https://doi.org/10.48009/2_iis_2014_421-430

Issues in Information Systems
Volume 15, Issue II, pp. 421-430, 2014

DETERMINING USER ACTIONS IN OS X BASED ON QUICKLOOK THUMBNAIL CACHE DATABASE ENTRIES

Sara Newcomer, Lockheed Martin, sara.r.newcomer@lmco.com

ABSTRACT

The purpose of this study was to document the structure of the ***index.sqlite*** *file associated with the QuickLook Thumbnail Cache in OS X, and test and confirm how and when entries are created in this database. Using OS X version 10.8.4 (Mountain Lion.), the structure of the database was analyzed and the information stored in fields was interpreted based on entries corresponding to known files. A new account was created, actions were documented, and the results of those actions on the database were reviewed. Many features in the OS X Graphical User Interface (GUI) generate thumbnails or previews of files. This article will provide a brief overview of the OS X GUI, as well as the results of the study designed to confirm which GUI features result in database entries. The end goal is to make accurate statements about a user's actions in the OS X GUI based on information contained in the database.*

Keywords: Index.sqlite, QuickLook, Thumbnails, OS X, Digital Forensics

INTRODUCTION

The QuickLook Thumbnail Cache database *index.sqlite* is located in a hidden operating system directory, not typically navigated to by users, inaccessible to non-admin accounts. Since this database tracks user's Finder navigation activities, it can be used to determine the folders accessed by a user on media attached to the system. This included locally attached storage and network-attached storage. A separate database is maintained for each user account on the system.

Many features of the OS X Graphical User Interface (GUI) generate thumbnails or previews of files including:

- Viewing a directory in Finder
- Viewing a file with QuickLook
- Displaying information about a file
- Files located in a stack on the Dock

Not all of these actions result in changes to the database.

BACKGROUND

For those not familiar with the OS X GUI, the information below will explain Finder, QuickLook, Stacks on the Dock, and how thumbnails or previews of files are generated automatically for these applications.

OS X runs on the Finder application. For the GUI to function, Finder must be running. The Finder application is somewhat analogous to the Windows Explorer in Microsoft (MS) Windows OS; Finder provides the interface for users to browse file systems connected to the computer, both remote and local. Finder allows users to view files using four different viewing options; Icon, List, Column, and Cover Flow.

In all Finder views a thumbnail can be created for files. The default view for Finder is Icon, and newly mounted volumes automatically open in Icon view the first time they are opened with Finder. Icon view generates a custom thumbnail for many file types including text, video, image, MS Office, iWork, and pdf files. Cover Flow generates larger previews of the files in the directory, allowing a user to scroll through the files. As a file is highlighted in Column view, a preview of the highlighted file is created. List, Column, and Cover Flow views all generate small thumbnails of files in a directory; however, small custom thumbnails are not generated for as many files types as the large thumbnails created in Icon view or the previews created in Cover Flow. For example in the List, Column, and Cover Flow views in Figure 1, the files are all images. A small custom thumbnail is created for each file. If the files were a file type that does not generate small custom thumbnails like a pdf or docx file, a small generic thumbnail would be used. Similar to the preview created in Column view, when Finder is used to display information about a file a preview of the file is included.

QuickLook is an application integrated into Finder with the release of 10.5 (Leopard). In addition to creating the custom thumbnails and previews in Finder discussed above, QuickLook will open a separate window to preview a file when a user presses the spacebar. QuickLook works for many complex file types including: PDF, MS Office, iWork, html, txt, and multimedia files (e.g. JPEG, MP3, PNG, WAV, MPEG4, AVI). QuickLook provides a quick way to view the contents of a file without having to open its associated application. With a file highlighted in Finder, pressing the space bar opens the QuickLook window. Using the mouse or arrow buttons on the keyboard, a user can browse through the contents of the directory in the open QuickLook window; QuickLook remains open until the user presses the space bar again or clicks the close button in the upper left corner of the QuickLook window. The QuickLook interface also allows a user to view files using the full screen, open the file in the default application, scroll through files with multiple pages, and in the newer versions of the OS quickly share a file on social media sites.

The Dock is a menu bar in OS X that is part of the Desktop. It contains shortcuts to applications and folders for easy access. Stacks is a feature that provides access to frequently used directories. In different versions of OS X, stacks for different directories have been seen by default on the Dock including *Documents*, *Applications*, and *Downloads*. In the latest release, only the Users *Downloads* directory has a stack on the Dock by default. Users can customize the Dock to add or remove stacks. The default display for a stack is for the icon of the file most recently placed in the directory to be shown at the front on the Dock.

RESEARCH METHODOLOGY

The research focuses on the questions: What information is stored in the QuickLook thumbnail cache database, and when are entries created in the database?

The first step was determining what information is stored in the *index.sqlite* and what tools can be used to view the information. During this phase, actions that generate thumbnails or previews of files were performed on a set of control data to help identify the starting point for the second phase.

