Using util::ConfigFile

- One must include the util_config_file.h header file and then write
  - util::ConfigFile settings;
  - ...
  - settings.load("my_config_file.cfg");
  - bool on = util::to_value<bool>(settings.get_value(“on”,”true”));
  - …
- Observe that one simply invokes the load method with the path for the cfg-file that should be loaded. After this one can get values calling the get_value method. This method returns the value as a std::string. The first argument is the name of the variable, the second argument is the default value if no such variable exist in the cfg-file. Notice that above we use the util::to_value functionality to convert the string-value into the actual correct type that we wish to have.

Using multi-valued names

- Recall the syntax “list = item1 item2 item3”. To obtain all values one must write

    - std::vector<std::string> values = settings.get_values(“list”);

- If one only wrote settings.get_value(“list”) one would just get “item1” as the result. The above “values” vector contains all three string values “item1”, “item2” and “item3”.

-