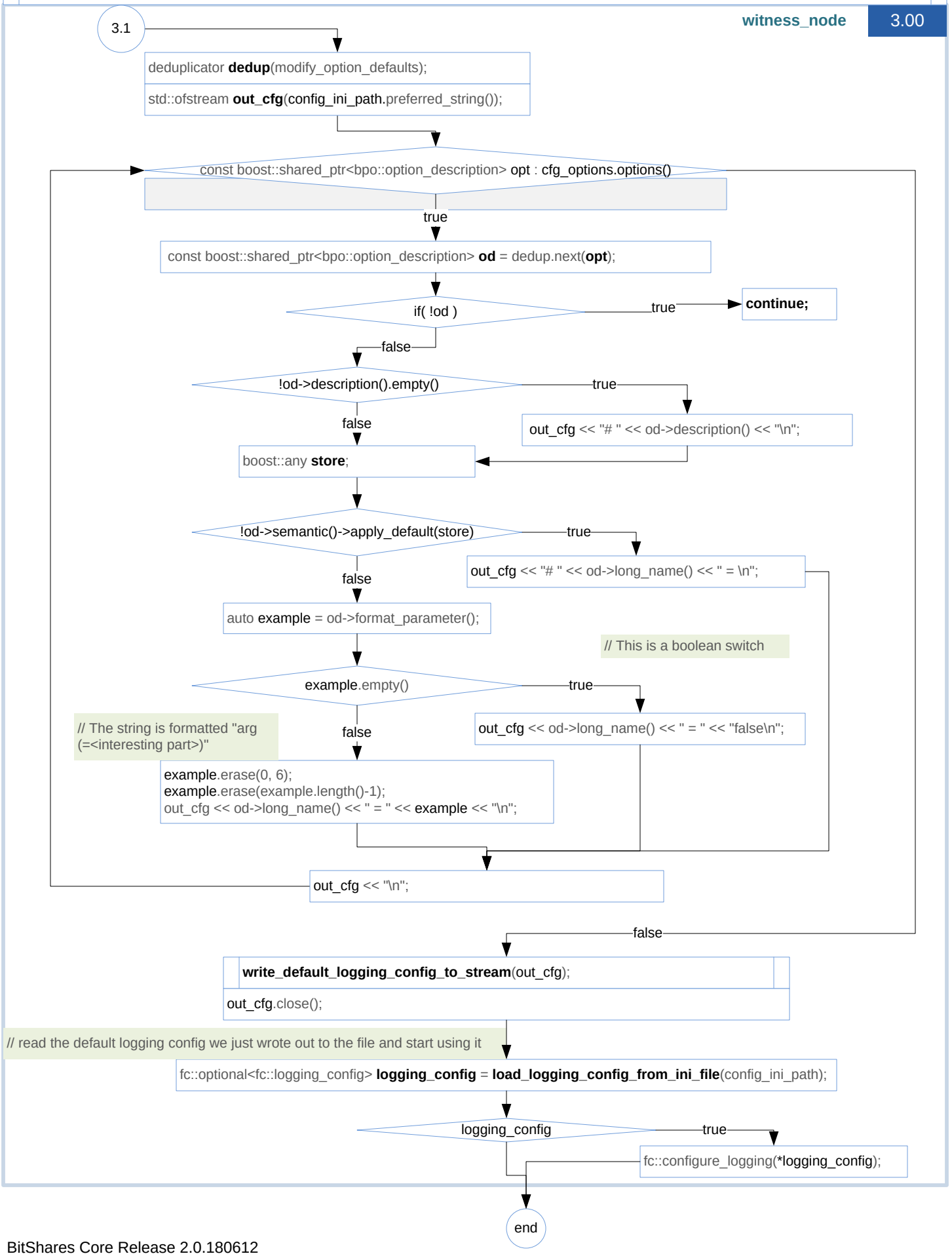
static void **create_new_config_file**(const fc::path& **config_ini_path**, const fc::path& **data_dir**, const bpo::options_description& **cfg_options**)