The *index.sqlite* file was copied to the Desktop using the *sudo cp* command in Terminal. The following tools were used to view the *index.sqlite* file: BlackLight Version 2013 R1 (OS X), BlackLight Version 2013 R2.1 (OS X), SQLite2009Pro v3.7.3 (Windows), and SQLite Database Browser v 2.0 b1 (Windows). The only tool capable of displaying all of the information in the database and provide access to all data contained in the fields was BlackLight Version 2013 R2.1. More recent versions of BlackLight also display all fields.

The second step was creating a new user account in OS X, documenting actions as they were performed in the GUI, and analyzing the entries created in the QuickLook thumbnail cache database. This analysis determined which actions resulted in database changes.

The new account created was setup as a Standard user, not an Administrator account. The Administrator account on the system was used to copy *index.sqlite* for analysis.

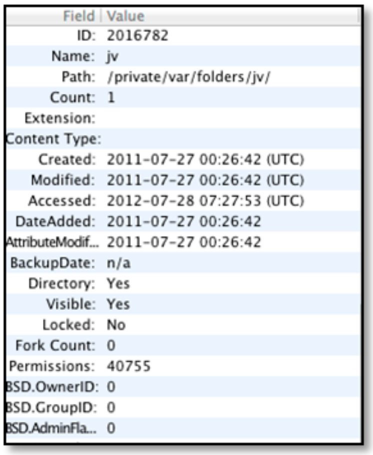
The third step was to review QuickLook thumbnail cache databases on other OS X systems to create observations about user's action on those systems. Systems with multiple users, deleted user accounts, and older version of OS X were analyzed. Reviewing these systems provided additional information about the data stored in *index.sqlite*.

RESULTS

Location of QuickLook Thumbnail Cache Database

OS X uses the */private/var/folders* directory to cache information related to some applications, including QuickLook. The */private* directory and subdirectories are hidden by default in the OS X GUI, making them an area that users may not realize is storing information traceable to specific user accounts. The */private/var* directory is owned by the system user account, root. As shown in Figure 6, inside the */private/var* directory, there are directories named with two random characters (lowercase alphanumerical characters and sometimes an underscore). There is a separate directory for each user on the system. There will always be a directory named *zz* for the root account.

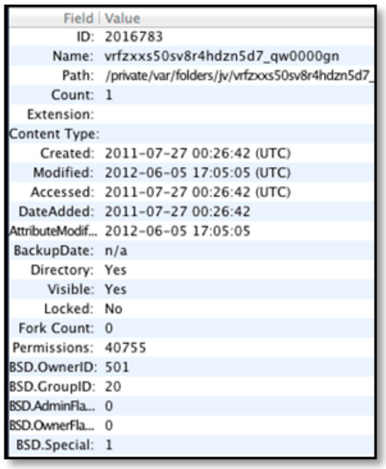
Determining which directory belongs to which user cannot be done at this level; like the */private/var/* directory, all of the two character directories belong to the system user account root. The root account always has the user identifier (UID) 0, identified in BlackLight in the field BSD.OwnerID (Figure 1).



Field	Value
ID:	2016782
Name:	jv
Path:	/private/var/folders/jv/
Count:	1
Extension:	
Content Type:	
Created:	2011-07-27 00:26:42 (UTC)
Modified:	2011-07-27 00:26:42 (UTC)
Accessed:	2012-07-28 07:27:53 (UTC)
DateAdded:	2011-07-27 00:26:42
AttributeModif...	2011-07-27 00:26:42
BackupDate:	n/a
Directory:	Yes
Visible:	Yes
Locked:	No
Fork Count:	0
Permissions:	40755
BSD.OwnerID:	0
BSD.GroupID:	0
BSD.AdminFla...	0

Figure 1: Owner of */private/var/folders/jv*

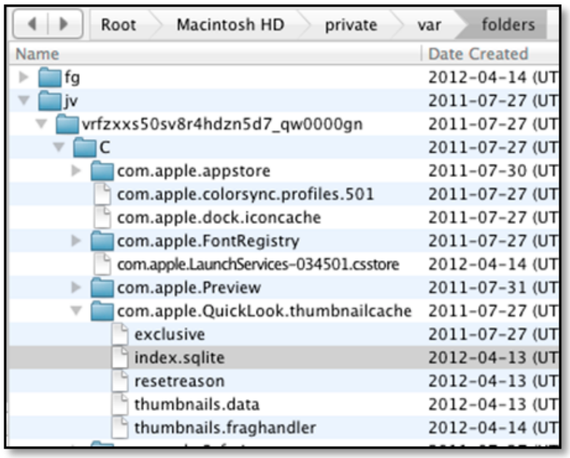
Inside the two character directories are directories with random 30 character names. The 30 characters contain lowercase alphanumerical characters and sometimes an underscore (_). It is at this level, the directory will be owned by a user account other than root. When an account is deleted in OS X, the directory structure in */private/var/folders* related to that user account remains intact. As with active accounts, the UID can be used to link the directory to a deleted account.



