

# Chapter 1

## Logger API

### 1.1 Opening the logger

```
bool Open(Facility facility = Facility::user,  
          const std::vector<MessageSink *> & sinks = {},  
          const char *ident = nullptr)
```

If `sinks` is empty, the logger will send all log messages to the loopback address.

If `ident` is `nullptr`, the logger will use the ident given by the runtime. On macOS and FreeBSD, this comes from `getprogname(3)`. On Linux, this comes from `program_invocation_short_name(3)`.

### 1.2 Alternate or additional sinks

The logger sends messages to *sinks*, which must implement the `Dwm::Mclog::MessageSink` interface.

#### 1.2.1 Dwm::Mclog::MessageSink interface

```
virtual bool Process(const Message & msg)
```

#### 1.2.2 Adding or removing sinks

Sinks may be added to the logger by using the `AddSinks()` member of the logger.

```
bool AddSinks(const std::vector<MessageSink *> & sinks)
```

Sinks may be removed from the logger by using the `RemoveSinks()` member of the logger.

```
bool RemoveSinks(const std::vector<MessageSink *> & sinks)
```

#### 1.2.3 Dwm::Mclog::LogFile

```
LogFile(const std::string & path, mode_t permissions = 0644,  
        RollPeriod period = RollPeriod::days_1, uint32_t keep = 7)
```

#### 1.2.4 Dwm::Mclog::OstreamSink

```
OstreamSink(std::ostream & os)
```

### **1.2.5 Dwm::Mclog::SyslogSink**

```
SyslogSink(const char *ident, int facility)
```

## **1.3 Sending log messages**

## **1.4 Closing the logger**