I usually separate configuration from main class, so a "Settings" class wich contains all global settings. Example :

In "settings.h"


```
class Settings
{
    private:
......(all app wide settings here)

    public:
......(all getters/setters for public accessible settings)

};

Settings &globalSettings();
```


and, in "settings.cpp"


```
Settings &globalSettings()
{
    static Settings settings;

    return settings;
}
```


So the only public stuff is the global 'globalSettings()' function wich gives access to app-wide settings.
In settings constructor you can then put code to load (Xmlize) status, and in destructor code to save (Xmlize) status.

Ciao

Max