

hp Under My Thumbs

- Revisiting Windows thumbnail databases and some new revelations about the forensic implications.

When Windows Vista arrived all the forensic reviews talked of the new thumbcache files and made no mention of thumbs.db files as if they did not exist anymore on Vista; this is not the case and thumbs.db is still an artefact to be found in Vista and Windows 7. This paper revisits the forensic implications of the thumbs.db and thumbcache files and challenges one of the standard implications drawn from the presence of thumbnail databases.

<http://computerforensics.parsonage.co.uk>

Introduction

In those difficult cases where an examiner has to scrape together the smallest pieces of evidence to form a case Windows thumbnail databases often prove to be very useful. When Windows Vista arrived one of the new artefacts mentioned in forensic reviews was the thumbcache files which were said to replace the thumbs.db files found in earlier versions of Windows. There was never any mention that the thumbs.db file did still exist in Vista under certain circumstances, in fact at the time of writing one Forensics Wiki states “thumbs.db no longer exists in Vista/7 as individual files”. This paper takes a new look at the forensic implications of Windows thumbnail databases and uncovers some surprising findings which result in a challenge to one of the main implications drawn from the presence of Windows thumbnail databases.

A quite irrelevant aside, rock fans may recognise a likeness in the title to “Under My Thumb” a 60’s hit for the Rolling Stones, covered by The Who and many others, it was also a Northern Soul classic by Wayne Gibson and it just happened to be on the radio as I started to write this paper.

A little revision on windows thumbnail databases

I will not go into detail about the well-known aspects of Windows thumbnails as they are covered in many previous papers and presentations concerning Windows artefacts but I will summarise the fundamental facts.

The thumbs.db file was the place where Windows XP and earlier versions stored thumbnails and the thumbs.db file was created in the same folder in which the pictures represented by the thumbnails resided.

When Windows Vista arrived it was said the thumbnail database was no longer named thumbs.db and was stored on a per user basis in the following folder –

C:\Users\{UserName}\AppData\Local\Microsoft\Windows\Explorer

The files containing the thumbnails in this folder are named according to the maximum pixel size of the thumbnails –

thumbcache_32.db

thumbcache_96.db

thumbcache_256.db

thumbcache_1024.db

These thumbcache files are accompanied by two other files, thumbcache_sr.db and thumbcache_idx.db, the latter in conjunction with the Windows Search Database keeps track of the thumbnails and the location of the original full-sized files (1).

In Windows Vista and Windows 7 the menu item View\Thumbnails is no longer used and the option is Views\Large Icons or similar (Figure 1). In this paper I will continue to use the term thumbnail view

to mean a view where thumbnail pictures are displayed in Windows Explorer which will be as a result of choosing Extra Large, Large or Medium Icons. I have also referred exclusively to thumbnails of pictures but I note that thumbnails are created of other types of file and these too will be found in thumbnail databases.

Forensic value of thumbnails

The two main points of value often raised by the presence of Windows thumbnail databases are –

- 1) A large proportion of computer users have no knowledge of the presence of Windows thumbnail databases so that whilst they might delete incriminating pictures the evidence of their illicit activity often remains in the thumbnail databases.
- 2) The presence of pictures in a Windows thumbnail database is taken as an indicator of guilty knowledge; for the pictures to exist in the thumbnail database the folder containing the pictures must have been opened in Windows Explorer in a thumbnail view thus implying that the user must have knowledge of them.

Nothing in this paper will challenge the first point but the second point will be one under discussion.

Some practical experiments to test in what circumstances thumbnails are created

Using Windows Vista, the first part of the experiment is to start off with clean thumbcache files by creating a new user and then logging on and logging off. This will create a clean set of thumbcache files, clean in the sense that the files should not contain any thumbnail pictures. If you are to try this experiment then keep a copy of the thumbcache files in a logical evidence file if you can, and check them before you go any further to see if there are any thumbnail pictures in the “clean” files. It is surprisingly easy to populate the thumbcache files so each step below is aimed at trying to prevent this. In fact my first attempt to create a clean set of thumbcache files failed as the default new user on my Vista machine had the Windows Sidebar on and this contained the gadget “Slide Show” which pointed to Sample Pictures resulting in thumbnails of the Sample Pictures in the Slide Show ending up in the thumbcache_256 file.

Log in to your new user profile and in your Public\Pictures folder set the View to Details (Figure 1) and make sure you access the Public folder via your Network so that the folder path when you click in the address bar is shown as a full UNC path (Figure 2).