Field	Value
ID:	2016783
Name:	vrfzxxs50sv8r4hdzn5d7_qw0000gn
Path:	/private/var/folders/jv/vrfzxxs50sv8r4hdzn5d7_
Count:	1
Extension:	
Content Type:	
Created:	2011-07-27 00:26:42 (UTC)
Modified:	2012-06-05 17:05:05 (UTC)
Accessed:	2011-07-27 00:26:42 (UTC)
DateAdded:	2011-07-27 00:26:42
AttributeModif...	2012-06-05 17:05:05
BackupDate:	n/a
Directory:	Yes
Visible:	Yes
Locked:	No
Fork Count:	0
Permissions:	40755
BSD.OwnerID:	501
BSD.GroupID:	20
BSD.AdminFla...	0
BSD.OwnerFla...	0
BSD.Special:	1

Figure 2: Owner of */private/var/folders/jv/vrfzxxs50sv8r4hdzn5d7_qw0000gn*

Navigating further down into the directory structure varies depending on the version of OS X. In 10.5 (Leopard) and 10.6 (Snow Leopard) you may see a directory named *–Caches–*. In 10.7 (Lion) and 10.8 (Mountain Lion) the directory is named *C*. Regardless of the version, a subdirectory named *com.apple.QuickLook.thumbnailcache* will contain several files, including the SQLite database *index.sqlite*.



Root > Macintosh HD > private > var > folders

Name	Date Created
fg	2012-04-14 (UT
jv	2011-07-27 (UT
vrfzxxs50sv8r4hdzn5d7_qw0000gn	2011-07-27 (UT
C	2011-07-27 (UT
com.apple.appstore	2011-07-30 (UT
com.apple.colorsync.profiles.501	2011-07-27 (UT
com.apple.dock.iconcache	2011-07-27 (UT
com.apple.FontRegistry	2011-07-27 (UT
com.apple.LaunchServices-034501.csstore	2012-04-14 (UT
com.apple.Preview	2011-07-31 (UT
com.apple.QuickLook.thumbnailcache	2011-07-27 (UT
exclusive	2011-07-27 (UT
index.sqlite	2012-04-13 (UT
resetreason	2012-04-13 (UT
thumbnails.data	2012-04-13 (UT
thumbnails.fraghandler	2012-04-14 (UT

Figure 3: Location of the *index.sqlite* File

Information Stored in QuickLook Thumbnail Cache Database

Entries are created in the *index.sqlite* file as custom thumbnails for files are created. This occurs when a directory is viewed with Finder and for items in a stack on the Dock. The view selected in Finder will determine if a custom thumbnail is created, as well as the number and size of the thumbnails created.

Looking at an example, a CD with the volume label "Shoes" is placed in an OS X system and the volume is opened in Finder.

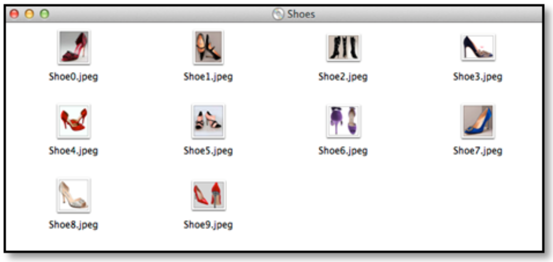


Figure 4: */Volumes/Shoes* Default Finder View

By default, this volume is opened in Icon view, and icons are generated for the ten JPEG files stored on the CD. Since this is the first time this volume was mounted, Finder defaults to Icon view. The size of the thumbnails generated from Icon view will vary depending on the icon size Finder is set to show.

Without making any changes to default settings, the expected sizes of any custom thumbnails generated should be as follows:

Table 1: Size and Number of Entries Created by Viewing a Directory in Finder

Finder View	Number of Entries	Default Thumbnail Size(s)
Icon	1	Variable between 48 and 64
List	1	16
Column	1	16
Cover Flow	2	16 and something larger

Custom thumbnails with a size of 16 for file types other than multimedia files will not be created, and there will not be an entry in the database. Generally speaking, if the user can see a custom icon for a file, an entry is made or updated in the QuickLook thumbnail cache database. If for some reason OS X does not create a custom icon for a file, there will be no entry in the QuickLook cache.

Contents of the QuickLook Thumbnail Cache Database

Tables included in the database include:

- *preferences*
- *files*
- *thumbnails*
- *pending_secure_delete_buffer*
- *reserved_buffer*

Of these five tables, information stored in the ***files*** and ***thumbnails*** table provide information of interest.

