Example of cfg-syntax

- A cfg file may look like this
  - # I am a comment
  - on = true   #I am a comment too
  - list = item1 item2 item3 item4
  - filename = script.m
  - max_iterations = 100
  - stop_threshold = 0.01
- The syntax is simple and hopefully quite trivial. One can write any setting one wants into a config file. Our config utility will extract all variable names and associate their values with these names. The data is stored in a kind of runtime lookup table and can be queried at runtime.

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.