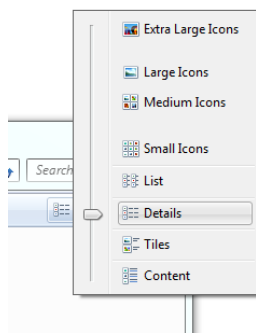


Figure 1

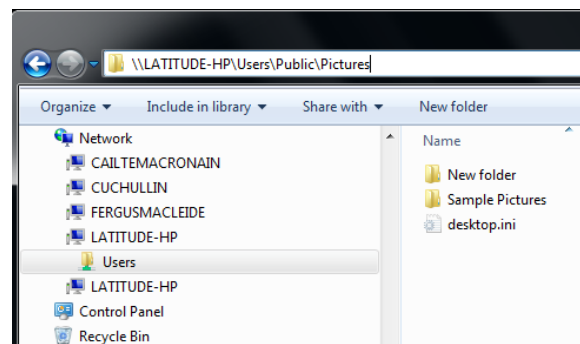


Figure 2

Check the properties of the “New folder” and under the Customize tab ensure it is set to Documents (Figure 3); this should prevent the folder automatically displaying thumbnails when you open it.

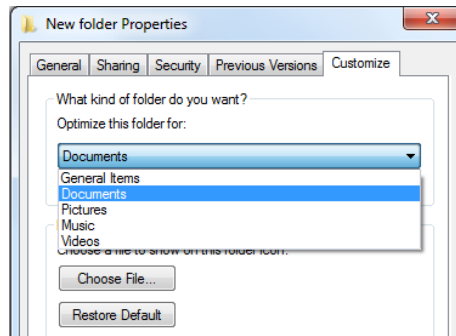


Figure 3

In Windows Explorer select the “New folder” and then from the menu bar select Organize/Layout/Details Pane to switch on the Details Pane (Figure 4). At the bottom of Windows Explorer on the left you will see the Details Pane marked here in red.

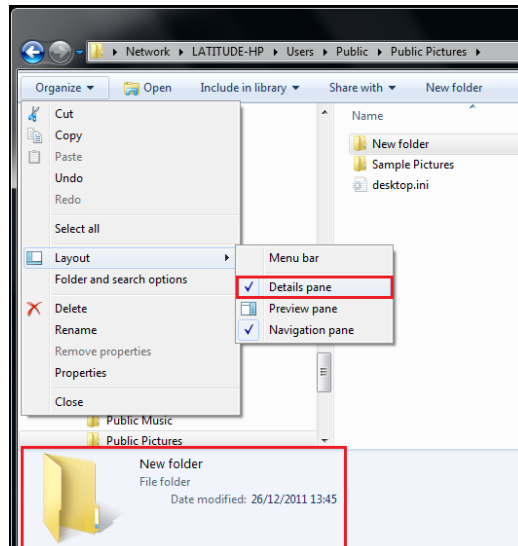


Figure 4

On another computer create a folder containing two .jpg files, two .bmp files and two .png files and copy this folder onto a thumb drive. Insert the thumb drive into your experimental computer and open Windows Explorer at the root of the thumb drive. Check the properties of the folder containing the pictures and under the Customize tab ensure it is set to Documents; this should prevent the folder displaying thumbnails when you open it.

Open the folder containing your six test pictures and drag and drop the two .bmp pictures into your “New folder” in your Public Pictures (Figure 5).

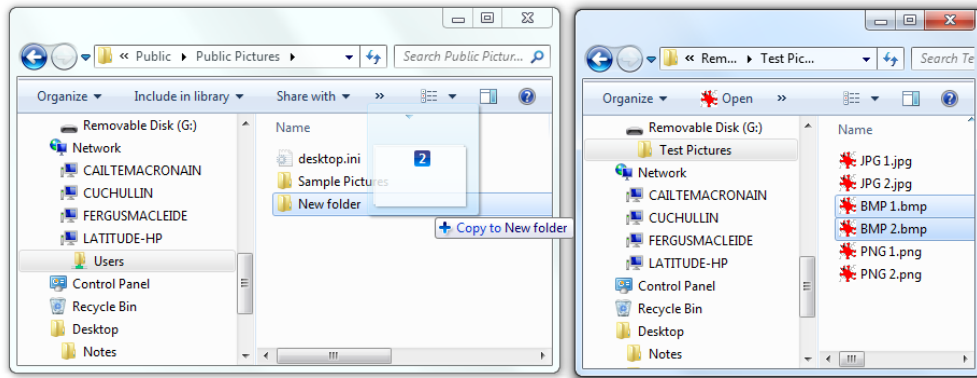


Figure 5

Now open a command prompt at the “New folder” and delete the two bmp pictures (Figure 6); the purpose of doing the experiment this way is to avoid opening the “New folder” in Windows Explorer.

