Log Library

Minimal Overhead

- About 5-10 µsec. per call.

Fail-safe

- Does not stall.
- Can lose data.
- Does nothing gracefully.

Interrupt-safe

- Only MakeLogRecord allocates memory.

Provides a Name Service

Provides a Data Service

Does not use MacOS Toolbox (except to format the timestamp).

Name Service

Create a new LogRecord

```
logRecordPtr =
  MakeLogRecord("MyLogRecord", 64);
```

MakeLogRecord allocates memory in the resident pool.

Reference existing LogRecords

```
logRecordPtr =
  GetLogRecordPtr("MyLogRrecord");
```

Returns NULL if no such record.

Iterate over all LogRecords

```
LogRecordIterateCreate(&cookie);
logRecordPtr = LogRecordIterate(&cookie);
LogRecordIterateDispose(&cookie);
```

These functions use the Name Registry property iterator functions.

Data Service

Store a LogEntryRecord

```
status = WriteLogEntry( /* From C only     */
    logRecordPtr,  /* This LogRecord       */
    recordTag,     /* OSType identifier    */
    recordFormat,  /* Bit-encoded longword */
    zero to ten data words
);
status = StoreLogEntry( /* Pascal callable */
    logRecordPtr,  /* This LogRecord       */
    &thisLogEntry  /* Get data from here   */
  );
```

Read a LogEntryRecord

```
status = ReadLogEntry( /* noErr on success   */
    logRecordPtr,    /* This LogRecord    */
    &thisLogEntry    /* Store data here   */
  );
```

Controlling Logging

Enable or Disable Logging

```
wasEnabled = EnableLogRecord(
    logRecordPtr,    /* This LogRecord    */
    enableLogging    /* Enable if TRUE    */
);
```

Logging is enabled by default. FALSE stops logging.

Controlling Data Overrun

```
wasPreserveFirst = PreserveLogRecord(
    logRecordPtr,    /* This LogRecord    */
    preserveFirst    /* Earliest if TRUE  */
);
```

The latest N entries are preserved by default. TRUE preserves the earliest entries.

LogEntryRecord Format

```
struct LogEntryRecord {
  AbsoluteTime eventTime; /* When stored   */
  UInt32       sequence;  /* Entry number    */
  OSType       idCode;    /* Caller's ID   */
  UInt32       format;    /* Data format   */
  UInt32       data[kLogEntryDataSize];
};
```

Data format types:

Signed and unsigned decimal values

Hex data with OSType

Hex data without OSType

Character String

All data must be cast to 32-bit words.

Formatting Log Entries

```
LogConvertTimestamp(  /* Uses MacOS Toolbox */
    &thisLogEntry,
    &eventDateTime,
    &residualNanoseconds
  );
FormatLogEntryTimestamp(
    resultStringPtr,
    &eventDateTime,
    residualNanoseconds
  );
FormatLogEntryData(
    &thisLogEntry,
    resultStringPtr
  );
```

68000 Support

- The library may be called from 68000 code. It contains private glue routines for the Name Registry functions.
- There is a MacsBug dcmd that displays all available LogRecord data.

Conversion to Maxwell

- The library must be able to copy data between task area and a system-resident area.
- The dcmd will need adaptation.