The *files* Table

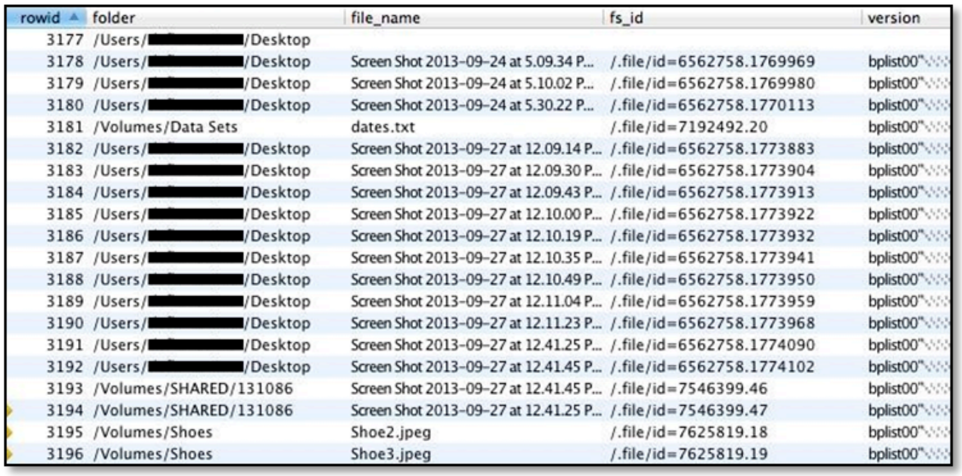
The ***files*** table is the starting point for finding information of interest. Table 1, shown below, provides a list of the fields with a description of the data contained:

Table 2: Fields Contained in *files* Table

Field Name	Description
rowid	Order in which the entries were placed in the database. The entries for all files present in a directory when it is initially viewed will have consecutive entries. This field is the primary key for entries and will be used within the SQLite database to link information from other tables.
Folder	Full path to the file. Files that are located on media other than the OS volume will have a path that begins with ***/Volumes***.
file_name	Name of the file.
fs_id	Contains data in that resembles the following: /.file/id=6562758.1769965 The first number after the "=" is an identifier created for the volume. The second number, after the ".", is a file identifier, the Catalog node ID (CNID) [2] of the file if the file resides on an HFS+ or HFSX volume. CNIDs are unique and sequential on HFS+ and HFSX volumes. The CNID can be used to link the entry to a specific file on the referenced volume. For media using file systems other than HFS+ or HFSX, it may not be possible to link the entry to a specific file with this file identifier.
version	This field contains a binary property list (plist). Additional information about the contents of this plist is detailed below.

The figure below shows how the ***files*** table is displayed in BlackLight. The information described above in Table 2 can be seen. Note the entries that have a path beginning with */Users* in the “folders” field. The identifier created for the volume, seen in “fs_id”, is 6562758. Entries that have a path beginning with */Volumes* have different volume identifiers. Figure 5 shows several other volumes, including:

Path Listed in “folders”	**Description of Media**
/Volumes/Data Sets	Network connected share
/Volumes/SHARED	Thumb drive
/Volumes/Shoes	Optical media (CD-R)

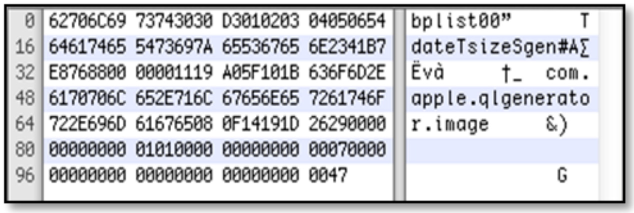


rowid ▲	folder	file_name	fs_id	version
3177	/Users/[redacted]/Desktop			
3178	/Users/[redacted]/Desktop	Screen Shot 2013-09-24 at 5.09.34 P...	/.file/id=6562758.1769969	bplist00"
3179	/Users/[redacted]/Desktop	Screen Shot 2013-09-24 at 5.10.02 P...	/.file/id=6562758.1769980	bplist00"
3180	/Users/[redacted]/Desktop	Screen Shot 2013-09-24 at 5.30.22 P...	/.file/id=6562758.1770113	bplist00"
3181	/Volumes/Data Sets	dates.txt	/.file/id=7192492.20	bplist00"
3182	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.09.14 P...	/.file/id=6562758.1773883	bplist00"
3183	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.09.30 P...	/.file/id=6562758.1773904	bplist00"
3184	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.09.43 P...	/.file/id=6562758.1773913	bplist00"
3185	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.10.00 P...	/.file/id=6562758.1773922	bplist00"
3186	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.10.19 P...	/.file/id=6562758.1773932	bplist00"
3187	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.10.35 P...	/.file/id=6562758.1773941	bplist00"
3188	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.10.49 P...	/.file/id=6562758.1773950	bplist00"
3189	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.11.04 P...	/.file/id=6562758.1773959	bplist00"
3190	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.11.23 P...	/.file/id=6562758.1773968	bplist00"
3191	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.41.25 P...	/.file/id=6562758.1774090	bplist00"
3192	/Users/[redacted]/Desktop	Screen Shot 2013-09-27 at 12.41.45 P...	/.file/id=6562758.1774102	bplist00"
3193	/Volumes/SHARED/131086	Screen Shot 2013-09-27 at 12.41.45 P...	/.file/id=7546399.46	bplist00"
3194	/Volumes/SHARED/131086	Screen Shot 2013-09-27 at 12.41.25 P...	/.file/id=7546399.47	bplist00"
3195	/Volumes/Shoes	Shoe2.jpeg	/.file/id=7625819.18	bplist00"
3196	/Volumes/Shoes	Shoe3.jpeg	/.file/id=7625819.19	bplist00"