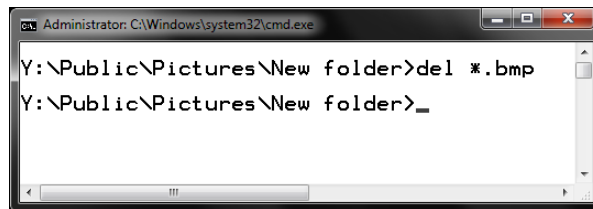


Figure 6

Drag and drop the two.jpg pictures into “New folder” and delete these at the command prompt and then repeat with the two .png files.

Now open the “New folder” and you should find it contains a thumbs.db file (Figure 7).

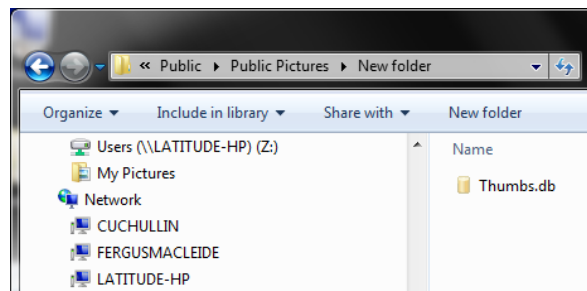


Figure 7

If you then use your choice of program to extract the thumbnails from the thumbs.db file you should find what I found when conducting the same experiment, shown below (Figure 8).

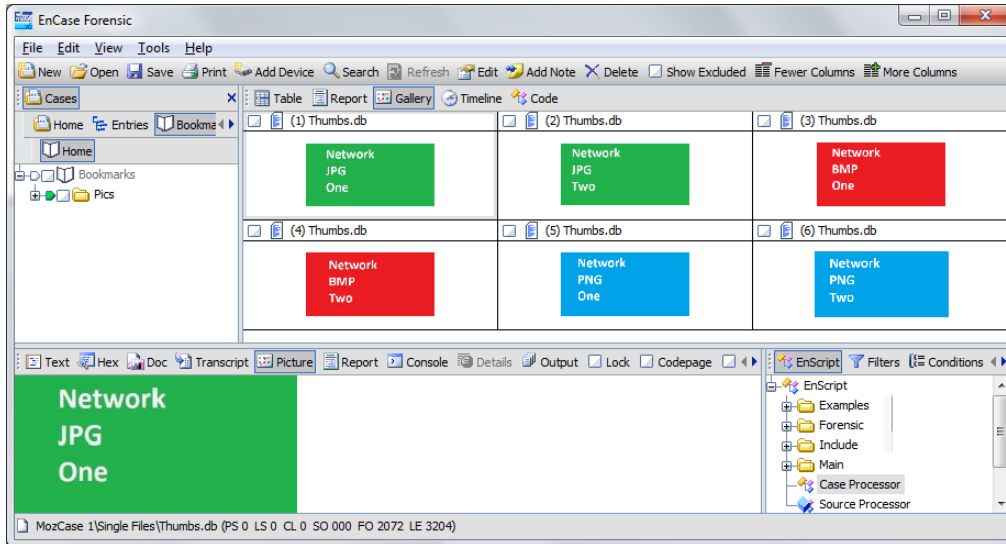


Figure 8

I found that each pair of pictures dropped into the “New folder” had been saved in the thumbs.db file.

In this experiment the “New folder” was not opened in thumbnail view at any time yet all six of the pictures that were in the folder are contained in the thumbs.db file. The reason for this appears to be that the Windows Explorer option to display the Details Pane has caused the creation of the thumbnails. However it seems that only two thumbnails are created regardless of the number of pictures in the folder, but when pictures are deleted from the folder and new pictures are added two new thumbnails will be created.

Whilst this experiment is in some respects quite convoluted it does show as a proof of concept that the assertion that thumbnails are present in a thumbs.db file because the pictures have been displayed in Windows Explorer in a thumbnail view is not accurate.

In Windows Vista and Windows 7 when pictures are accessed via a full UNC path and displayed in thumbnail view (Medium, Large or Extra Large Icons) a thumbs.db file is created in the folder in which the pictures reside. The size of the thumbnail in the thumbs.db file is a 256 size thumbnail regardless of the view chosen i.e. Medium, Large or Extra Large Icons. In the presentation referenced at (1) the author stated “*Existence of a thumbs.db file indicates a folder was remotely accessed*” this is incorrect; the existence of a thumbs.db file indicates that the folder was accessed via a full UNC path (or via the Network icon or a network share) which can be done on a local machine as well as a remote machine.

