

Logs

csc207, Fall 2015



Logging

Logging is the process of recording events that occur during execution of a program in a central location.

Messages may be written to a log file or to another location such as the standard error stream, `System.err`.

Aside: stdin, stdout, and stderr

These are streams that connect a program to its environment.

- `stdin`: “the standard input stream”
(defaults to the keyboard)
- `stdout`: “the standard output stream”
(defaults to the terminal)
- `stderr`: “the standard error stream”
(defaults to the terminal)

We can override the defaults and redirect these streams to something different.

This is the case for unix commands, and for Java programs (and Python programs, etc.)

java.util.logging.Logger

java.util.Logger provides logging capabilities.

When we generate a log message, we give it a level of severity:

SEVERE (highest)

WARNING

INFO

CONFIG

FINE

FINER

FINEST (lowest)

By default, only messages with level INFO or higher are shown. But we can control that threshold using `setLevel`.

java.util.Handler

A handler receives messages from the logger and either writes them to file or the console, or passes them on to be handled elsewhere.

In Java, Handler classes include:

`ConsoleHandler` (sends log messages to stderr) and |
`FileHandler` (sends log messages to a file).

Why log?

Logging is very useful for diagnosing problems if

- you can't use a debugger, or
- using a debugger would mask the problem, for example, if the problem you are diagnosing is related to timing.

For your Android app, logging will be very helpful.