

hashdb Quick Reference

<https://github.com/NPS-DEEP/hashdb/wiki>

New Database

```
create [-b <block size>] [-a <byte alignment>]
  [-m <max count:max sub-count>] [-t <hash
  prefix bits:hash suffix bytes>] <hashdb.hdb>
```

Create a new hash database.

Import/Export

```
ingest [-r <repository name>] [-w
  <whitelist.hdb>] [-s <step size>] [-x rel]
  <hashdb.hdb> <import directory>
```

Import from path recursively into hash database, labeling hashes in the whitelist and hashes matching entropy traits. Can disable recursion, entropy, labels

```
import_tab [-r <repository name>] [-w
  <whitelist.hdb>] <hashdb.hdb> <tab.txt>
```

Import from tab file into hash database, labeling hashes in the whitelist.

```
import <hashdb.hdb> <hashdb.json>
```

Import JSON format data into hash database.

```
export [-p <begin:end>] <hashdb.hdb>
  <hashdb.json>
```

Export all or part of hash database in JSON format.

Database Manipulation

```
add <A.hdb> <B.hdb>
add_multiple <A.hdb> <B.hdb> ... <C.hdb>
add_repository <A.hdb> <B.hdb> <repository name>
add_range<A.hdb> <B.hdb> <m:n>
```

$A \rightarrow B$ add A into B

$A + B + \dots \rightarrow C$ add A, B, \dots into C .

$A_r \rightarrow B$ add when repository name matches.

$A_{m:n} \rightarrow B$ add hashes that have source counts within range, inclusive.

```
intersect <A.hdb> <B.hdb> <C.hdb>
intersect_hash <A.hdb> <B.hdb> <C.hdb>
subtract <A.hdb> <B.hdb> <C.hdb>
```

$A \cap B \rightarrow C$ add when hash and source are common.

$A \cap B \rightarrow C$ add when hashes are common.

$A - B \rightarrow C$ add when hash and source not common.

```
subtract_hash <A.hdb> <B.hdb> <C.hdb>
subtract_repository <A.hdb> <B.hdb> <repository
  name>
```

$A - B \rightarrow C$ add when hashes are not common.

$A_{\bar{r}} \rightarrow B$ add unless repository name matches.

Scan

```
scan_list [-j e|o|c|a] <hashdb.hdb> <hashes file>

scan_hash [-j e|o|c|a] <hashdb.hdb> <hex block
  hash>
scan_media [-s <step size>] [-j e|o|c|a] [-x r]
  <hashdb.hdb> <media image file>
```

Scan hashes file for hash match, return expanded, expanded optimized, count only, or approximate count.

Scan for hash match, return expanded, expanded optimized, count only, or approximate count.

Scan media image for hash match, return expanded, expanded optimized, count only, or approximate count. Can disable recursion.

Statistics

```
size <hashdb.hdb>
sources <hashdb.hdb>
histogram <hashdb.hdb>
duplicates [-j e|o|c|a] <hashdb.hdb> <number>
hash_table [-j e|o|c|a] <hashdb.hdb> <hex file
  hash>
read_media <media image file> <offset> <count>
read_media_size <media image file>
```

Print size information for internal database tables.

Print source information.

Print hash distribution.

Print hashes sourced the given number of times.

Print hashes associated with the source file hash.

Print raw bytes from the media image file.

Print the size of the media image file.

Performance Analysis

```
add_random <hashdb.hdb> <count>
scan_random [-j e|o|c|a] <hashdb.hdb> <count>
add_same <hashdb.hdb> <count>
scan_same [-j e|o|c|a] <hashdb.hdb> <count>
```

Add random hashes, log to `timestamp.json`.

Scan random hashes, log to `timestamp.json`.

Add same hashes, log to `timestamp.json`.

Scan same hashes, log to `timestamp.json`.

bulk_extractor Scanner

```
bulk_extractor -E hashdb -S hashdb_mode=import -o outdir1 -R my_import_dir
bulk_extractor -E hashdb -S hashdb_mode=import -o outdir1 my_media_image
bulk_extractor -E hashdb -S hashdb_mode=scan -S hashdb.scan_path=
  outdir1/hashdb.hdb -o outdir2 my_media_image2
```

Import directory.

Import media image.

Scan media image.