It is possible to work out which version of Windows the machine viewing the pictures is running because the thumbs.db created differs between XP, Vista, and Windows 7. The actual data stream carrying the picture has a different header (Figure 9) in each version and the XP version of thumbs.db carries the original file name of the picture.

- The first byte in each stream is the size of the header x0C in XP and x18 in Vista and W7,
- The bytes outlined in blue are the size in bytes of the thumbnail picture,

- The eight bytes in red in Vista are the Modified Date of the original picture in Windows Filetime but the small order byte is always x00 in all the streams I have seen.
- In Windows 7 the Filetime is replaced with an eight byte value, and I would be interested to hear from anyone who knows what this is.

XP	0x000	0C00 0000	0100 0000	730E 0000	FFD8 FFE0s...ÿøÿà
	0x010	0010 4A46	4946 0001	0101 0060	0060 0000	..JFIF.....`'..
	0x020	FFDB 0043	0003 0202	0302 0203	0303 0304	ÿÛ.C.....
Vista	0x0000	1800 0000	0300 0000	E527 0000	0100 0000å'.....
	0x0010	00B1 B256	AC6B CC01	FFD8 FFE0	0010 4A46	.±²V~kÏ.ÿøÿà..JF
	0x0020	4946 0001	0101 0000	0000 0000	FFDB 0043	IF.....ÿÛ.C
Windows 7	0x0000	1800 0000	0300 0000	C627 0000	0000 0000Æ'.....
	0x0010	7A8D 6531	B0D0 89CF	FFD8 FFE0	0010 4A46	z e1°ÐÏÿøÿà..JF
	0x0020	4946 0001	0101 0000	0000 0000	FFDB 0043	IF.....ÿÛ.C

Figure 9

Just note that these are current thumbs.db files; be aware that the structure of the thumbs.db and the streams within has changed a number of times in the past.

During experimentation an interesting observation has been made in relation to viewing pictures in thumbnail view in Windows Explorer via a UNC path - pictures are only cached in the thumbs.db file when they have actually been exposed to view. So if there are a large number of pictures in a folder such that it would be necessary to scroll down to see all the pictures it seems that the pictures are not cached unless the user has scrolled down to expose them for view in the window. If a large number of pictures are found in a folder and the thumbs.db file contains more pictures than could have been viewed in just one window then the user must have scrolled the window implying knowledge of the contents of the folder.

Continuing with the experiment, on another computer create a zip file containing two .jpg pictures and copy this to a thumb drive. Insert your thumb drive into your experimental computer and extract the pictures from the zip file, using the Windows built in option to do this, and extract the files into a new folder on your desktop but uncheck the option to show extracted files when complete (Figure 10).

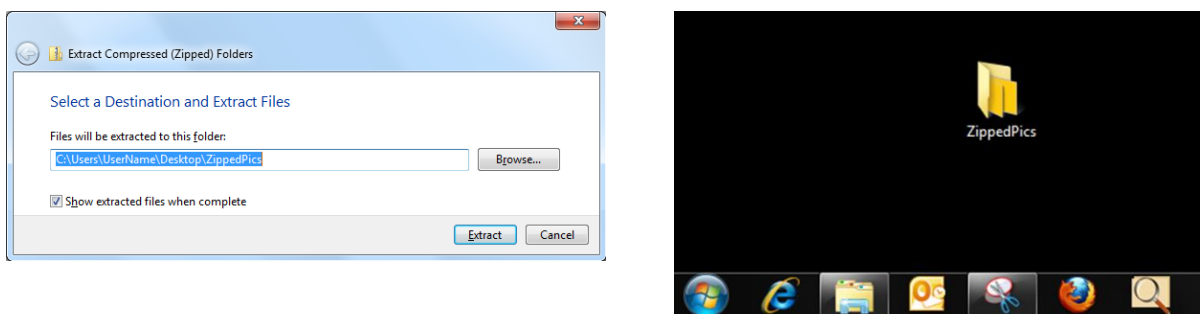


Figure 10

Now take your thumbcache files from the current user's folder and extract the thumbnails from them. This is what I found in my thumbcache_256.db file (Figure 11).

