

# **An Overview of the Jumplist Configuration File in Windows 7**

**Harjinder Singh Lalli**

University of Warwick, International Digital Laboratory (WMG),  
University of Warwick, Coventry, CV4 7AL, UK;  
h.s.lallie@warwick.ac.uk

**Parmjit Singh Bains**

University of Derby, School of Computing and Mathematics, Kedleston  
Road, Derby, DE22 1GB, UK;  
parmjitsinghbains@hotmail.co.uk

## **ABSTRACT**

The introduction of Jumplists in Windows 7 was an important feature from a forensic examiners viewpoint. Jumplist configuration files can provide the examiner with a wealth of information relating to file access and in particular: dates/times, Volume GUIDs and unique file object IDs relating to those files. Some of the information in the Jumplist could be used to build a more precise timeline relating to system and file usage. In this article, we analyse the structure of a Jumplist configuration file and in particular a record from a Jumplist configuration file and highlight some of the important entries therein.

**Keywords:** Jumplists; personal privacy; windows registry; recent documents

## **1. INTRODUCTION**

The introduction of Bitlocker and Jumplists in Windows 7 were possibly the two most important developments in this operating system from a forensic examiners viewpoint. Jumplists were introduced in Windows 7 as part of the *taskbar extensions* (MSDN, 2011a) and are shortcuts to recently or frequently accessed items (referred to as *JumpPaths* or *destinations*) or tasks (referred to as *JumpTasks*) associated with programs or web sites.

Jumplists appear in the Start menu and on the Windows taskbar and provide shortcuts to items (for instance, documents in *Microsoft Word* and songs in *Media Player*) as well as to application specific common tasks (such as 'New task', 'New E-Mail Message', 'New Contact' etc. in Microsoft Outlook). The position in which an item appears on a Jumplist is determined by the application which processes usage data on the items that are used (i.e., the number of times a particular item is opened). The taskbar Jumplist can have four sections: *taskbar tasks* (for instance, allowing users to close the application), *application tasks* (described above), *recent/frequent files* and *pinned items*.

It is important to note that the entries in the *RecentDocs* registry key do not directly correlate or correspond to the entries in the Jumplist for the same application. As an example, if we open a Microsoft Word document (.doc) through Windows Explorer, the position of the entry created in the *RecentDocs* key does not always correlate to the position in the Jumplist. The main difference between the MRUs (Most Recent Used items) in the registry and the entries in the Jumplists is in the amount of information that the Jumplist provides in its configuration file (referred to herein as the *Jumplist configuration file*), which we discuss later in this article.

Jumplists have been used in computer science in binary search trees to make searches more efficient (Andersson & Ottmann, 1991; Brönnimann, Cazals, & Durand, 2003; Ottman, 1991). The Jumplist function in Windows does not necessarily conform to the binary search tree function in that it is not implemented as a search tree, and seems to act more like a ‘database’ that stores entries relating to files and their usage.

Currently there is very little published on the configuration of Jumplist files. Microsoft published developer notes relating to modules that can be used in the management and configuration of Jumplists (MSDN, 2011a, 2011b); however none of this seems to explain the structure of the configuration file. There are a number of software utilities (discussed further in this study) that can help with the management of Jumplists (Hedgehog, 2011; Regdat, 2011). Jumplists (and Windows 7 in general) have also attracted some interest from University research students (Smulikowski, 2009). Furthermore, whilst there are utilities that can help to analyse the structure of Jumplists, it is important that the digital forensics practitioner understand that structure so as to prove the findings of those utilities. Whilst these are useful studies/utilities, none of the studies have explored or analysed the structure of the Jumplist configuration file and there certainly seems to be a gap here which if explored further would be of benefit to the digital forensics community.

In the present work, we intend to further explore the Windows 7 Jumplist function with a particular emphasis of the file configuration and structure. The rest of this article is structured as follows. We begin in section 2 to explain the methodology that we used in this study. In section 3, we explore some of the key differences between Windows 7 and XP – particularly in the context of recent accessed files and then proceed in section 4 to analyse the management and structure of Jumplists and in particular the Jumplist configuration files.

## **2. METHODOLOGY**

The experiment considered in this study was conducted on a new installation of Microsoft Windows 7 (64 bit, service pack 1). A number of files, URLs and programmes were accessed therein so as to create Jumplist entries. We recorded the time/date of the file access as well as the precise file-path-name combination so as to be able to correlate this file access with its entry in the Jumplist

configuration file.

