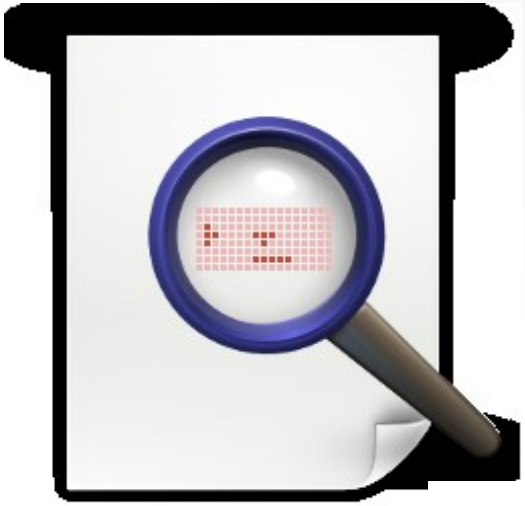
Tools for identifying duplicate files and known software files

Creighton Barrett
Digital Archivist, Dalhousie University Archives

BitCurator User Forum, Northwestern University
April 27-28, 2017

DALHOUSIE UNIVERSITY 1818 2018

Tools



AccessData®

FSlint (finds file system “lint”)

- Duplicates
- Installed packages
- Bad names
- Name clashes
- Temp files
- Bad symlinks
- Bad IDs
- Empty directories
- Non stripped binaries
- Redundant whitespace