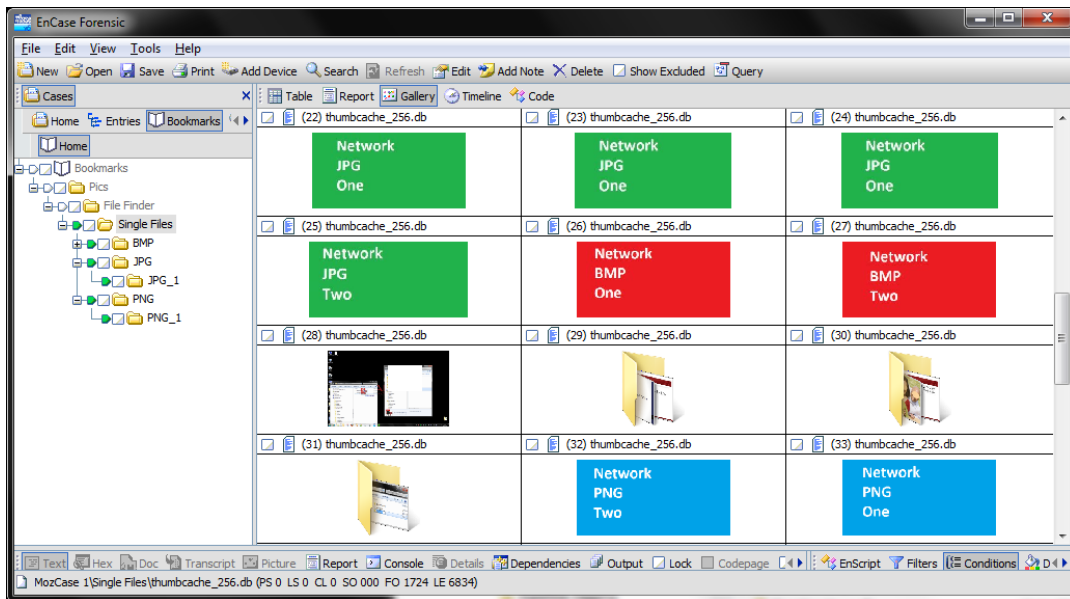


Figure 11

I found the thumbnails for the six pictures I had copied into the “New folder” earlier which were also found in the thumbs.db file. Additionally I found thumbnails for the two pictures which were extracted from the zip file into the folder on the Desktop (Figure 12).

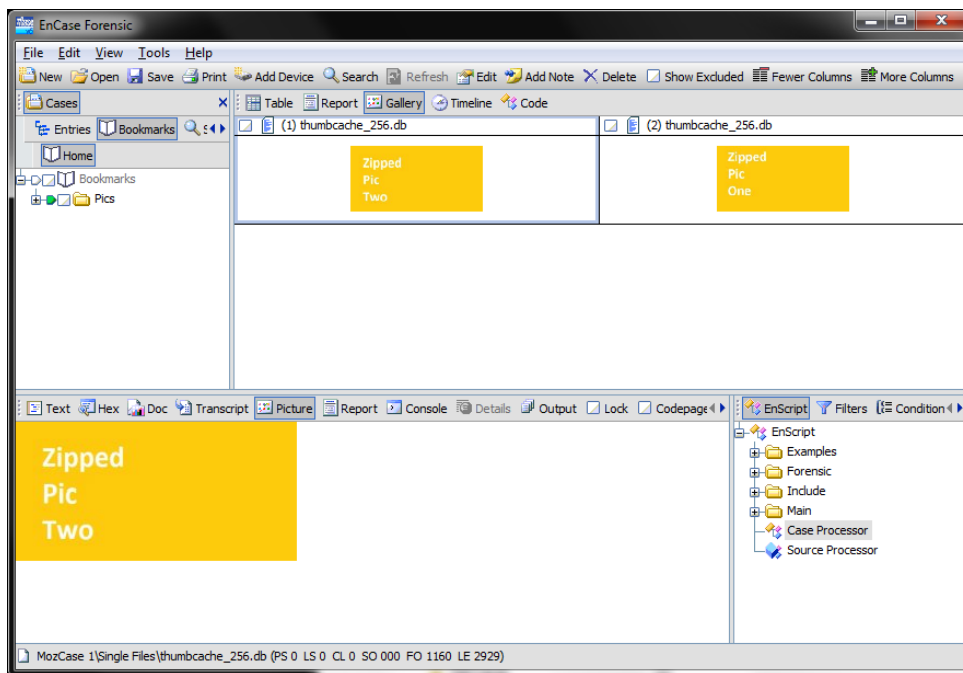


Figure 12

Again the two pictures in the folder on the Desktop had never been viewed in Windows Explorer in thumbnail view; in fact they had never been viewed on the computer at all.

The folder icon on the Desktop for the folder containing the pictures extracted from the zip file has two small representations of the pictures displayed on it. It appears that when these two small

previews on the folder icon are created the two thumbnails are also created in the thumbcache_256.db file.