Figure 5: Example of *files* Contents as Seen in BlackLight Preview Pane

The “version” Field

The “version” field is used to track the file the thumbnail was created from using the size and modified time of the file. Looking at the binary contents of the “version” field, the following information can be seen:

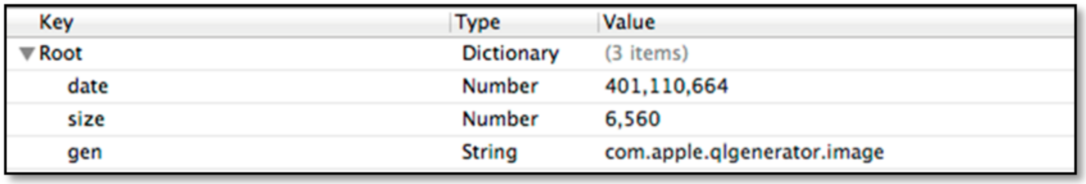


```
 0  62706C69 73743030 D3010203 04050654   bplist00"      T
16  64617465 5473697A 65536765 6E2341B7   dateTsizeSgen#A∑
32  E8768800 00001119 A05F101B 636F6D2E   Ëvà       †_  com.
48  6170706C 652E716C 67656E65 7261746F   apple.qlgenerato
64  722E696D 61676508 0F14191D 26290000   r.image      &)
80  00000000 01010000 00000000 00070000
96  00000000 00000000 00000000 0047                    G
```

Figure 6: Example Contents of a Binary Property List

The ASCII within a binary plist shows information about the keys in the plist. Notice the words *date*, *size,* and *gen* in Figure6.

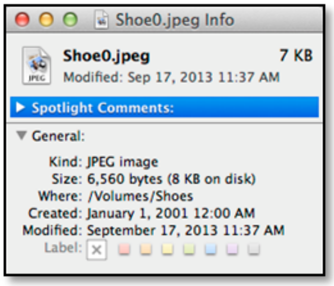
Opening up the plist in Xcode Version 4.6.2 the following data can be seen:



Key	Type	Value
▼Root	Dictionary	(3 items)
date	Number	401,110,664
size	Number	6,560
gen	String	com.apple.qlgenerator.image

Figure 7: Contents of the Binary Plist from the Version Field

The three keys, *date*, *size,* and *gen* are clearly seen, as are the values stored in the keys. OS X commonly tracks timestamps inside plists in numerical formats that reference the number of seconds since a specific date and time, often referred to as an epoch time. This particular plist uses a Cocoa/WebKit reference date. The Cocoa/WebKit time technically has a 1 Jan 1970 epoch plus a constant offset that puts zero time at 1 Jan 2001. Since it uses the constant offset from an epoch time, it is referred to as a reference date. The Cocoa/WebKit reference date 401,110,664 located within the date field of the binary plist converts to 17 Sep 13 11:37:44 UTC [1]. Below is file information shown by OS X for the file that generated the entry containing the plist shown in Figures 12 and 13 in the "version" field.



Shoe0.jpeg Info

Shoe0.jpeg 7 KB
Modified: Sep 17, 2013 11:37 AM

Spotlight Comments:

General:

Kind: JPEG image
Size: 6,560 bytes (8 KB on disk)
Where: /Volumes/Shoes
Created: January 1, 2001 12:00 AM
Modified: September 17, 2013 11:37 AM
Label:

Figure 8: File Information

The modified time and the file size match the information stored in the binary plist in the "version" field. With the modified time of the file stored in the binary plist, the OS can detect when a file has been modified and needs a new thumbnail generated. The *gen* key in the binary plist tracks the QuickLook "plug in" used to create the thumbnail.

The *thumbnails* Table

The ***thumbnails*** table will show the sizes of the thumbnails created for entries in the ***files*** table. One entry in the ***files*** table may have multiple entries in the ***thumbnails*** table if multiple views were used or settings were changed in Finder. Entries in ***thumbnails*** are linked with the ***files*** table using the fields "rowid" from ***files*** and "file_id" from ***thumbnails***. Some of the other fields in ***thumbnails*** relate to:

- Size of the thumbnail generated
- Number of times the thumbnail was accessed
- Last date and time the thumbnail was accessed

Table 3 shown below provides a description of some of the fields that may be useful when interpreting data stored in the ***thumbnails*** table:

Table 3: Fields Contained in *thumbnails* Table

Field Name	Description
file_id	An identifier that links the entry using the field "rowid" from the ***files*** table. Without this information there is no way to link the entries in this table to a file path or file name.
size	A one dimensional size for the thumbnail created. When custom thumbnails are created to be used as icons in Finder the "width" and "height" will both be the same as the size, and "icon_mode" will be 1.
icon_mode	Shows whether the entry is for a Finder icon. The only entries that are not icons are previews generated for Cover Flow. These previews may not be square. The "width" or the "height" may not be equal to the "size" field, but at least one will. The value stored in this field will be 0 for Cover Flow previews.
hit_count	The number of times the thumbnail has been accessed.
last_hit_date	The date and time, stored in Cocoa/WebKit time, when the thumbnail was last accessed.
width	Width of the thumbnail generated.
height	Height of the thumbnail generated.