Each time a new file was accessed, we observed that it created a Jumplist entry. Around 10 to 12 entries were created for a number of applications including Microsoft Word, Paint, Internet Explorer, Firefox, Notepad and Windows File Explorer (new directories). Whilst we have referred specifically to the Microsoft Paint Jumplist configuration file throughout this study, we considered a number of Jumplist configuration files for a variety of applications to confirm that the observations made in this study do apply to other applications as well, i.e. that the Jumplist facility did not work differently for different applications.

The Jumplist files were saved and their physical structures/bit streams were analysed using a hex viewer/editor, the hexadecimal structures included in this study are from the examples analysed above. Once the lists were populated and the Jumplist configuration files analysed, with the order of files recorded therein, files already appearing in the Jumplist configuration files were re-opened so as to observe the effect this had on the files position in the Jumplist configuration file.

In addition to this, Jumplist configuration files on other Windows installations (machines that had been used for a while) were considered in order to compare the Jumplist configuration file structures and consider particularly the effect that usage over time has on the list. Two machines - machine (a): 32 bit and (b): 64 bit, were used for this purpose. Machine (a) had an installation date of February 2010 and machine (b) – August 2010. Machine (a) had 67 Microsoft Word entries in the corresponding Jumplist (the most out of all the Jumplists) and machine (b) had 43 Internet Explorer entries. This allowed us to view a larger number of Jumplist configuration file entries and to determine whether the structure is altered over time. There was no evidence of the structure changing over time and it was noted that whilst the Jumplist that the user sees is configured to display a maximum of 10 entries, Jumplist configuration files contain many more entries than this. We could not determine from this research as to whether there was an upper limit to the number of Jumplist entries contained within the configuration file.

Before a Jumplist configuration file was accessed, we created a disk image of the experiment machines, and extracted the Jumplist configuration file extracted from the image and analysed on a separate machine. This is useful from a forensic perspective in that it can provide a historical account of file access.

Elsewhere in this study we outline that there is no official information available from Microsoft regarding the precise structure of the Jumplist entries (notwithstanding the information regarding the function of Jumplists and the advice provided through the MSDN community), as a result we *reverse engineered* the Jumplist configuration files and entries contained therein using the Hex editor as previously described. We acknowledge that this process is not completely fool-proof, therefore this work serves to act as reasonably precise observations and more particularly – work in progress.

### 3. WINDOWS 7 AND XP

We begin by looking at some of the functionality that has previously existed in Windows XP relating to recently used documents/files. Whilst a detailed analysis is certainly outside the scope of this article, it is useful to consider that one of the key changes between operating systems tends to be the changes in directory structures, we present a few examples of that herein to provide some context.

#### 3.1 Recent Docs/MRU

Some of the additions in Windows 7 were originally introduced in Windows Vista. For instance, *InPrivate browsing* was introduced with Vista from Internet Explorer (IE) 8 onwards. When a user uses InPrivate browsing, cookies are cleared when the session is closed, webpage history and associated form data, passwords, auto completes and search histories are not stored.

Whilst Jumplists are a new feature in Windows 7, recent document lists were available in Windows XP (and in fact from Windows 95) and were accessible through a number of mechanisms such as the registry. In Windows 7, recent document lists for the *Microsoft Paint* program (as an example) can be found in the Registry in the following location:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\  
CurrentVersion\Applets\Paint\Recent File List
```

When a user accesses the *Microsoft Paint* program, the application presents recently accessed user files in the *file* menu. The corresponding registry entries do not provide any further information about the file other than its `<drive>:\<path>\<filename>`. The examiner needs to be aware that these entries can be modified and therein change the entries that appear in the applications file list.

The *Recent File List* for most applications can also be found through the registry - for instance recently accessed *Windows Explorer* documents can be accessed through:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\  
CurrentVersion\Explorer\RecentDocs
```

This registry entry is subdivided into numerous file types (.7z, .bmp, .CAB, .Case etc.) and recent documents for each file type can be accessed thereon.

*ComDlg32.dll* contains common dialog boxes used by windows application to store information about files opened and saved on the machine. The last 10 open/saved files including web browser files can be found under:

```
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\  
CurrentVersion\Explorer\ComDlg32\OpenSavePidLMRU
```

Other MRU locations in the registry include: *LastvisitedMRULegacy* and the *RunMRU* key.