Try another experiment - dragging and dropping files. When files are dragged and dropped from one folder to another in Windows Explorer, either by moving or copying, when the cursor is moved away from the current location a thumbnail representation of the pictures being copied or moved appears under the cursor as a stack, overlaid with the number of pictures in the activity, this happens regardless of the View setting in the folder (Figure 13).

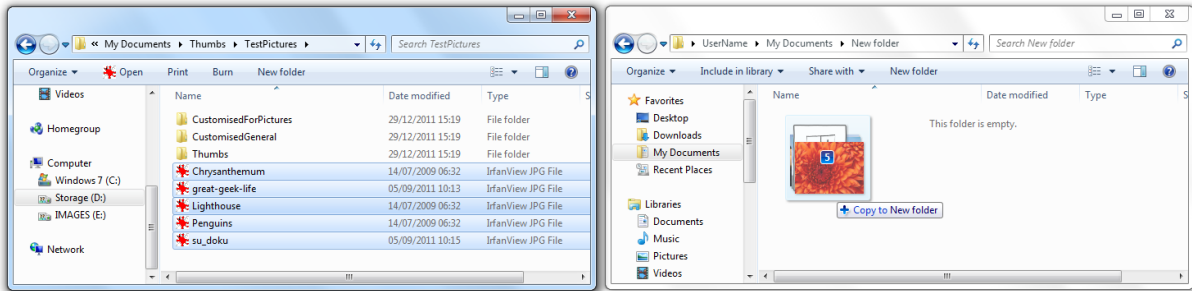


Figure 13

In tests the thumbnails of the pictures being moved or copied have been created in thumbcache_96, even if the copy is cancelled before it is completed by pressing the Escape key. The number of thumbnails created in any one action seems to be limited to a maximum of four, but this is another instance where thumbnails have been created without the user necessarily seeing the content with the exception of the thumbnail at the front of the stack.

Taking the experimentation further, on a thumb drive I created a folder called Test Pictures and within that folder created four sub-folders each containing three pictures, the content of each picture was a graphic "Folder # Pic #" where # represented the folder and picture number (picture 1 in folder 1 is shown in Preview in Figure 14 below). Note that the Preview below is just for the purpose of showing the content of the file and the Preview was not used during the experiment.

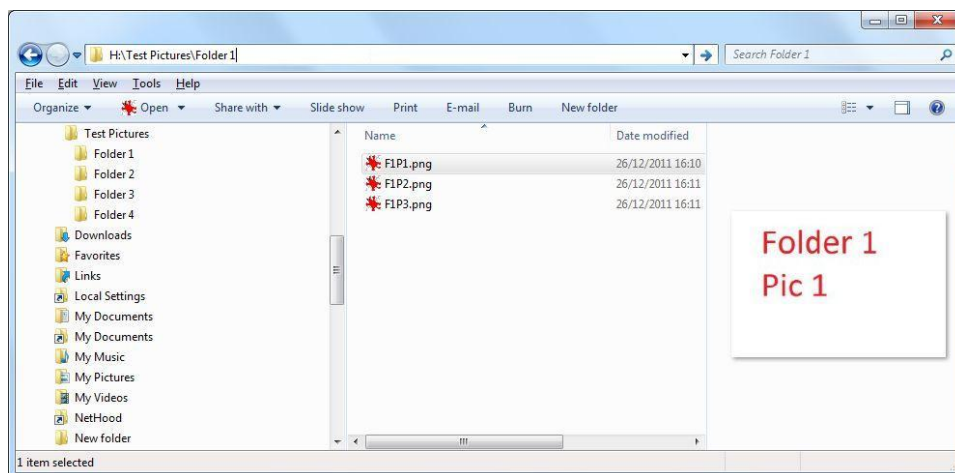


Figure 14

Using a different user profile I copied this folder into the My Documents folder of a new user profile and logged into the new user profile and accessed just the Test Pictures folder and the Test Picture folder was set to “Large Icons”. I did not open any of the sub-folders so the view was as follows (Figure 15).

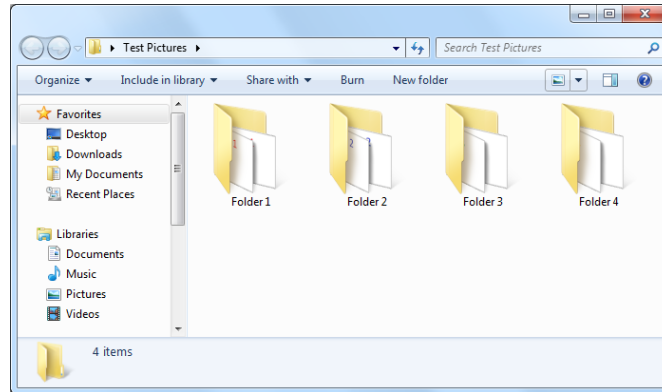


Figure 15

I then extracted the thumbcache_256.db file and found that two pictures from each of the sub-folders had been cached despite the fact that none of them had been displayed in thumbnail view (Figure 16).

