

**NAME**

lh\_stats, lh\_node\_stats, lh\_node\_usage\_stats, lh\_stats\_bio, lh\_node\_stats\_bio,  
lh\_node\_usage\_stats\_bio - LHASH statistics

**SYNOPSIS**

```
#include <openssl/lhash.h>

void lh_stats(LHASH *table, FILE *out);
void lh_node_stats(LHASH *table, FILE *out);
void lh_node_usage_stats(LHASH *table, FILE *out);

void lh_stats_bio(LHASH *table, BIO *out);
void lh_node_stats_bio(LHASH *table, BIO *out);
void lh_node_usage_stats_bio(LHASH *table, BIO *out);
```

**DESCRIPTION**

The **LHASH** structure records statistics about most aspects of accessing the hash table. This is mostly a legacy of Eric Young writing this library for the reasons of implementing what looked like a nice algorithm rather than for a particular software product.

*lh\_stats()* prints out statistics on the size of the hash table, how many entries are in it, and the number and result of calls to the routines in this library.

*lh\_node\_stats()* prints the number of entries for each 'bucket' in the hash table.

*lh\_node\_usage\_stats()* prints out a short summary of the state of the hash table. It prints the 'load' and the 'actual load'. The load is the average number of data items per 'bucket' in the hash table. The 'actual load' is the average number of items per 'bucket', but only for buckets which contain entries. So the 'actual load' is the average number of searches that will need to find an item in the hash table, while the 'load' is the average number that will be done to record a miss.

*lh\_stats\_bio()*, *lh\_node\_stats\_bio()* and *lh\_node\_usage\_stats\_bio()* are the same as the above, except that the output goes to a **BIO**.

**RETURN VALUES**

These functions do not return values.

**SEE ALSO**

[bio\(3\)](#), [lhash\(3\)](#)

**HISTORY**

These functions are available in all versions of SSLeay and OpenSSL.

This manpage is derived from the SSLeay documentation.