In addition to the differences outlined above, there are a number of other subtle differences between Windows 7 and XP that are primarily related to differences in directory locations. For instance, some Windows XP browser artefacts are found at:

```
<drive>:\Documents and Settings\<username>\Local  
Settings\Temporary Internet Files
```

In Windows 7 they are found at:

```
<drive>:\Users\<username>\AppData\Local\Microsoft\  
Windows\Temporary Internet Files
```

The 'Last Active' directory (used to save browser sessions/tabs in case of system failure) in an XP system is stored at:

```
<drive>:\Documents and Settings\<username>\Local  
Settings\Application Data\Microsoft\Internet  
Explorer\Recovery\Active\Last Active
```

In Windows 7 it is:

```
<drive>:\Users\<username>\AppData\Local\Microsoft\  
Internet Explorer\Recovery\Active\Last Active
```

Most of this is due to the *Documents and Settings* folder having been replaced in Windows Vista onwards with the *Users* folder. Similarly, there are differences in the locations of some of the keys such as the user assist key and the location of some un-installation keys.

## **4. JUMPLIST FUNCTIONALITY AND STRUCTURE**

### **4.1 Forensic Value**

It is outside the scope of the present study to consider the anti-forensic issues relating to Jumplists, however it is important to note that Jumplist configuration files can be edited and modified using a hex editor to point to files that do not exist. In particular a configuration file could point to files on an external device that has previously been registered (in the registry). In this case, when the Jumplist is accessed, it will show the added files, but the application may not be able to access them. However, such a modification would need to be precise and must be reflected in the modification of other elements of the Jumplist configuration file which we discuss later in this study.

Nevertheless, the Jumplist configuration files can be relied upon as a record of a

users' activity if taken into consideration with other evidence. If a program is removed from the *Taskbar* or *Startmenu*, its' related Jumplist configuration file entry is not deleted but remains in its relative location.

Jumplist configuration files can be evidence of the use of a particular program and the linked item. For instance a Firefox Jumplist configuration file pointing to a particular website can be evidence that the user used Firefox to access the website. These features are explored further in this article.

Another useful feature of Jumplist configuration files is that a more accurate timeline can be built which may point to particular modifications to the file. An item may have been attached to a specific volume thereby pointing to other volumes that may be of interest. This evidence is not easily available elsewhere. For instance, access to an external device is reflected in the registry, however this does not provide a timeline – the registry simply records the serial number and will not update any other fields to indicate subsequent access. Its record in the Jumplist provides more information and is now timestamp specific.

## **4.2 Managing Jumplists**

The Jumplist feature is enabled by default when Windows 7 is installed. However, if a user accesses the task bar and start menu properties in the control panel there are two options under *privacy* that allow the user to control whether the recently accessed programs and/or the recently accessed items should be stored and displayed. If these options are unchecked and changes are applied, then all the Recent Items, Jumplist and Registry data relating to previous tasks, and applications accessed will be deleted. The same dialogue allows users to alter the number of items that appear in each Jumplist - typically up to 10 items. The Jumplist configuration file would still be available for access (as is any other deleted file) through a recognised tool such as EnCase or FTK.

A number of 'untested' tools such as *Jumplist Backup Restore*, *Jumplist File Extract* (Regdat, 2011) and *Jump-List Launcher* (Hedgehog, 2011) claim to allow the examiner to backup/restore Jumplists and delete particular Jumplist configuration file entries. *Jumplist File Extract* can display the location of the related file; the file name, the time and date it had been modified; time and date it was created; time and date the file was accessed; size of the file, and the machine name i.e., the computer name.

These tools can be useful to a forensic examiner; however, the results would need to be verified using an accepted tool such as EnCase or FTK. To be able to do that, the examiner would need to understand the structure of the file.

Items are added to both the *Recent File List* (registry) and the Jumplist configuration file when it is opened by the user either through a Windows *file open* dialogue box, or selected through *file explorer* (double clicked). Items added to Jumplists are considered to be *recently* or *frequently* accessed items, there is a subtle difference between the two – *frequently* accessed items may appear on a

Jumplist but may not necessarily be *recently* accessed.

An application may use either the generic Windows Explorer dialogue box or its own proprietary *file open* dialogue to allow the user to open files. Where the Windows dialogue is used, Windows launches a module called *SHAddToRecentDocs* which in turn manages the application's Jumplist by adding an entry in the Jumplist configuration file (MSDN, 2011b). If however, an application uses its own *file open* dialogue, then the application is responsible for maintaining the Jumplist entries.

