The Oracle Logging Framework (OLF) is an Oracle PL/SQL based logging and debugging utility that supports "levels" similar to Log4J. The key feature of the OLF is the ability to dynamically set a logging level based on almost any combination of module, action, username, and sid and instance\_id. The OLF is integrated with the [Instrumentation Library for Oracle](https://method-r.com/software/ilo/) (ILO) to provide task timing and extended SQL tracing, and since the OLF requires the setting of module and action to be effective, the ILO is the best suited utility to provide that functionality.

The OLF has been moved to Git ...

The OLF provides the ability to support the following instrumentation categories:

* Debugging
* Logging
* Runtime registration
* Metric collection
* Logging Levels

The Oracle Logging Framework (OLF) follows the basic Log4j logging levels and values:

* FATAL - 50000
* ERROR - 40000
* WARN - 30000
* INFO - 20000
* DEBUG - 10000

Log4j also includes an ALL and an OFF level, and the Oracle Logging Framework (OLF) includes these levels as well as a TIMED level that is used to insure that task timing is always logged. Each level is set so that ALL < DEBUG < INFO < WARN < ERROR < FATAL< TIMED < OFF. A default level of FATAL is assigned in the OLF code and can be overridden in the dynamic configuration or it can be explicitly set.

**Dynamic Control**

One of the biggest features of the OLF is the ability to dynamically set log levels. Whether it's for a single user, program or task you don't want to have to stop everything and reset a configuration file and then restart the application. You really want to be able to set the logging dynamically on the fly. With the OLF you can do this through the dblog\_config table.

**Simple Example**

A very simple example of a logged PL/SQL block is the following:

declare

  l\_num number;

begin

  ilo\_task.begin\_task(

    module => 'module',

    action => 'action');

  --

  dblog.info('Before statement');

  --

  select 1 into l\_num from dual;

  --

  dblog.info('After statement');

  --

  ilo\_task.end\_task;

exception

  when others then

    dblog.error('An error occurred');

    --

    ilo\_task.end\_all\_tasks;

    --

    RAISE;

end;