Though the data in the ***files*** table and the ***thumbnails*** table can be linked, there may be many entries in the ***files*** table that do not currently have entries in the ***thumbnails*** table. Information is not retained in the ***thumbnails*** table as long as information from the ***files*** table. If there is an entry in the ***files*** table without an entry in the ***thumbnails*** table, the directory the file was in was viewed in Finder. No additional information can be determined. If there is an entry in the ***thumbnails*** table for a file of interest, it is possible to determine:

- The date and time the thumbnail was last shown ("last_hit_date")
- Total number of times the thumbnail was shown ("hit_count")
- Possibly if a Finder view other than Icon view was used for the directory, depending on the file type
- If Cover Flow was used to view the folder

In the ***thumbnails*** table, multiple entries for the same file indicate different sizes of the thumbnail were generated. This can happen if different Finder views are used, or if a user changes the Icon view thumbnail size for the directory. When changing the Icon size, a slide bar is provided.

Analysis of the QuickLook Thumbnail Cache Database

The *index.sqlite* database can grow very large. The ***files*** table can contain thousands of records. The ***thumbnails*** table contains fewer entries, but needs to be linked with the ***files*** table. Keep in mind the process for parsing information from the binary plist stored in the ***files*** table is very manual. Whenever possible, use the CNID to link an entry to a file instead of parsing this information. For files that are no longer on the system, parsing the binary plist will provide the size and the modified date of the file. If information for the files of interest stored in the ***thumbnails*** table, it can be used to determine the last time the thumbnail for a particular file was shown and the number of times the thumbnail was accessed. Though an Excel spreadsheet can be used, the manual work of matching entries from the ***files*** and ***thumbnails*** tables can be eliminated by importing exported spreadsheets into a database and setting up the relationship between the tables.

Sample Analysis

To create this sample, a new account was created in OS X, actions were documented, and the results of those actions on *index.sqlite* were reviewed.

When looking at actual data, the first step is to identify the files of interest. For this example files of interest were created on two CDs with the volume labels *Pictures* and *Shoes*. The *Pictures* CD contains four directories in the root of the volume: *Choos*, *Degas*, *Monet*, and *Photos*. The *Shoes* CD contains ten JPEG files at the root of the volume. The information stored in the ***files*** table created by OS X when these volumes were accessed was exported to a csv file from BlackLight.

The entries with "folder" paths beginning with */Volumes*, indicate the entries relate to a volume other than the OS volume. In OS X mounted media may include additional volumes on internal hard drive(s), USB thumb drives, external hard drives, optical media, network connected shares, and disk image (dmg) files. The media used to create the */Volumes/Pictures* and */Volumes/Shoes* entries were both optical media, but to determine the exact type of media during an examination look at other OS artifacts and, if possible, examine the media that created the entries.

If during an examination it is determined that the media used to create these entries is formatted with HFS+, or HFSX, the CNIDs in the "fs_id" field should be used to link the entries to a specific volume on the media. This can only be done if there is media with a volume to compare. For volumes not formatted with HFS+ or HFSX, the volume name, files and folders listed for the entire volume, and information stored in the "version" field of the ***files*** table can be used to link the entries to specific media; the version field tracks the modified date and size of the file. Similarly to the situation when a file is no longer present in the active file system, for missing media the "version" field may provide a timeframe of activity.

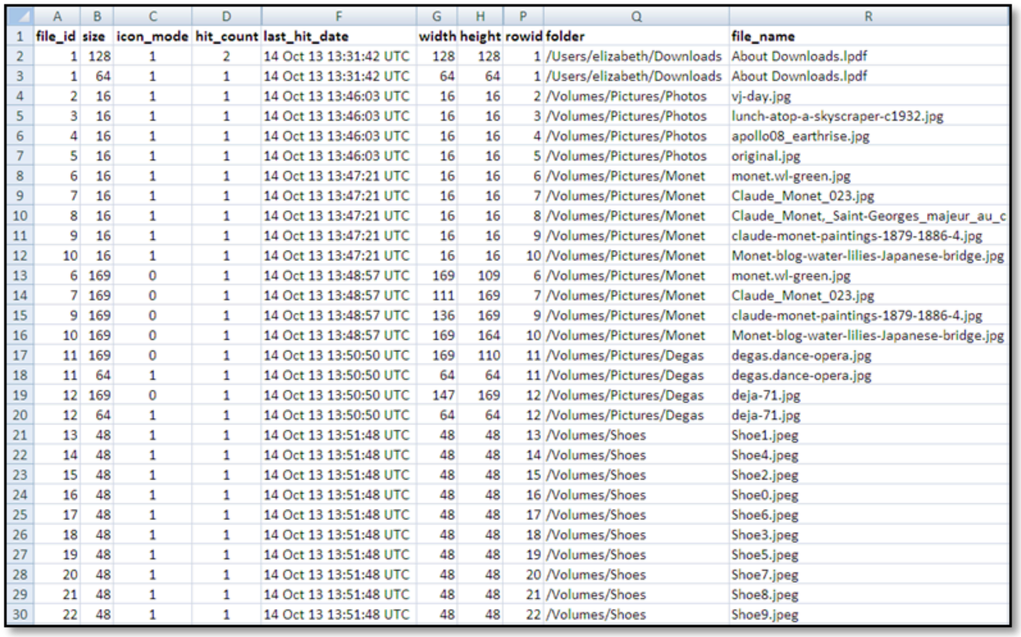
In this example optical media was used, which is not formatted in HFS+ or HFSX. Multiple pieces of information can be used to link the entries in the ***files*** table and the media using the following information:

- Volume name

  - Using the volume labels *Pictures* and *Shoes*
- Entries are seen for some of the directories and files on the volumes
  - *Pictures* contains the directories *Photos*, *Monet* and *Degas*
  - *Shoes* contains the files *Shoe0.jpeg – Shoe9.jpeg* on the root of the volume
- Confirming the last modified timestamp and size of one or more of the files

Exporting the ***files*** table from BlackLight into a csv file exports the "version" field, but the values were converted to ASCII. To access the hexadecimal values of the binary plist stored in this field tag the entries in BlackLight, export a report of the tagged entries to a MS Word file, copy the hexadecimal values into a hex editor and save them as a .plist, and open the newly created plist file with Xcode. All file sizes and modification dates were parsed from the binary plists in the "version" field and added to the csv file. The information parsed for file size and modification date matches the files on the original media.

The next step is to look at the information stored in the ***thumbnails*** table. To merge the information, entries from the ***files*** table were matched with information in the ***thumbnails*** table, matching "rowid" with "file_id". Some fields were hidden in Excel to reduce the amount of information shown.



	A	B	C	D	F	G	H	P	Q	R
1	file_id	size	icon_mode	hit_count	last_hit_date	width	height	rowid	folder	file_name
2	1	128	1	2	14 Oct 13 13:31:42 UTC	128	128	1	/Users/elizabeth/Downloads	About Downloads.lpdf
3	1	64	1	1	14 Oct 13 13:31:42 UTC	64	64	1	/Users/elizabeth/Downloads	About Downloads.lpdf
4	2	16	1	1	14 Oct 13 13:46:03 UTC	16	16	2	/Volumes/Pictures/Photos	vj-day.jpg
5	3	16	1	1	14 Oct 13 13:46:03 UTC	16	16	3	/Volumes/Pictures/Photos	lunch-atop-a-skyscraper-c1932.jpg
6	4	16	1	1	14 Oct 13 13:46:03 UTC	16	16	4	/Volumes/Pictures/Photos	apollo08_earthrise.jpg
7	5	16	1	1	14 Oct 13 13:46:03 UTC	16	16	5	/Volumes/Pictures/Photos	original.jpg
8	6	16	1	1	14 Oct 13 13:47:21 UTC	16	16	6	/Volumes/Pictures/Monet	monet.wl-green.jpg
9	7	16	1	1	14 Oct 13 13:47:21 UTC	16	16	7	/Volumes/Pictures/Monet	Claude_Monet_023.jpg
10	8	16	1	1	14 Oct 13 13:47:21 UTC	16	16	8	/Volumes/Pictures/Monet	Claude_Monet,_Saint-Georges_majeur_au_c
11	9	16	1	1	14 Oct 13 13:47:21 UTC	16	16	9	/Volumes/Pictures/Monet	claude-monet-paintings-1879-1886-4.jpg
12	10	16	1	1	14 Oct 13 13:47:21 UTC	16	16	10	/Volumes/Pictures/Monet	Monet-blog-water-lilies-Japanese-bridge.jpg
13	6	169	0	1	14 Oct 13 13:48:57 UTC	169	109	6	/Volumes/Pictures/Monet	monet.wl-green.jpg
14	7	169	0	1	14 Oct 13 13:48:57 UTC	111	169	7	/Volumes/Pictures/Monet	Claude_Monet_023.jpg
15	9	169	0	1	14 Oct 13 13:48:57 UTC	136	169	9	/Volumes/Pictures/Monet	claude-monet-paintings-1879-1886-4.jpg
16	10	169	0	1	14 Oct 13 13:48:57 UTC	169	164	10	/Volumes/Pictures/Monet	Monet-blog-water-lilies-Japanese-bridge.jpg
17	11	169	0	1	14 Oct 13 13:50:50 UTC	169	110	11	/Volumes/Pictures/Degas	degas.dance-opera.jpg
18	11	64	1	1	14 Oct 13 13:50:50 UTC	64	64	11	/Volumes/Pictures/Degas	degas.dance-opera.jpg
19	12	169	0	1	14 Oct 13 13:50:50 UTC	147	169	12	/Volumes/Pictures/Degas	deja-71.jpg
20	12	64	1	1	14 Oct 13 13:50:50 UTC	64	64	12	/Volumes/Pictures/Degas	deja-71.jpg
21	13	48	1	1	14 Oct 13 13:51:48 UTC	48	48	13	/Volumes/Shoes	Shoe1.jpeg
22	14	48	1	1	14 Oct 13 13:51:48 UTC	48	48	14	/Volumes/Shoes	Shoe4.jpeg
23	15	48	1	1	14 Oct 13 13:51:48 UTC	48	48	15	/Volumes/Shoes	Shoe2.jpeg
24	16	48	1	1	14 Oct 13 13:51:48 UTC	48	48	16	/Volumes/Shoes	Shoe0.jpeg
25	17	48	1	1	14 Oct 13 13:51:48 UTC	48	48	17	/Volumes/Shoes	Shoe6.jpeg
26	18	48	1	1	14 Oct 13 13:51:48 UTC	48	48	18	/Volumes/Shoes	Shoe3.jpeg
27	19	48	1	1	14 Oct 13 13:51:48 UTC	48	48	19	/Volumes/Shoes	Shoe5.jpeg
28	20	48	1	1	14 Oct 13 13:51:48 UTC	48	48	20	/Volumes/Shoes	Shoe7.jpeg
29	21	48	1	1	14 Oct 13 13:51:48 UTC	48	48	21	/Volumes/Shoes	Shoe8.jpeg
30	22	48	1	1	14 Oct 13 13:51:48 UTC	48	48	22	/Volumes/Shoes	Shoe9.jpeg