### **4.3 Automatic and custom destinations directories**

Two Jumplist configuration files ('-ms files') are created for each Windows application – an *automatic destinations* file and a *custom destinations* file, these are stored in separate directories. The automatic destinations file for a particular application is populated whenever a user opens a file as described above, this directory can be found at:

```
<drive>:\Users\<username>\Appdata\Roaming\Microsoft\
Windows\Recent \AutomaticDestinations
```

The customs destination file for that application is populated by the application and contains categorised entries for that application, this directory can be found at:

```
<drive>:\Users\<username>\Appdata\Roaming\Microsoft\
Windows\Recent\CustomDestinations
```

Both these directories are accessible in *Windows Explorer* by entering the file path as the location.

Jumplist configuration files have a hexadecimal name (typically 16 digits) followed by an extension which is either *automaticDestinations-ms* or *customDestinations-ms*. The hexadecimal prefix relates specifically to an application, for instance: *28c8b86deab549a1* is an *Internet Explorer* Jumplist configuration file, *74d7f43c1561fc1e* - a *Media Player* Jumplist configuration file, *7e4dca80246863e3* to a *control panel* Jumplist configuration file, *918e0ecb43d17e23* - a *notepad* Jumplist configuration file etc. The filename for the 'automaticDestinations' Jumplist configuration file for *notepad* would therefore be: *918e0ecb43d17e23.automaticDestinations-ms*. Possibly one of the most important Jumplist configuration files is: *1b4dd67f29cb1962.automaticDestinations-ms* which is the *Windows Explorer* Jumplist configuration file, this file points to the creation of particular folders (and the dates thereof).

A third directory of note is the *recent destinations directory* (another hidden directory) which stores links to all recently accessed files. This can be found at:

```
<drive>:\Users\<username>\AppData\Roaming\Microsoft\  
Windows\Recent
```

It is important to note that shortcuts are added to this directory when a file is accessed and a corresponding entry is created in the relevant Jumplist configuration file. If an entry is deleted in either (i.e. the shortcut in the directory or the entry in the Jumplist configuration file), it does not automatically delete the other.

#### **4.4 Structure of a Jumplist**

In this section we present a brief discussion of some of the important data contained within the Jumplist configuration file. The following discussion relates to the *Microsoft Paint* Jumplist configuration file *12dc1ea8e34b5a6.automaticDestinasions-ms* and we provide an analysis of a particular record in that Jumplist. The discussion contained herein applies to all other Jumplist entries in the same file as well as other files including entries that may be found in Jumplists in the custom destinations directory.

Figure 1 displays the record with highlighted fields which represent the following:

- MAC times and date/time that the entry was added to the Jumplist configuration file
- File Size
- Application to which the entry refers
- Volume Serial
- Full Volume path/filename entry
- Computer Name
- Volume GUID
- File Object ID
- File Name and Path