FSlint – Duplicates

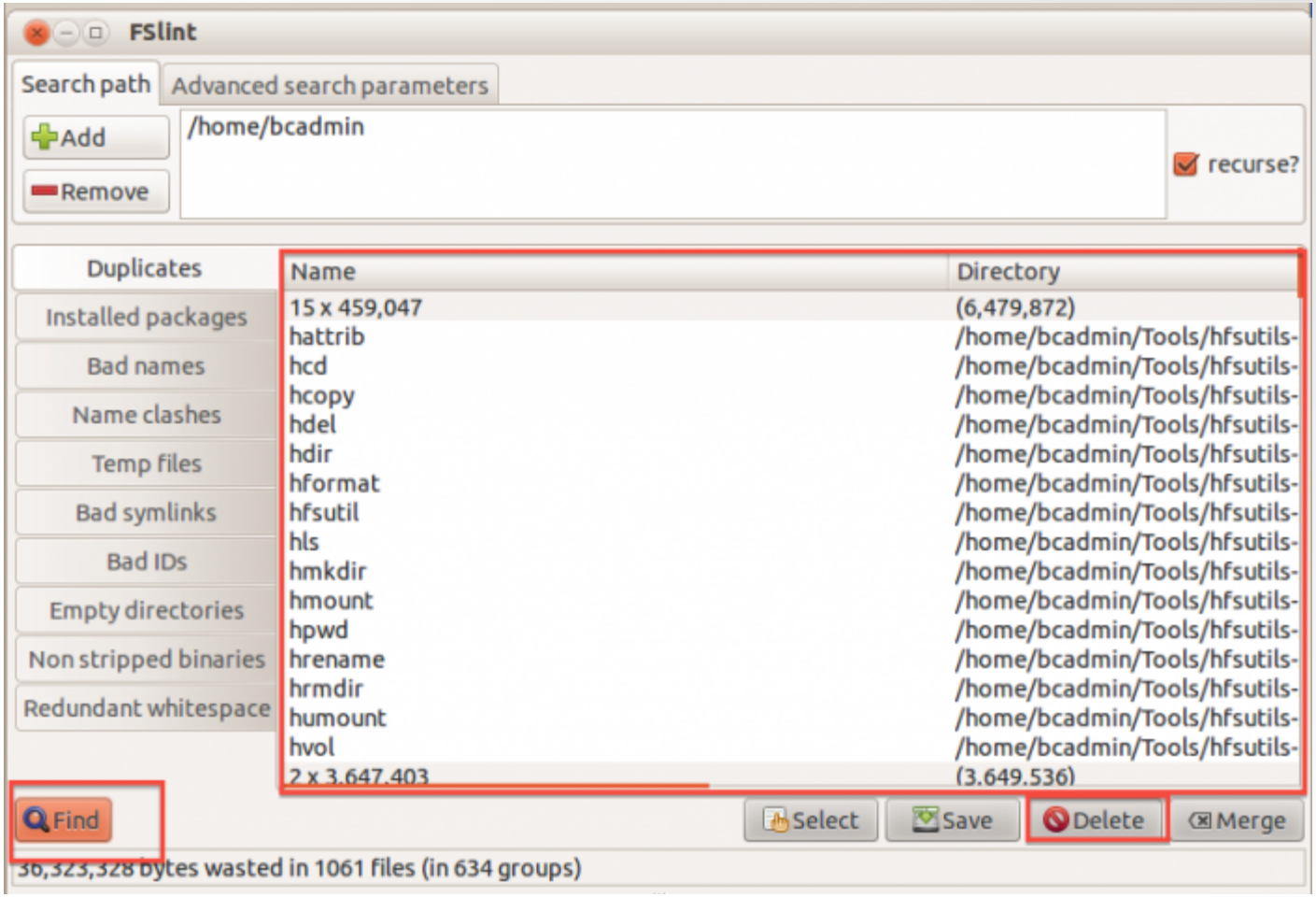
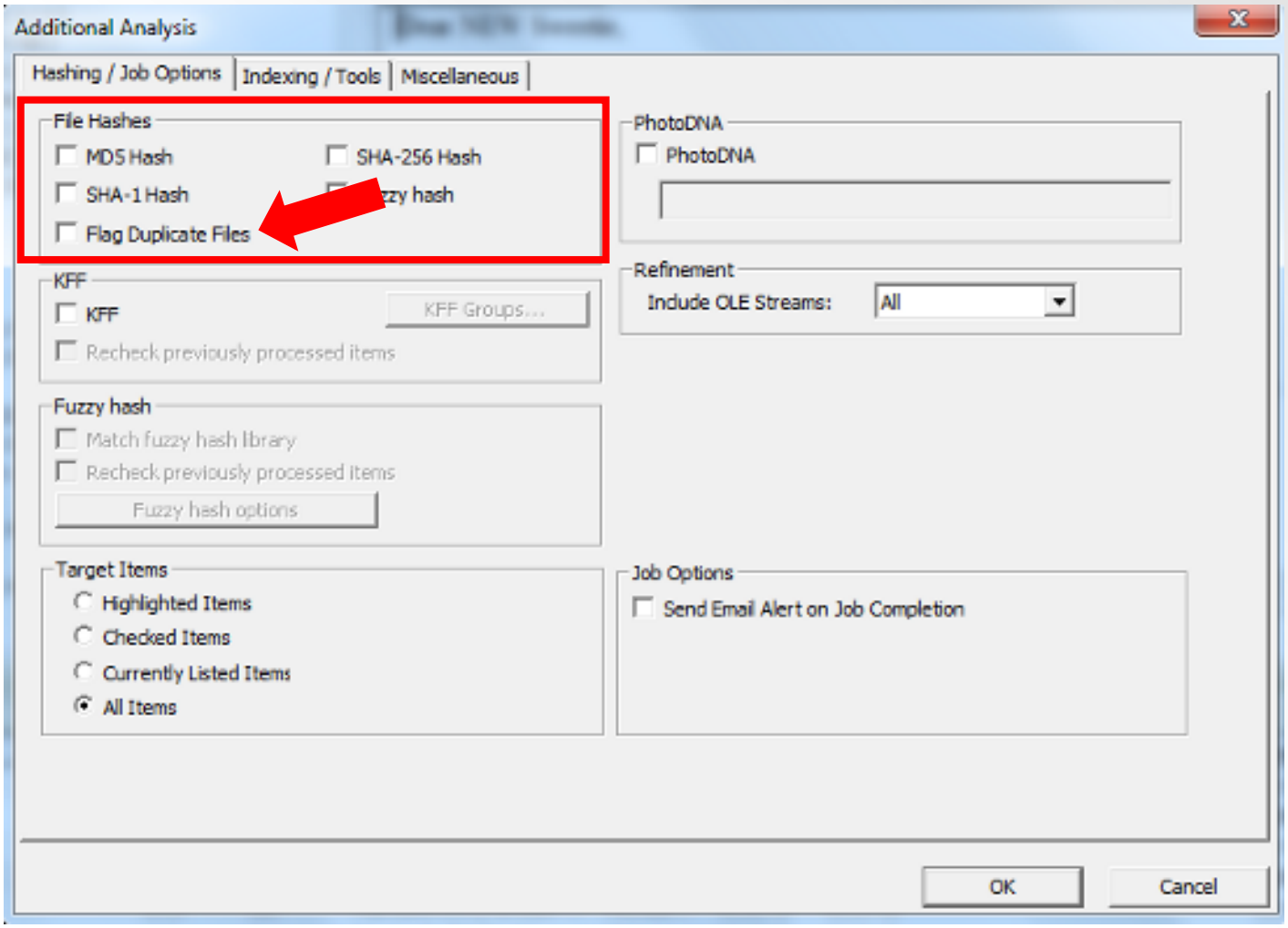