witness_node 3.00



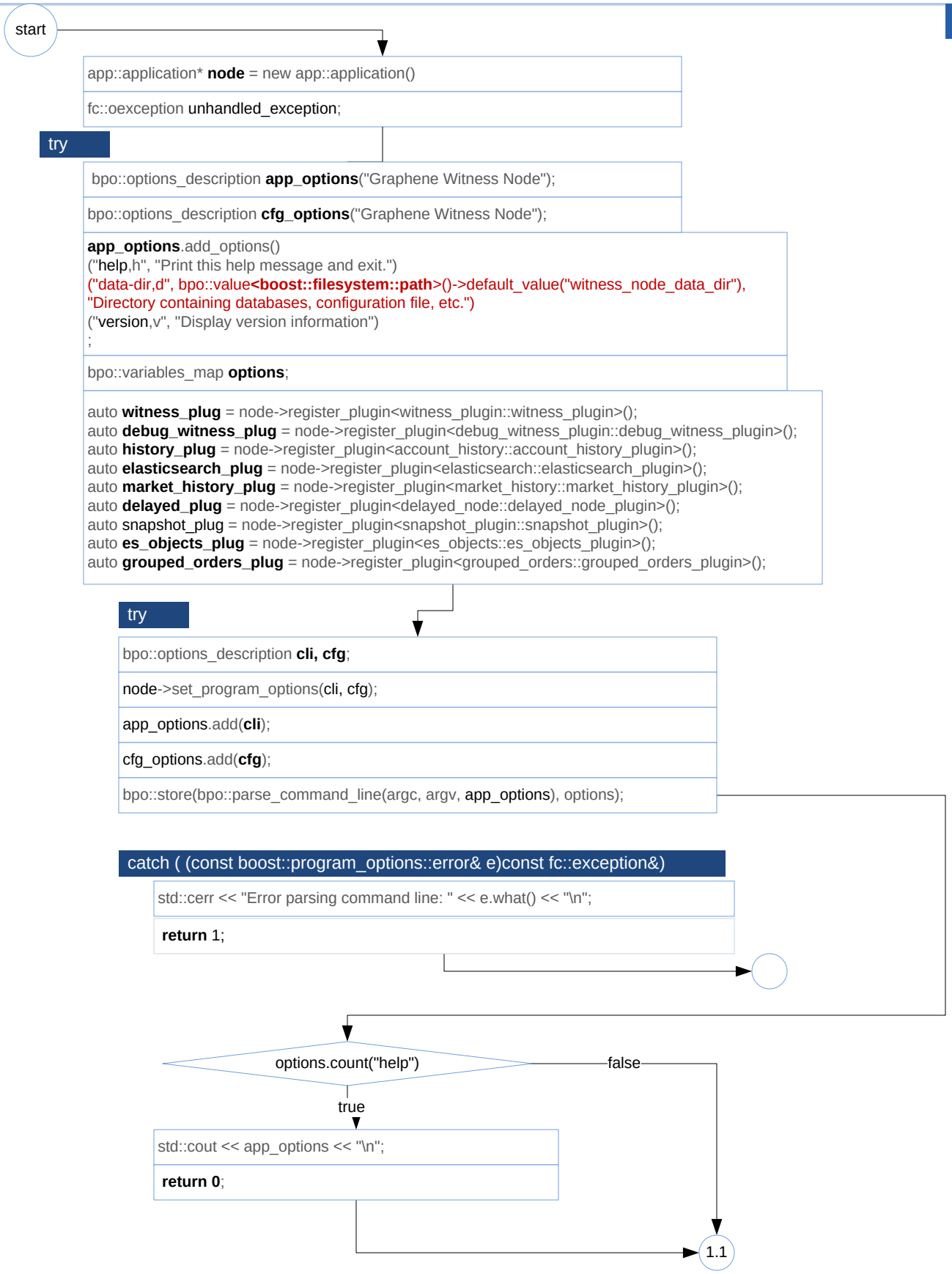
static void **create_new_config_file**(const fc::path& **config_ini_path**, const fc::path& **data_dir**, const bpo::options_description& **cfg_options**)

witness_node 3.00



int **main**(int argc, char** argv) witness_node

1.00



int **main**(int argc, char** argv) witness_node

1.00

1.1

options.count("version")

true

```
std::cout << "Version: " << graphene::utilities::git_revision_description << "\n";
std::cout << "SHA: " << graphene::utilities::git_revision_sha << "\n";
std::cout << "Timestamp: " << fc::get_approximate_relative_time_string(fc::time_point_sec(graphene::utilities::git_revision_unix_timestamp))
<< "\n";
std::cout << "SSL: " << OPENSSL_VERSION_TEXT << "\n";
std::cout << "Boost: " << boost::replace_all_copy(std::string(BOOST_LIB_VERSION), "_", ".") << "\n";
std::cout << "Websocket++: " << websocketpp::major_version << "." << websocketpp::minor_version << "." << websocketpp::patch_version
<< "\n";
```

return 0;

false

```
fc::path data_dir;
```

options.count("data-dir")

true

```
data_dir = options["data-dir"].as<boost::filesystem::path>();
```

false

data_dir.is_relative()

true

```
data_dir = fc::current_path() / data_dir;
```

false

```
fc::path config_ini_path = data_dir / "config.ini";
```

!fc::exists(**config_ini_path**)

true

3.00

```
create_new_config_file( config_ini_path, data_dir, cfg_options );
```

false

2.00

```
load_config_file( config_ini_path, cfg_options, options );
```

```
bpo::notify(options);
node->initialize(data_dir, options);
node->initialize_plugins( options );

node->startup();
node->startup_plugins();
```

```
fc::promise<int>::ptr exit_promise = new fc::promise<int>("UNIX Signal Handler");
```

1.2

int **main**(int argc, char** argv) witness_node

1.00