Error handling / error logging in C++ for library/app combo

Asked 16 years, 3 months ago Modified 1 year, 5 months ago

Viewed 2k times

4

I've encountered the following problem pattern frequently over the years:

- I'm writing complex code for a package comprised of a standalone application and also a library version of the core that people can use from inside other apps.
- Both our own app and presumably ones that users create with the core library are likely to be run both in batch mode (off-line, scripted, remote, and/or from command line), as well as interactively.
- The library/app takes complex and large runtime input and there may be a variety of error-like outputs including severe error messages, input syntax warnings, status messages, and run statistics. Note that these are all *incidental* outputs, not the primary purpose of the application which would be displayed or saved elsewhere and using different methods.
- Some of these (probably only the very severe ones) might require a dialog box if run interactively; but it needs to log without stalling for user input if run in batch mode; and if run as a library the client program

obviously wants to intercept and/or examine the errors as they occur.

- It all needs to be cross-platform: Linux, Windows, OSX. And we want the solution to not be weird on any platform. For example, output to stderr is fine for Linux, but won't work on Windows when linked to a GUI app.
- Client programs of the library may create multiple instances of the main class, and it would be nice if the client app could distinguish a separate error stream with each instance.
- Let's assume everybody agrees it's good enough for the library methods to log errors via a simple call (error code and/or severity, then printf-like arguments giving an error message). The contentious part is how this is recorded or retrieved by the client app.

I've done this many times over the years, and am never fully satisfied with the solution. Furthermore, it's the kind of subproblem that's actually not very important to users (they want to see the error log if something goes wrong, but they don't really care about our technique for implementing it), but the topic gets the programmers fired up and they invariably waste inordinate time on this detail and are never quite happy.

Anybody have any wisdom for how to integrate this functionality into a C++ API, or is there an accepted paradigm or a good open source solution (not GPL,

please, I'd like a solution I can use in commercial closed apps as well as OSS projects)?

c++ error-handling api-design error-logging

Share

Improve this question

Follow

edited Jun 29, 2023 at 8:54

Brian Tompsett - 汤莱恩

5,875 ● 72 ● 61 ● 133

asked Sep 2, 2008 at 13:36

Larry Gritz

13.6k ● 7 ● 43 ● 42

2 Answers

Sorted by: Highest score (default)

1

We use Apache's Log4cxx for logging which isn't perfect, but provides a lot of infrastructure and a consistent approach across projects. I believe it is cross-platform, though we only use it on Windows.

It provides for run time configuration via an ini file which allows you to control how the log file is output, and you could write your own appenders if you want specific behaviour (e.g. an error dialog under the UI).

If clients of your library also adopt it then it would integrate their logging output into the same log file(s).

Differentiation between instances of the main class could be supported using the nested diagnostic context (NDC)

feature.

Share Improve this answer

Follow

answered Sep 2, 2008 at 14:56

Rob Walker

47.4k ● 15 ● 100 ● 137

1

Log4Cxx should work for you. You need to implement a provider that allows the library user to catch the log output in callbacks. The library would export a function to install the callbacks. That function should, behind the scenes, reconfigure log4cxxx to get rid of all appenders and set up the "custom" appender.

Of course, the library user has an option to not install the callbacks and use log4cxx as is.

Share Improve this answer

Follow

answered Sep 4, 2008 at 18:55

user3458