(1) thumbcache_256.db	(2) thumbcache_256.db	(3) thumbcache_256.db	(4) thumbcache_256.db
Folder 1 Pic 3	Folder 1 Pic 2	Folder 2 Pic 3	Folder 2 Pic 2
(5) thumbcache_256.db	(6) thumbcache_256.db	(7) thumbcache_256.db	(8) thumbcache_256.db
Folder 3 Pic 3	Folder 3 Pic 2	Folder 4 Pic 3	Folder 4 Pic 2
(9) thumbcache_256.db	(10) thumbcache_256.db	(11) thumbcache_256.db	(12) thumbcache_256.db

Figure 16

So here there are eight thumbnail pictures that have never been viewed yet they are in the thumbcache_256.db file.

The general rule seems to be that the two pictures used for the Folder Icon which are also created as thumbnails are the two most recently modified files in the folder. I have found a very small number of examples of folder icons where this is not the case. If there are modifications to the contents of a folder the pictures displayed on the folder icon will change if any of the modifications result in different pictures having the two most recently modified dates. The tested modifications that affect this are where pictures are deleted, added or existing pictures modified. I also found that the actual

folder icon in the thumbcache file is amended as opposed to a new folder icon being added to the thumbcache file.

Returning to the finding in Figure 16, that eight pictures exist as thumbnails without ever being viewed, the fact is that if had I viewed other pictures in a thumbnail view in Windows Explorer they too would be in the same thumbcache file. In consequence the quandary arises – where the full-sized pictures no longer exist, how can any forensic examiner draw an inference that the user must have knowledge of the pictures in the thumbcache when a proportion of the pictures may have been created automatically possibly without ever being seen? Those thumbnail pictures that have been created automatically will be mixed together with any that have been viewed in a genuine thumbnail view.

The only answer to this quandary that I can suggest at present is that the folder icons in thumbcache_96 which have pictures on the icons should be matched up with pictures in the thumbcache_256 and any remaining pictures in thumbcache_256 that do not relate to a folder icon have not been created automatically.