Throughout this discussion we have paired hex digits to represent bytes in order to make the discussion easier to follow. Where necessary, we have used the following character as a form of delimiter: “|” to create distinctions between the fields.



```

4C 00 00 00 01 14 02 Accessed Time 00 00 C0 00 00 Modified Time 00 00 46 File Size 00 Created Time BA 67 BC 4D
5C 0A CB 01 D3 40 7B 4E CB 9C CB 01 BA 67 BC 4D 5C 0A CB 01 7E 48 02 00 00 00 00 00 01 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 85 02 14 00 1F 50 E0 4F D0 20 EA 3A 69 10 A2 D8 08 00 2B 30
30 9D 19 00 2F 41 3A 5C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 90 3E B8 53 10 00 5F 73 75 62 00 00 00 36 00 08 00 04 00 EF BE 8F 3D AE 7E 90 3E B8 53 2A
00 00 00 23 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 14 00 46 00 31 00 00 00 00 00 62 3E C8 79 10 00 73 75 62 00 34 00 08 00 04 00 EF BE 8F
3D C6 2D 62 3E C8 79 2A 00 00 00 EF D7 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 73 00 75 00 62 00 00 00 12 00 56 00 31 00 00 00 00 00 00 8C 3E 07 6A 10 00 70 69 63 74 75 72 65
73 00 00 3E 00 08 00 04 00 EF BE 8F 3D 4D 32 8C 3E 07 6A 2A 00 00 00 6D 5F 02 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 70 00 69 00 63 00 74 00 75 00 72 00 65 00 73 00 00 00 18
00 52 00 31 00 00 00 00 00 41 3E A2 BA 10 00 5F 46 61 6D 69 6C 79 00 3C 00 08 00 04 00 EF BE 8F
3D 4D 32 41 3E A2 BA 2A 00 00 00 EA 75 02 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 5F 00 46 00 61 00 6D 00 69 00 6C 00 79 00 00 00 16 00 4A 00 31 00 00 00 00 00 00 00 00 00 00 00
00 32 30 30 34 00 00 36 00 08 00 04 00 EF BE 8F 3D 2D 33 90 3D B4 15 2A 00 00 00 00 00 00 14 00 02 00 31
00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 30 00 30 00 34 00 00 00 14 00 02 00 31
00 00 00 00 00 00 00 00 15 10 00 32 30 34 5F 30 01 00 00 00 00 00 04 00 EF BE 8F 3D 30 33 90
3D B1 15 2A 00 00 00 7A 02 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 30 00 34 00 5F 00 30 00 31 00 00 16 00 82 00 52 00 7B 78 02 00 00 3D 30 2F 52 20 00 32 30 30
34 5F 30 7E 31 2E 4A 50 47 00 00 00 4C 00 08 00 04 00 EF BE CC 3C E4 92 00 3D B0 16 2A 00 00 00 0F
7A 02 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 32 00 30 00 30 00 34 00 5F 00 30
00 31 00 5F 00 30 00 30 00 37 00 2E 00 6A 00 70 00 67 00 00 00 1C 00 1A 00 00 00 1A 00 00 EF BE 02
00 50 00 42 00 72 00 75 00 73 00 68 00 00 00 <Volume Name><Drive> 00 00 00 00 00 00 00 00 00 00 00 00 00 1C
00 00 00 37 00 Application 00 00 00 71 00 00 00 1B <Path><Filename> 00 00 00 8D 70 1B 0E 10 00 00 00 4E
65 77 20 56 6F 6C 75 6D 65 00 41 3A 5C 5F 73 75 62 5C 73 75 62 5C 70 69 63 74 75 72 65 73 5C 5F
46 61 6D 69 6C 79 5C 32 30 30 34 5C 32 30 30 34 5F 30 31 5C 32 30 30 34 5F 30 31 5F 30 30 37 2E
6A 70 67 00 00 39 00 00 00 09 00 00 A0 2D 00 00 00 00 31 53 50 53 55 28 4C 9F 79 9F 39 4B A8 D0 E1
D4 2D E1 D5 F3 11 00 00 00 07 00 00 00 00 00 0B Computer Name F 00 00 00 00 00 00 00 00 00 00 00 60 00
00 00 03 00 00 A0 Volume GUID 00 00 00 00 00 00 68 61 72 6A 2D 70 63 00 00 00 00 00 00 00 00 00 5C EF
40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95 4F 59 FC 2F A7 6C E0 11 96 42 80 00 60 0F E8 00 5C EF
40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95 4F 59 FC 2F A7 6C File Object ID 80 00 60 0F E8 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01 00 00 00 03 00 00 00 00 00 00 00 00 00 40 40 03 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00 00
14 7E F5 50 56 73 ED A9 5C EF 40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95 96 59 FC 2F A7 6C E0 11
96 42 80 00 60 0F E8 00 5C EF 40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95 96 59 FC 2F A7 6C E0 11
96 42 80 00 60 0F E8 00 68 61 72 6A 2D 70 63 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00 00 00
00 00 80 3F 73 6F 53 A1 C9 00 CC 01 Added to Jumplist Time 0 41 00 3A 00 5C 00 5F 00 73 00 75 00 62 00

```

Figure 1. Sample Jumplist Configuration File Extract

### 4.5 Timestamps and File size

Timestamps in Jumplist configuration files are stored as Windows 64 bit Little Endian values. Modified, Accessed and Created Timestamps (MAC times) appear within the Jumplist configuration file typically after the pattern:

```

4C 00 00 00 01 14 02 00 00 00 00 00 C0 00 00 00 00 00
00 00 46 83 00 00 00 20 00 00 00

```

```

4C 00 00 00 01 14 02 Accessed Time 00 00 C0 00 00 Modified Time 00 00 46 File Size 00 Created Time BA 67 BC 4D
5C 0A CB 01 D3 40 7B 4E CB 9C CB 01 BA 67 BC 4D 5C 0A CB 01 7E 48 02 00 00 00 00 00 01 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 85 02 14 00 1F 50 E0 4F D0 20 EA 3A 69 10 A2 D8 08 00 2B 30