Figure 9: Combined Information from *files* and *thumbnails* Tables

With the all of the information in one place (database or spreadsheet), it is easy to sort or query the data to focus on the information of interest. Figure 9 shows the information sorted by the column "last_hit_date" to see a timeline of events. The following actions were performed explaining the information seen in the *index.sqlite* database:

- The first two rows are for the default file in the *Downloads* folder and stack on the Dock *About Downloads.lpdf*, created when the user account first logged in.
- The */Volumes/Picture* CD was opened in Finder, which defaulted to Icon view, showing the four directories: *Choos*, *Degas*, *Monet* and *Photos*. No entries were created since there are only directories at the root of the volume; custom thumbnails are not created for directories.
- The view was changed to List view and the *Photos* directory was expanded. Small (16 x 16) custom thumbnails were created for the four files in the directory, creating rows 4 to 7.
- The Finder view was changed to Column view and the *Monet* folder was navigated to. Small custom icons were created for the five files in the directory, rows 8 to 12.

- The file *monet.wl-green.jpg* was highlighted in the Column view creating a preview of the file. No entry was made in the *thumbnails* table.
- The view was changed to Cover Flow, generating 13 to 16. One of the files in the directory did not have an entry created, though a custom thumbnail was observed in Cover Flow.
- Finder was closed, and then reopened. */Volumes/Pictures/Degas* was opened and defaulted to Cover Flow, the view was changed to Icon view. These actions created rows 17 to 20.
- The directory */Volumes/Pictures/Choos* was never opened in Finder and there are no entries for the files from this directory in the *index.sqlite* database.
- From the Desktop, */Volumes/Shoes* was opened in Finder; defaulting to Icon view rows 21 to 30 were created.

After this initial exercise, additional tests were performed in the user profile to generate follow-on data for analysis. Entries were created when the preferences for Icon size was changed with */Volumes/Pictures/Degas* open. The slider was dragged across without lifting the mouse button from 64 x 64 to 204 x 204. The action was performed very slowly, taking approximately two minutes, so that changes in timestamps could be observed. Some entries had multiple hit counts. Human precision is less than perfect with a mouse and attempting to drag across very slowly resulted in hitting some sizes more than one time.

CONCLUSIONS

All files on the Desktop are viewed as Icons and will have a custom thumbnail generated. Files in folders that have a stack on the Dock will have a custom thumbnail generated as well. If a file has an entry in the files table the OS has displayed a thumbnail of the file in Finder or it was shown in a stack on the Dock. What does not create or update entries in the QuickLook Thumbnail Cache database includes:

- File previews created by highlighting a file in Column view
- Viewing file information
- Opening a file in a QuickLook window

On a system where user accounts had been deleted, the database was not deleted. Information regarding the activities of deleted user accounts could be determined. Reviews of systems with historical information showed entries were not deleted from the ***files*** table rapidly. This provides a historical record for folders and files that were previously accessed. The entire file is tracking navigation through file systems for each user on the system.

If there is an entry in the ***thumbnails*** table, information about the number of times the thumbnail was accessed, how many times the directory was viewed, and the date and time when it was last viewed can be ascertained. If there are multiple entries in the ***thumbnails*** table, multiple Finder views were used or the icon size was changed in Icon view.

Additional questions that could be answered by further research:

Are there size or time limitations for this file?
When do entries get deleted from the tables?
Are there differences in how this data is stored the latest version(s) of OS X?

REFERENCES

1. BlackBag Training Team (2013). *Mac Forensics Tips and Tricks: The Epoch Converter Utility* [online]. Available: https://www.blackbagtech.com/blog/2013/04/08/mac-forensics-tips-and-tricks-the-epoch-converter-utility
2. Singh Amit (2013). *Mac OS X Internals: A System Approach.* ISBN 0-321-27854-2.