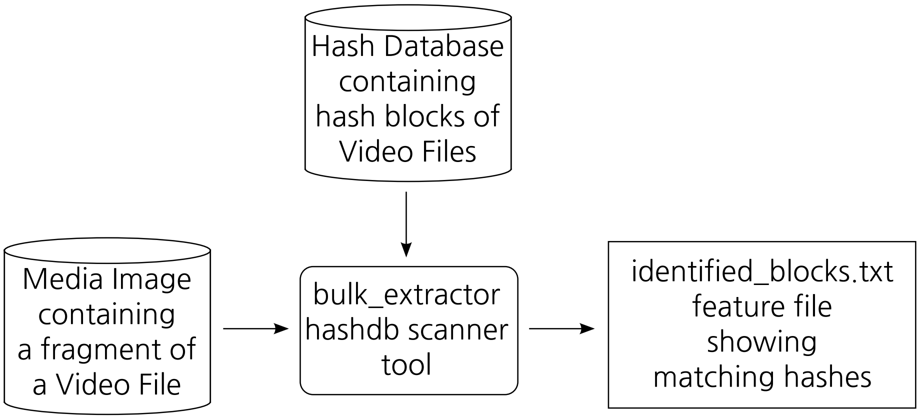
Demo: Finding Fragments of Previously Encountered Data
using *hashdb* and *bulk_extractor*

In this demo, we find that a media image contains part of a previously encountered video file. This demo uses the following resources:

- A media image containing a fragment of a video file.
- A *hashdb* block hash database containing block hashes from the previously encountered video file.
- The *hashdb* tool.
- *bulk_extractor* compiled with the *hashdb* scanner.

Here is the workflow:


Hash Database containing hash blocks of Video Files
Media Image containing a fragment of a Video File
bulk_extractor hashdb scanner tool
identified_blocks.txt feature file showing matching hashes

Scan the media image for parts of a video file.

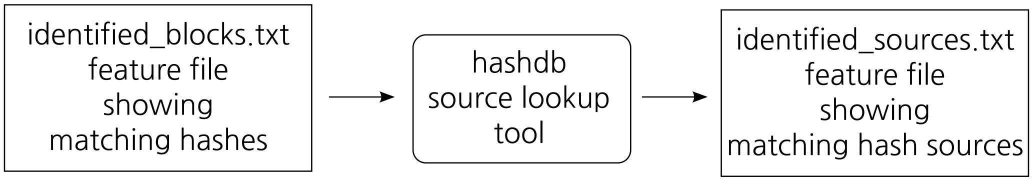
Setup:

1. Download and install *hashdb* from `http://digitalcorpora.org/downloads/hashdb` as described at `https://github.com/simsong/hashdb/wiki/Installing-hashdb`.
2. Download and install *bulk_extractor* compiled with *hashdb* from `http://digitalcorpora.org/downloads/hashdb` as described at `https://github.com/simsong/hashdb/wiki/Installing-hashdb`.
3. This demo requires hash dtabase `mock_video.hdb` created by demo "Demo: Creatng a Block Hash Database using *hashdb* and *md5deep*" available at `http://digitalcorpora.org/downloads/hashdb/demo/create_hdb_demo.pdf`. Please follow that demo to create your `mock_video.hdb` hash database and copy it into your current working directory.
4. Download the media file to scan from here: `http://digitalcorpora.org/downloads/hashdb/demo/mock_video_redacted_image`. This media file contains a fragment of the demo video file, specifically, a contiguous 64KiB section near the end of about 10 MiB of video data:

Steps:

1. Now scan for matching hash values: Using a command window, go to your working directory and then run *bulk_extractor*, specifying the paths to the hash database and the media:
   ```
   $ bulk_extractor -e hashdb -o outdir -S hashdb_mode=scan \
     -S hashdb_scan_path_or_socket=mock_video.hdb \
     mock_video_redacted_image
   ```
2. View the feature file using an editor or use the *Bulk Extractor Viewer* tool. For example to view with Windows Notepad, type:
   ```
   $ notepad outdir/identified_blocks.txt
   ```
   An example hash block match looks like this:
   ```
   12452352    3b6b477d391f73f67c1c01e2141dbb17    1
   ```

Seeing hash `3b6b477...` at Forensic path `12452352` shows that a hash block match was found, but what file does it match? We find the file that contains the hash by using a *hashdb* source lookup:



Look up the file that has the hash.

Steps to look up source information about the identified blocks:

1. Using a command window, go to your working directory and then run the *hashdb* tool:
   ```
   $ hashdb expand_identified_blocks mock_video.hdb \
     outdir/identified_blocks.txt > outdir/identified_sources.txt
   ```
2. Now view file `outdir/identified_sources.txt` to see features containing source information. This example line:
   ```
   12452352    3b6b477d391f73f67c1c01e2141dbb17 \
   repository_name=repository_mock_video.xml, \
   filename=/home/bdallen/demo/mock_video.mp4, \
   file_offset=10485760
   ```
   states that the block at Forensic path `12452352` matches the block `10485760` bytes into the `mock_video.mp4` video file in the hash database, indicating a positive match with fragments of data in the previously encountered video file.

This completes the demo.