```

Figure 2. MAC times and Filesize

The sequence presented in Figure 2 is the *Created*, *Accessed* and *Modified* time stamps as per the metadata in the original file and corresponds in this example to the following:

```
BA 67 BC 4D 5C 0A CB 01 | D3 40 7B 4E CB 9C CB 01 | BA 67 BC 4D 5C 0A CB 01
Created Time Stamp      | Accessed Time Stamp      | Modified Time Stamp
Sat, 12 June 2010      | Thu, 16 December 2010  | Sat, 12 June 2010
18:23:06 UTC           | 02:45:31 UTC            | 18:23:06 UTC
```

The *accessed* time stamp does not reflect the date/time that the entry was added to the Jumplist configuration file, a fourth entry – which we will refer to as the *added to Jumplist* time does do this. For a given Jumplist entry, the *added to Jumplist* time is presented towards the end of the record as follows:

```
96 42 80 00 60 0F E8 00 | 68 61 72 6A 2D 70 63 00 00 00 00 00 00 00 00 00 03 00 00 00 00 00 00 00
00 00 80 3F | 73 6F 53 A1 C9 00 CC 01 | Added to Jumplist Time | 0 41 00 3A 00 5C 00 5F 00 73 00 75 00 62 00
```

Where the sequence 73 6F 53 A1 C9 00 CC 01 in this case is the date/time: Fri, 22 April 2011 08:45:28 UTC

The entries in the Jumplist are not modified dynamically: when an item is opened using *Microsoft Paint*, the item’s MAC times are recorded within the Jumplist configuration file from the metadata in the item. If the item should subsequently be moved or modified using a different application, the time stamps in the *Microsoft Paint* Jumplist configuration file is not modified to reflect this. The Jumplist configuration file can therefore act as a historical record which might point to movements/modifications of files.

The file size immediately follows the MAC Timestamps and should be read in reverse order, hence the file size 7E 48 02 (Figure 2) should be read as: 02487E which is 149630 bytes.

#### 4.6 Application, Volume Serial, Volume Name, Label, Path, Filename

The name of the application to which the Jumplist configuration file refers appears after the long file name and before the *<volume name><drive>:\<path><filename>*

```
00 31 00 5F 00 30 00 30 00 37 00 2E 00 6A 00 70 | 00 67 00 00 00 1C 00 1A 00 00 00 1A 00 EF BE 02
00 | 50 00 42 00 72 00 75 00 73 00 68 | 00 00 00 <Volume Name><Drive> | 00 00 | <Volume Serial> | 01 00 00 00 1C
00 00 00 37 00 | <Application> | 00 00 00 71 00 00 00 | <Path><Filename> | 00 00 00 | 8D 70 1B 0E | 10 00 00 00 | 4E
65 77 20 56 6F 6C 75 6D 65 00 41 3A 5C 5F 73 75 62 5C 73 75 62 5C 70 69 63 74 75 72 65 73 5C 5F
46 61 6D 69 6C 79 5C 32 30 30 34 5C 32 30 30 34 5F 30 31 5C 32 30 30 34 5F 30 31 5F 30 30 37 2E
6A 70 67 | 00 00 39 00 00 00 09 00 00 A0 2D 00 00 00 31 53 50 53 55 28 4C 9F 79 9F 39 4B A8 D0 E1
```

Figure 3. Application, Volume Serial, Volume Name, Label, Path, Filename  
Filename

The application: 50 00 42 00 72 00 75 00 73 00 68 refers to **P B r u s h** which is the Microsoft Paint application.

The Volume serial is a 32 bit number which is based on the date and time the volume was created. The date and time cannot easily be calculated from the

volume serial number, however if an investigator has the date and time that the volume was created (discernible from other sources such as the boot sector and root directory entry), they can certainly confirm that the current volume matches the entry in the Jumplist (Wilson, 2005). The Volume Serial should be read in reverse order and in this case (Figure 3) is therefore OE1B708D

The <volume name><drive>:\<path><filename> for the example at hand (Figure 3) would be:

```
New Volume A:\_sub\sub\pictures\Family\2004\2004_01\
2004_01_007.jpg
```

Note that if the file was accessed from a network drive, the volume path would reflect the network path. for instance, the Volume Name in the following case (Figure 4) is \\HARJDESKTOP and the file is a network accessed file

```
6B 00 5F 00 74 00 6D 00 70 00 00 00 16 00 00 00 14 03 00 00 01 00 00 A0
|5C 5C 48 41 52 4A 44 45 53 4B 54 4F 50|5C 56 69 64 65 6F 20 43 61 70 74
75 72 65 5C 44 49 56 78 5C 61 67 6B 5F 74 6D 70 00 00 00
```

Figure 4. Jumplist configuration file pointing to external machine

This is very useful as the Jumplist configuration file points to files as well as the volumes, GUIDs and times that the files were accessed. A Jumplist configuration file may evidence different computer names indicating with some approximation (if we can consider the *added to Jumplist* times) as to when it might have been changed.

