

hashdb Quick Reference

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

New Database

```
create [-b <block size>] <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>
```

$A \rightarrow B$ add A into B

```
add_multiple <A.hdb> <B.hdb> ... <C.hdb>
```

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

```
add_repository <A.hdb> <B.hdb> <repository name>
```

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

```
add_range <A.hdb> <B.hdb> <m:n>
```

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

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

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

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

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

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

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

```
subtract_hash <A.hdb> <B.hdb> <C.hdb>
```

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

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

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

Scan

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

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

```
scan_hash [-j e|o|c|a] <hashdb.hdb> <hex block  
  hash>
```

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

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

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

Statistics

```
size <hashdb.hdb>
```

Print size information for internal database tables.

```
sources <hashdb.hdb>
```

Print source information.

```
histogram <hashdb.hdb>
```

Print hash distribution.

```
duplicates [-j e|o|c|a] <hashdb.hdb> <number>
```

Print hashes sourced the given number of times.

```
hash_table [-j e|o|c|a] <hashdb.hdb> <hex file  
  hash>
```

Print hashes associated with the source file hash.

```
read_media <media image file> <offset> <count>
```

Print raw bytes from the media image file.

```
read_media_size <media image file>
```

Print the size of the media image file.

Performance Analysis

```
add_random <hashdb.hdb> <count>
```

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

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

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

```
add_same <hashdb.hdb> <count>
```

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

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

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

bulk_extractor Scanner

```
bulk_extractor -E hashdb -S hashdb_mode=import -o outdir1 -R my_import_dir
```

Import directory.

```
bulk_extractor -E hashdb -S hashdb_mode=import -o outdir1 my_media_image
```

Import media image.

```
bulk_extractor -E hashdb -S hashdb_mode=scan -S hashdb_scan_path=  
  outdir1/hashdb.hdb -o outdir2 my_media_image2
```

Scan media image.