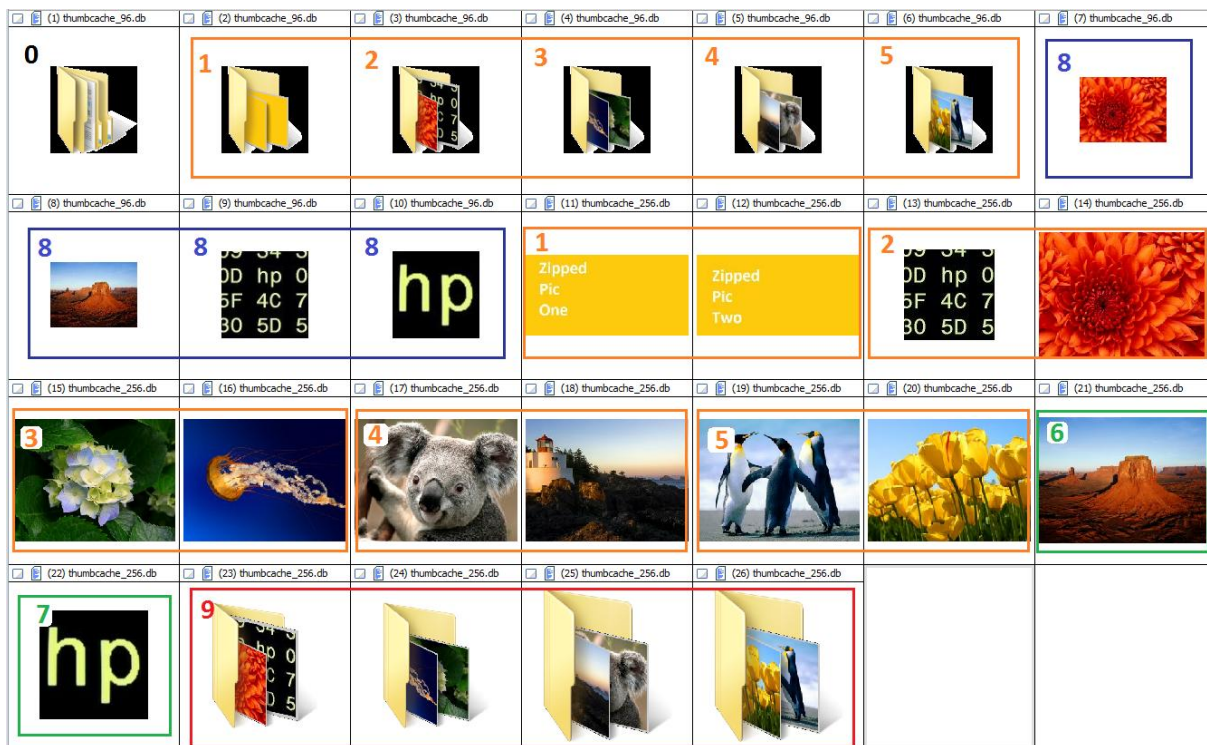


Figure 17

In Figure 17 above on the top line the five folder icons carved from thumbcache_96 marked in orange can be matched up with their respective pairs of icons from thumbcache_256. There remain just two icons in the thumbcache_256 (excepting the last four folder icons) and it might be concluded that these two were viewed in thumbnail view.

In this instance the following actions were undertaken to cause the creation of the thumbnails in Figure 17 –

- 1) Folder Icon (1) was on the Desktop and contained the two files “Zipped Pics”, the folder icon and two thumbcache_256 pictures (1) were created automatically,
- 2) The first Folder Icon (0) was a folder containing just sub folders (2, 3, 4, & 5) and was viewed in Medium Icon View,
- 3) Folder Icon (2) on the top line was a folder which contained four pictures, the two shown on the Folder Icon and the two outlined in green (6 & 7),
- 4) the four folders (2, 3, 4, & 5) were viewed together as sub-folders in Medium Icon View causing the creation of the four folder icons and the respective pairs of pictures in thumbcache_256,
- 5) then Folder 2 was opened and viewed as “Medium Icons” causing the creation of the four pictures outlined in blue (8) in the thumbcache_96,
- 6) then the view was changed to “Extra Large Icons” causing the creation of the two thumbnails marked in green (6 & 7) (two of the four pictures already existed (2) in thumbcache_256),
- 7) then the back button was clicked to return to the four sub-folders and the view was changed to “Extra Large Icons” causing the creation of the four folder icons in thumbcache_256 (9).

Another property of both thumbs.db and thumbcache files that might be useful is the fact that new thumbnails are appended to the existing file and so are a timeline of activity. I have certainly observed that the four folder icons (2, 3, 4, & 5) in Figure 17 are contiguous in the thumbcache_96 file and the respective 8 pictures in the thumbcache_256 are also contiguous. It is possible that some useful conclusions could be drawn from the timeline of activity in the thumbnail databases especially if the folders and pictures are still available.

The final area of the thumbcache files is the thumbcache_1024 file for which I have found difficulty in identifying what causes this file to be populated with thumbnail pictures. I have found that switching on the Preview pane in Windows Explorer will cause a thumbnail to be created in this file if the divider of the Preview Pane is dragged wide enough that the preview changes from a 256 thumbnail to a 1024 size thumbnail, although I have found that not every picture viewed is cached, just some of them. I would be interested to hear any views on how this file is populated.

Summary

It has become clear in conducting these experiments that thumbnail pictures can be created in both the thumbs.db and thumbcache files without those pictures ever being exposed to view by the user. Consequently it is no longer tenable for the assertion to be made that the presence of thumbnail pictures indicates that the pictures have been displayed to the user in thumbnail view in Windows Explorer. Any assertions made about the forensic implications of Windows thumbnail databases need to be carefully considered in light of the above experimental findings.

Finally, the observations described in this paper will only be certain in the context and at the time in which they were made. Experience shows that the dynamic nature of computer technology means the findings can only be a guide and that the observations must be tested in the environment under examination. A wise investigator will never just accept what they read no matter from what source. I hope you will find this paper useful in providing a starting point for your own testing and

observations and if you do carry out some experiments I would be pleased to see the results of your findings. Please take care with your tests as I have found thumbnail databases are susceptible to the slightest change in conditions.

If the information in this paper helps you in an investigation please let me know, I would be pleased to learn my efforts have been put to some practical use, as that was the purpose of writing it.

`computerforensics@parsonage.co.uk`

1. **Larson, Troy.** Windows 7 Thumbnail Cache. *Slideshare*. [Online] October 2010.
<http://www.slideshare.net/ctin/windows-7-forensics-thumbnaildtlr4>.