## 4.7 Computer Name, Volume GUID and File Object ID

### 4.7.1 Computer Name

The computer name is used to identify the machine on a network, this can be found repeated twice for each record (Figure 5).

```
6A 70 67|00 00 39 00 00 00 09 00 00 A0 2D 00 00 00 31 53 50 53 55 28 4C 9F 79 9F 39 4B A8 D0 E1
D4 2D E1 D5 F3 11 00 00 07 00 00 00 0B 0|Computer Name|F 00 00 00 00 00 00 00 00 60 00
00 00 03 00 00 A|Volume GUID|00 00 00 00 00 00|68 61 72 6A 2D 70 63|00 00 00 00 00 00|5C EF
40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95|4F 59 FC 2F A7 6C E0 11 96 42 80 00 60 0F E8 00|5C EF
40 2A 5E 79 13 48 AA 12 C1 A8 F4 53 F1 95|4F 59 FC 2F A7 6C |File Object ID| 80 00 60 0F E8 00|00 00
```

Figure 5. Computer Name, Volume GUID, & File Object ID

This refers to the computer name to which the Jumplist configuration file refers and not the computer name from which the item was accessed.

### 4.7.2 Volume GUID

Volume names are created by a user and can be modified thereafter, they may not be unique, hence an operating system assigns a Volume GUID (Globally Unique Identifier - also known as a *unique volume name*) to the volume. The Volume

GUID is split into 5 sections comprising of 4 bytes, 2 bytes, 2 bytes, 2 bytes 6 bytes (16 bytes in total) in this case is:

```
5CEF402A-5E79-1348-AA12-C1A8F453F195
```

To establish the Volume GUID, the first 3 sections are to be read backwards, the volume GUID therefore is:

```
2A40EF5C-795E-4813-AA12-C1A8F453F195
```

The following Jumplist configuration file (Figure 6) was extracted from the same volume (note that the machine name is the same), however the file to which the entry refers, was accessed from an external USB device with the volume GUID:

```
5F5F34FA-9615-4198-9782-CA7CFE78397E
```

```
00 00 00 60 00 00 00 03 00 00 A0 58 00 00 00 00
00 00 00|68 61 72 6A 2D 70 63|00 00 00 00 00 00
00 00 00|FA 34 5F 5F 15 96 98 41 97 82 CA 7C FE
78 39 7E|89 5A FC 2F A7 6C E0 11 96 42 80 00 60
0F E8 00|FA 34 5F 5F 15 96 98 41 97 82 CA 7C FE
78 39 7E|89 5A FC 2F A7 6C E0 11 96 42 80 00 60
0F E8 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 4C 00 00 00 01 14 02 00
00 00 00 00 C0 00 00 00 00 00 00 46 83 00 00 00
10 00 00 00 14 B3 9E 6B 67 16 CB 01 98 E8 FE E4
```

Figure 6. Jumplist configuration file pointing an external USB device

### 4.7.3 File Object ID

The File Object ID uniquely identifies the file on the hard disk, it is found in a Jumplist immediately after both instances of the Volume GUID. The File Object ID is structured in the same way as the Volume GUID and is split into 5 sections comprising of 4 bytes, 2 bytes, 2 bytes, 2 bytes 6 bytes (16 bytes in total) as follows:

```
4F59FC2F-A76C-E011-9642-8000600FE800.
```

Like the Volume GUID, the first 3 sections of the File Object ID are to be read backwards, the File Object ID in this case therefore is:

```
2FFC594F-6CA7-11E0-9642-8000600FE800.
```