Image source (BitCurator wiki):
https://wiki.bitcurator.net/index.php?title=Identify_and_delete_duplicate_files

FTK – Flag Duplicates

- Simpler process than FSlint, still a powerful feature
  - Checks entire file and generates MD5
  - Assigns primary status to first instance of each MD5
  - Assigns secondary status to subsequent instances of each MD5

FTK – Flag Duplicates



NSRL Reference Data Set (RDS)

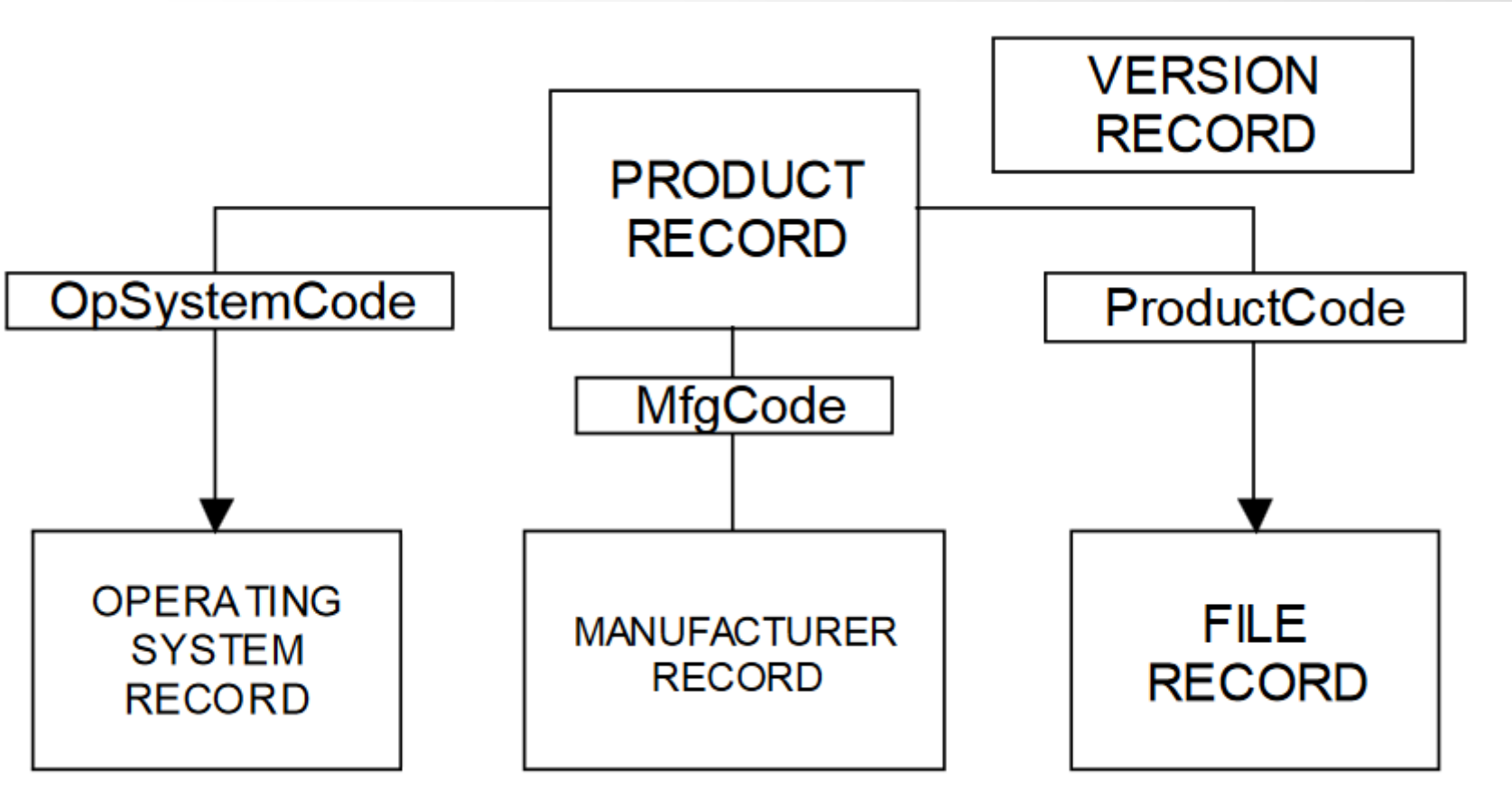


Image source (NSRL): https://www.nsrl.nist.gov/Documents/Data-Formats-of-the-NSRL-Reference-Data-Set-16.pdf

NSRL Reference Data Set (RDS)

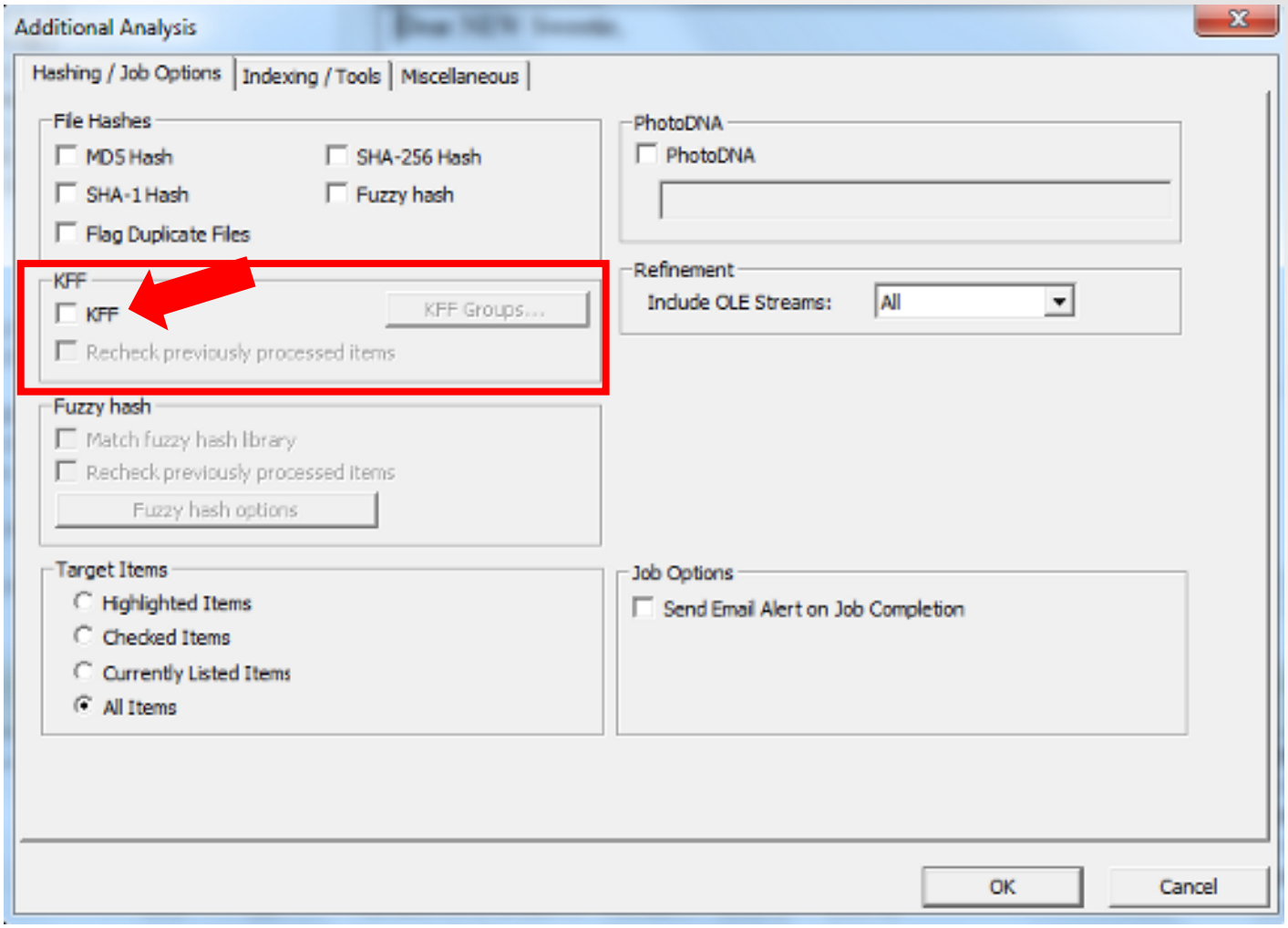
- Hashsets and metadata used in file identification
- Data can be used in third-party digital forensics tools

- RDS is updated four times each year
- As of v2.55, RDS is partitioned into four divisions:
  - Modern – applications created in or after 2000
  - Legacy – applications created in or before 1999
  - Android – Mobile apps for the Android OS
  - iOS – Mobile apps for iOS

FTK – Known File Filter (KFF)

- KFF data – hash values of known files that are compared against files in an FTK case
- KFF data can come from pre-configured libraries (e.g., NSRL RDS, DHS, ICE, etc.) or custom libraries
- FTK ships with version of NSRL RDS bifurcated into “Ignore” and “Alert” libraries
- KFF Server – used to process KFF data against evidence in an FTK case
- KFF Import Utility – used to import and index KFF data

FTK – Known File Filter (KFF)



Other tools to work with NSRL RDS

- nsrlsvr - https://github.com/rjhansen/nsrlsvr/
  - Keeps track of 40+ million hash values in an in-memory dataset to facilitate fast user queries
  - Supports custom libraries ("local corpus")

- nsrllookup - https://rjhansen.github.io/nsrllookup/
  - Command-line application
  - Works with tools like hashdeep: http://md5deep.sourceforge.net/

- National Software Reference Library - MD5/SHA1/File Name search - http://nsrl.hashsets.com

Bill Freedman fonds filtered in FTK

Filter	Description	# of files	Size
Unfiltered	All files in case	26,651,084	3,568 GB
Primary status	Duplicate File indicator IS “Primary”	731,417	83.48 GB
Secondary status	Duplicate File indicator IS “Secondary”	16,569,218	271.5 GB
KFF Ignore	Match all files where KFF status IS “Ignore”	2,548,119	44.29 GB
No KFF Ignore	Match all files where KFF status IS NOT “Ignore” + KFF status IS “Not checked”	24,102,965	3524 GB
Primary status + No KFF Ignore	Match all files where duplicate file indicator IS “Primary” + KFF status IS NOT “Ignore”	626,351	71.95 GB
Actual files + Primary status + No KFF Ignore	Match all disk-bound files where duplicate file indicator IS “Primary” + KFF status IS NOT “Ignore”	103,412	61.81 GB

Questions

- Does it matter which duplicate file is selected for preservation? What if there are MD5 matches with different file names or extensions?
- Can queries against NSRL RDS be incorporated into BitCurator workflows?
- Could provenance-based libraries of “known file” hashes be incorporated into BitCurator workflows?
- Can repositories share provenance-based hash libraries (expose our “local corpus” of MD5s…)?