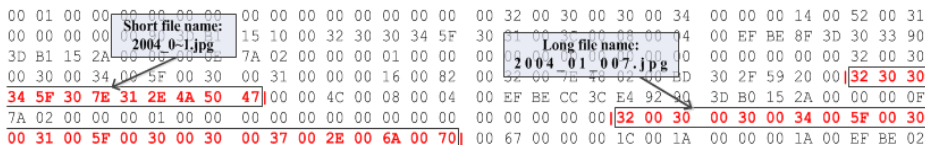
### 4.8 File Name and path

File and path names are presented as both long Windows names as well as 8 character shortened names, each element of the path is presented in turn in this way. So for instance the short and long filenames of the

<drive>:\<path><filename>:

A:\\_sub\sub\pictures\\_Family\2004\2004\_01\  
2004\_01\_007.jpg

Are presented as follows:



The image shows a hex dump of a file list entry. Two boxes highlight specific parts: 'Short file name: 2004\_0-1.jpg' and 'Long file name: 2004\_01\_007.jpg'. The hex dump shows the raw bytes of the file list entry, with some bytes highlighted in red to indicate the start of the short and long filename fields.

Figure 7. Long and short filenames in the Jumplist configuration file

## 5. CONCLUSIONS AND FUTURE WORK

We conclude that in particular, the Jumplist configuration file provides the following valuable evidence which is not easily or at all available elsewhere:

- The Jumplist configuration file contains evidence of countless/numerous file accesses, thereby providing scope for further lines of enquiry for the investigator particularly in noting filenames of files which were ‘once’ accessed on the current volume.
- The existence of a file entry in the Jumplist points specifically to the access of a particular file/volume (even though it might later be removed/deleted) supported by a timestamp (indicating MAC times) thereby providing a near-historical record from which a timeline could be built. This information is generally not available elsewhere.
- Jumplist configuration files can link files to particular applications. For instance a Firefox Jumplist configuration file pointing to a particular website can be evidence that the user used Firefox to access the website.

The issue of timeline development based on the Jumplist configuration file is an area which requires further research as does the question of being able to forge Jumplist entries.

## ACKNOWLEDGMENTS

We wish to acknowledge the kind help and input provided by: Andrew Sithers (Microsoft), Matthew Birkin (IT director, School of Computing and Mathematics, University of Derby) and Alexander Zigelski (author of Jumplist Launcher)

## **AUTHOR BIOGRAPHIES**



Harjinder Singh Lallie (BSc., MSc., MPhil, ABCS) is a senior teaching fellow in Cybersecurity at the University of Warwick (International Digital Laboratory, WMG). He has previously led courses successfully in Digital Forensics and Security at the University of Derby. His research focus is in the area of Digital Forensics and Information Security and is currently studying towards his PhD.



Parmjit Singh Bains (BSc, MSc) is a former student at the University of Derby (School of Computing and Mathematics) having completed his MSc in Forensic Computing & Security. His research interests involve smart phones utilising Cloud Computing for criminal intent, Cyber Forensics and identifying use of Anti-Forensics.

## **REFERENCES**

- Andersson, A., & Ottmann, T. (1991). *Faster uniquely represented dictionaries*. Paper presented at the Foundations of Computer Science.
- Brönnimann, H., Cazals, F., & Durand, M. (2003). Randomized Jumlists: A Jump-and-Walk Dictionary Data Structure *Lecture Notes in Computer Science, 2607/2003*, 283-294.
- Hedgehog. (2011). JumpList Launcher. Retrieved 18 April 2011, from <http://en.www.ali.dj/jumplist-launcher/>
- MSDN. (2011a). Taskbar Extensions. Retrieved 18 April 2011, from [http://msdn.microsoft.com/de-de/library/dd378460\(vs.85\).aspx#jump\\_lists](http://msdn.microsoft.com/de-de/library/dd378460(vs.85).aspx#jump_lists)
- MSDN. (2011b). SHAddToRecentDocs Function. Retrieved 12 April 2011, from [http://msdn.microsoft.com/en-us/library/bb762105\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/bb762105(v=vs.85).aspx)
- Ottman, T. (1991). Trees — a personal view *Lecture Notes in Computer Science, 555/1991*, 243-255.
- Regdat. (2011). Jumplist Backup Restore. Retrieved 12 April 2011, from <http://www.regdat.com/>
- Smulikowski, P. (2009). First Look at the Windows 7 Forensics - Forensic implications of the new Windows 7. University of Strathclyde, Strathclyde.
- Wilson, C. (2005). Volume Serial Numbers and Format Date/Time Verification. Retrieved 18 April 2011, from <http://www.digital-detective.co.uk/documents/Volume%20Serial%20Numbers.pdf>